Università degli Studi "G. D'Annunzio"
Dipartimento di Scienze

Reti neurali e rischio di credito: stato dell'arte e analisi sperimentale

Giacomo di Tollo

17 novembre 2005

Technical Report no. R-2005-003 *Research Series*

Reti neurali e rischio di credito: stato dell'arte e analisi sperimentale

Giacomo di Tollo [1]

Dipartimento di Scienze
Università degli Studi "G. D'Annunzio"
Viale Pindaro, 42 – 65127 Pescara, Italia

17 novembre 2005

Sommario. Il presente lavoro si pone come obiettivo quello di progettare una rete neurale in grado di classificare l'insolvenza: disponiamo di un insieme di indicatori (di bilancio e dei modelli "Centrale dei Rischi" e "andamentale") relativi a 106 piccole e medie imprese italiane per un triennio e ci proponiamo di utilizzarli per l'addestramento della rete. La rete risultante sarà caratterizzata dal fatto di prendere in input gli indicatori ritenuti più predittivi e di fornire in output il valore 0 se l'azienda è da ritenersi *sicura* oppure 1 se è da ritenersi *insolvente*.

Keywords: *Reti neurali, Rischio di Credito*

Indice

Introduzione **4**

1 Le Reti Neurali **5**
1.1 Cosa sono le reti neurali 6
1.2 Il neurone biologico 6
1.3 Le reti neurali artificiali 8
1.4 Il neurone binario a soglia 10
1.5 Evoluzione della ricerca sulle reti neurali 12
1.6 Black-Box 15
1.7 Apprendimento nelle reti neurali 17
1.7.1 Supervised learning 18
1.7.2 Unsupervised learning 18
1.7.3 Reinforcement learning 18
1.8 Principali modelli teorici 20
1.8.1 Il Percettrone 20
1.8.2 Back-Propagation 24
1.8.3 Reti di Ward 31
1.8.4 Reti Jump-Connection 31
1.8.5 Reti General-Regression 31
1.9 Vantaggi e svantaggi delle reti neurali 32

2 Le reti neurali ed il rischio di credito **34**
2.1 Il nostro caso: *Il Rischio di Credito* 38
2.2 Il modello più comune utilizzato dalle banche 40
2.3 Le reti neurali ed il rischio di credito 43
2.4 Basilea 44

3 Pre-processing dei dati **50**
3.1 Outlier 50
3.2 Normalizzazione dei dati 52
3.3 Descrizione dei dati a disposizione 54
3.4 Il trattamento dei dati mancanti 57
3.5 Considerazioni sui dati in esame 58
3.6 Ipotesi di utilizzo degli attributi RAE e SAE 62
3.7 Anomalia nella serie storica 64
3.8 Analisi di correlazione delle variabili 64

4 Criteri per la costruzione della rete **69**
4.1 Neuroni di input 70
4.2 Neuroni nascosti 71
4.3 Neuroni di output 72
4.4 Il coefficiente di apprendimento 72
4.5 La funzione di attivazione 72
4.6 Remapping function 73
4.7 Training set e test set 74

5 Costruzione della rete **74**
5.1 Gli esempi a disposizione 74
5.1.1 Training set e Test set 75
5.2 JavaNNS 75
5.3 La classificazione 77
5.4 Esperimenti con 8 attributi 78
5.4.1 Esperimenti con la rete cablata 79
5.4.2 Esperimenti con la rete standard 95
5.5 Esperimenti con 11 attributi 101
5.5.1 Esperimenti con la rete cablata 102
5.5.2 Esperimenti con la rete standard 104

Sviluppi futuri **108**

Conclusioni **109**

Elenco delle figure **113**

Elenco delle tabelle **113**

Bibliografia **116**

Indice

Introduzione

Le aziende possono finanziare le loro attività ed i loro progetti e fronteggiare le loro situazioni di difficoltà ricorrendo al capitale proprio oppure a fonti di finanziamento esterne. In questo secondo caso la situazione più frequente è quella in cui l'azienda chieda un finanziamento ad una banca.

Le banche si ritrovano così a fronteggiare diverse richieste di finanziamento da parte di una moltitudine di aziende, e in questa situazione devono decidere se concedere il prestito o meno.

La decisione in questione viene presa da esperti addetti che possono utilizzare diversi strumenti a tale scopo. I modelli più utilizzati sono i modelli lineari, che utilizzano indicatori ricavati dal bilancio per stimare lo stato di salute e quindi la capacità di restituire il credito concesso. Ultimamente però si stanno affermando dei nuovi strumenti di calcolo che si ispirano al funzionamento del cervello: le Reti Neurali. La loro capacità di modellare relazioni non lineari tra variabili e di offrire buone prestazioni in presenza di dati incorretti e affetti da rumore ha portato ad un aumento dell'attenzione rivolta a questi strumenti di calcolo, anche se la non universalità delle soluzioni proposte, la mancanza di una solida architettura teorica di base e l'approccio principalmente empirico ne hanno in grande misura ostacolato la diffusione applicativa. Il fatto che le reti neurali vengano paragonate a scatole nere (*black-box*), è esemplificativo anche della diffidenza che le circonda: molti pensano alle reti neurali come scatole magiche che prendono dei dati in ingresso e ci forniscono dei risultati in uscita, senza farci capire come pervengono alla soluzione: in linea di principio questo è vero, ma non bisogna dimenticare che le reti neurali sono e rimangono degli strumenti di calcolo e che in molti casi forniscono delle performance migliori di modelli già conosciuti e sperimentati.

Il presente lavoro si pone come obiettivo quello di progettare una rete neurale in grado di classificare l'insolvenza: disponiamo di un insieme di indicatori (di bilancio e dei modelli "Centrale dei Rischi" e "andamentale") relativi a 106 piccole e medie imprese italiane per un triennio e ci proponiamo di utilizzarli per l'addestramento della rete. La rete risultante sarà caratterizzata dal fatto di prendere in input gli indicatori ritenuti più predittivi e di fornire in output il valore 0 se l'azienda è da ritenersi *sicura* oppure 1 se è da ritenersi *insolvente*.

Il primo capitolo si occupa di definire cos'è una rete neurale e di fornirci una guida circa il background teorico ed i principali modelli; il secondo capitolo ci introduce nella problematica del rischio di credito e di come esso rivesta importanza nella gestione delle banche; il terzo capitolo consiste dell'analisi dei dati a nostra disposizione; il quarto capitolo ci fornisce le linee guida per la costruzione di una rete neurale per la classificazione dell'insolvenza, che viene sviluppata nel quinto capitolo.

1 Le Reti Neurali

Gli elaboratori attuali sono in grado di effettuare calcoli complessi e risolvere problemi che fino a pochi decenni fa sembravano inaffrontabili. La ricerca scientifica ed il nostro vivere quotidiano hanno subito un notevole miglioramento grazie all'introduzione dei moderni calcolatori che, nonostante l'evoluzione tecnologica degli ultimi decenni, sono ancora ispirati all'architettura definita da Von Neumann nel secolo scorso. Tuttavia, essi non sono ancora in grado di risolvere dei problemi che vengono invece affrontati dall'uomo senza particolari difficoltà. E' il caso ad esempio della percezione sensoriale e del riconoscimento di immagini. Si pensi al riconoscimento visivo: anche un bambino è in grado di riconoscere il volto del padre in qualsiasi prospettiva gli venga presentato; per l'elaboratore classico invece questo è un problema gravoso, visto che una figura tridimensionale può essere vista da diverse prospettive e il riconoscimento di questa è una operazione che risulta essere ben più complessa di un semplice confronto tra oggetti.

Questi problemi sono da ricondurre al fatto che l'elaboratore utilizza una rappresentazione ed elaborazione della conoscenza simbolica: l'interazione dell'uomo con la macchina al fine di programmarla ed il successivo funzionamento della stessa si fondano su linguaggi definiti da regole formali ben precise, in base alle quali è possibile accertarsi che una sequenza di simboli abbia un significato; la rappresentazione dei dati è costituita da una sequenza di simboli a cui è possibile dare un significato secondo delle regole definite.

Al contrario, nelle forme di vita intelligenti la conoscenza non è rappresentata in modo esplicito o simbolico, ma possiamo dire che essa è distribuita all'interno del sistema. Possiamo affermare che gli esseri intelligenti utilizzano un tipo di rappresentazione ed elaborazione sub-simbolica: osservando la rappresentazione di una conoscenza non si è in grado di risalirne al significato. Il cervello non utilizza meccanismi per la rappresentazione dei simboli, e le operazioni avvengono sugli elementi che lo compongono, senza che essi siano in corrispondenza con "qualcosa" al suo esterno.[1]

Spesso risulta impossibile dare ad un elaboratore tutte le informazioni necessarie per poter affrontare un problema complesso, per la risoluzione del quale sono necessarie conoscenze pregresse. Proprio partendo da questi limiti molti ricercatori hanno sentito l'esigenza di elaborare nuovi paradigmi di calcolo ispirati al funzionamento neuro-fisiologico del cervello umano. E' nata così la Neuro-Computazione, cioè quella disciplina che, prendendo spunto dalle regole della neuro-biologia, permette

[1] L'evoluzione degli studi sul cervello ci ha portato a capire che in realtà esiste una specializzazione delle diverse aree del cervello per rispondere e processare stimoli di diversi tipi, per cui il ragionamento sopra enunciato è da ricondurre al funzionamento di una specifica parte del cervello piuttosto che al sistema globale.

di inferire efficacemente le relazioni eventualmente esistenti tra variabili di input e variabili di output.

E' da notare che i primi contributi alla diffusione di tale disciplina sono dovuti a neuropsicologi e fisici, alcuni dei quali mossi dall'intenzione di comprendere e modellare efficacemente la rappresentazione della conoscenza ed il comportamento del cervello umano. Anche se da allora diversi passi in avanti sono stati fatti nella ricerca in questa direzione, questo approccio si è subito scontrato con la terribile difficoltà nel decifrare i meccanismi di un sistema complesso quale è il cervello. Purtuttavia, matematici, ingegneri e ricercatori hanno altrettanto velocemente capito che non era necessaria una perfetta conoscenza del cervello per i propri scopi: essi volevano semplicemente applicare i principi che sono alla base del suo funzionamento alla risoluzione di problemi che riguardano un dominio ben specificato. Sono così nate le reti neurali che, attraverso vicende alterne, sono state applicate ai più diversi ambiti, con prestazioni degne di nota in problemi quali classificazione, filtering, associazione di pattern, ottimizzazione, concettualizzazione e predizione.

Prima di iniziare con la nostra trattazione occorre precisare che le reti neurali, nonostante la mistificazione della quale sono state oggetto, sono semplicemente degli stumenti di calcolo.

1.1 Cosa sono le reti neurali

Le reti neurali sono strumenti di calcolo che si ispirano al funzionamento del cervello umano; esse sono composte da elementi di elaborazione operanti in parallelo che, presi singolarmente sono in grado di effettuare semplici operazioni: integrano l'informazione proveniente da altri elementi eseguendo una funzione di attivazione e comunicano il risultato di tale elaborazione ad altri elementi a cui sono collegati. L'interazione di queste semplici operazioni tramite elaborazione distribuita porta all'esecuzione di compiti molto complessi. Questi elementi di elaborazione si ispirano ai neuroni biologici, dei quali rappresentano però una forte semplificazione.

1.2 Il neurone biologico

All'interno del cervello umano è possibile identificare un gran numero di unità di elaborazione: i neuroni. Essi sono composti da tre regioni: il corpo cellulare (soma), i dendriti e l'assone.

Il corpo cellulare contiene il nucleo del neurone ed è rivestito da una membrana contenente dei canali che permettono la comunicazione tra l'interno e l'esterno del soma. I dendriti rappresentano i canali di input del neurone: essi ricevono i segnali provenienti dai neuroni a cui sono connessi. L 'assone invece è il canale di output: la sua lunghezza può spingersi a grande distanza dal soma, ed esso rappresenta il percorso attraverso il quale il segnale emesso dal neurone si propaga verso altre parti del sistema nervoso, anche molto remote.

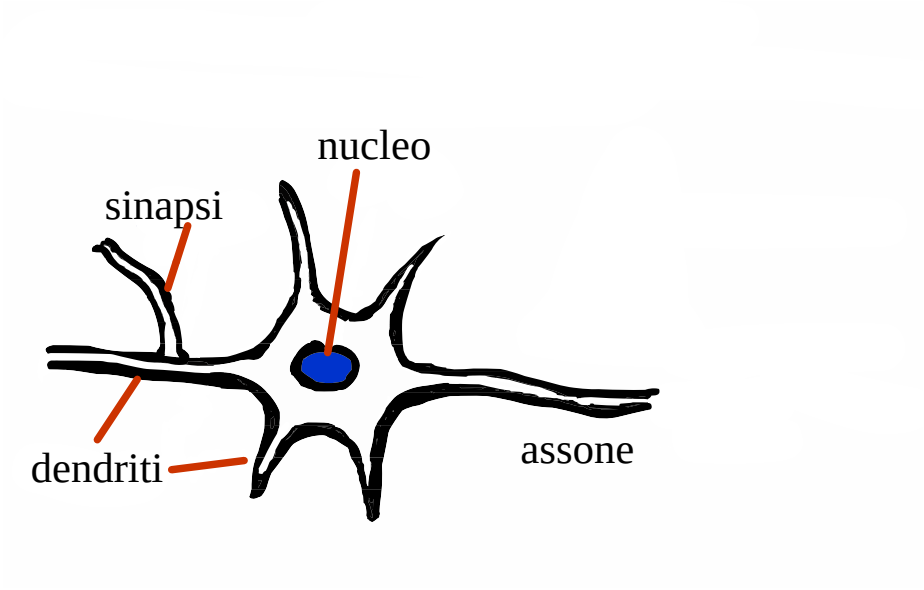

Figura 1. *Il neurone biologico*

Il trasferimento dell'informazione da assone a dendrite avviene in zone di contatto altamente specializzate: le sinapsi. Ogni neurone può avere un numero di sinapsi variabile da poche centinaia ad alcune migliaia. L'informazione trasmessa dal soma è costituita da un segnale elettrico che varia da una decina (in stato di riposo) a circa 500 impulsi al secondo. Il segnale trasmesso è tanto più alto quanto più il neurone viene eccitato. Questo segnale (potenziale d'azione) parte dal soma e viaggia lungo l'assone, fino a raggiungere la terminazione pre-sinaptica. Qui il potenziale d'azione permette il rilascio di un neurotrasmettitore che si propaga nella sinapsi e raggiunge la zona dendritica post-sinaptica, generando un nuovo segnale elettrico che, viene combinato a quello degli altri dendriti, e trasmesso lungo l'albero dendritico. I segnali possono avere effetti diversi a seconda che la sinapsi sia eccitatoria o inibitoria, e la loro combinazione provoca una variazione del cosiddetto "potenziale di membrana" del soma di arrivo. Se questo supera un certo "bias" (soglia) il neurone genera un nuovo segnale bio-elettrico che viene trasmesso tramite lo stesso meccanismo ai neuroni a cui è connesso, altrimenti nessun segnale verrà trasmesso (a parte il segnale spontaneo sopra citato).

In questo meccanismo una particolare importanza è data alle sinapsi, in quanto

l'efficacia di trasmissione del segnale, detta forza del legame sinaptico,varia da sinapsi a sinapsi. Il segnale ricevuto dipenderà quindi dagli impulsi trasmessi dagli altri neuroni e dalla forza del legame sinaptico. Già nel 1949 Hebb [13] dimostrò che questa forza è soggetta a cambiamento, rivelando che l'apprendimento è dovuto alle sinapsi. Comunque la modifica della forza del legame sinaptico può avvenire anche in modo temporaneo, e le ricerche attuali sono indirizzate verso l'identificazione dei fattori in base ai quali questa modifica può avvenire. Possiamo qui di seguito riassumere le principali caratteristiche del sistema di elaborazione del cervello:

- L'impulso elettrico trasmesso dal neurone viaggia alla velocità di 130 metri/secondo;
- Nel cervello è presente un numero di neuroni variabile tra qualche centinaia di miliardi e qualche migliaia di miliardi;
- La densità dei neuroni è di circa 80000 neuroni/millimetro quadrato e le connessioni possono essere presenti tra neuroni molto distanti tra loro;
- I gruppi di neuroni elaborano l'informazione simultaneamente: abbiamo cioè una elaborazione parallela. Questo porta all'emergere di processi cognitivi;
- La conoscenza è distribuita in tutta la rete e l'evoluzione della struttura cerebrale è continua;
- La rete è robusta rispetto ai guasti: il malfunzionamento di qualche neurone non pregiudica il funzionamento globale del cervello, nei confronti del quale si nota soltanto una diminuzione delle prestazioni.

1.3 Le reti neurali artificiali

Le reti neurali artificiali si ispirano al comportamento del cervello, del quale rappresentano una semplificazione. Esse sono composte da unità elementari (i neuroni formali) e da connessioni pesate ed orientate tra esse (le sinapsi). Ad ogni neurone è associato un valore numerico che rappresenta il valore che verrà trasferito dal neurone e che dipende dai segnali di ingresso trasmessi dalle sinapsi, da una funzione di attivazione ed una di output. Anche alle sinapsi è associato un numero, che determina l'efficacia della trasmissione: positivo nel caso di sinapsi eccitatorie e negativo altrimenti. La funzione di attivazione calcola il valore di input del neurone partendo dai valori di output pesati dei neuroni precedenti. La funzione di output poi utilizza questo valore per determinare il valore che verrà trasmesso dal neurone

(alcune funzioni utilizzano anche il valore precedente dell'attivazione).[2]
I neuroni possono essere classificati, in relazione alla loro funzione, in tre categorie:

- neuroni di input, cioè i neuroni le cui attivazioni rappresentano i valori di input della rete;
- neuroni di output, cioè i neuroni le cui attivazioni rappresentano l'output della rete;
- neuroni nascosti, i neuroni rimanenti, così chiamati perché non sono visibili dall'esterno.

Esistono tuttavia particolari tipologie di reti in cui un neurone nascosto può essere considerato come neurone di input oppure di output, anche se non è collocato ai bordi della rete neurale.
Il comportamento di una rete neurale è determinato da:

- la funzione di attivazione, che determina il valore di output del neurone a partire dall'attivazione dei neuroni connessi con esso;
- le sinapsi, che determinano la quantità di attivazione del neurone che viene trasferita ai neuroni ai quali questo è connesso;
- la topologia della rete;
- la dinamica temporale, che determina quando aggiornare i valori di attivazione dei diversi neuroni ed il criterio per aggiornarli (se tutti i neuroni devono essere aggiornati simultaneamente oppure se solo alcuni di questi devono essere aggiornati, ed in quest'ultimo caso come scegliere i neuroni da aggiornare).

Questi fattori possono essere determinati a priori dall'utente oppure si può fare in modo che sia la rete a determinare il valore di questi parametri tramite un processo di adattamento, che prende il nome di apprendimento. Generalmente una rete è costruita in un modo ibrido, in quanto alcune caratteristiche vengono definite dallo sviluppatore mentre altre vengono sottoposte ad apprendimento. Storicamente sono stati i valori delle sinapsi ad essere sottoposti ad apprendimento, e ciò ha portato all'elaborazione di algoritmi di apprendimento che sono diventati comuni ed hanno portato ad un aumento della diffusione delle reti neurali (un esempio su tutti: Back-Propagation). Ultimamente il paradigma dell'apprendimento è stato utilizzato anche su altri parametri, anche grazie all'introduzione di nuove tecniche di Intelligenza Artificiale quali gli algoritmi genetici che permettono, ad esempio, di selezionare la

[2] In letteratura l'uso dei termini attivazione ed output non è uniforme. D'ora in poi quando si parlerà di attivazione del neurone si vorrà indicare il valore di output, e la funzione di attivazione starà ad indicare la funzione che determina l'output del neurone.

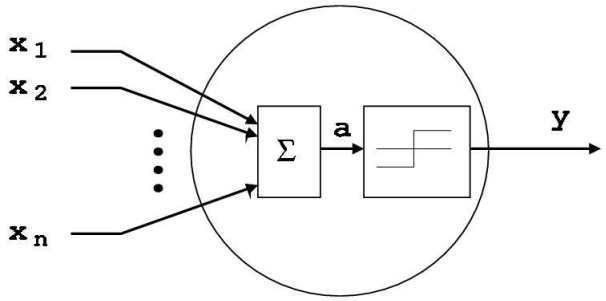

Figura 2. *Il neurone binario a soglia*

migliore architettura della rete. E' da notare infine che generalmente gli algoritmi di apprendimento sono sviluppati per essere applicati ad una particolare architettura di rete, quindi la scelta di un determinato algoritmo spesso influenza fortemente la scelta dell'architettura e della dinamica temporale della rete.

1.4 Il neurone binario a soglia e la funzione di attivazione

Come abbiamo accennato, una rete neurale può essere vista come un grafo composto da nodi (i neuroni appunto, proposti da McCulloch e Pitts [25] nel 1943 e chiamati *Processing Elements*) e connessioni (le sinapsi) pesate ed orientate tra essi. Il neurone è caratterizzato da un operatore di input, che determina la sommatoria dei valori

$$x_1, \ldots, x_n$$

in entrata, determinati dall'uscita dei neuroni connessi al suo ingresso, ognuno moltiplicato per il peso w della rispettiva sinapsi: la sommatoria determina il potenziale post-sinaptico del neurone j ($net(j)$).

$$net(j) = \sum_i y_i w_{ij}$$

Questo valore viene poi processato da una funzione opportunamente definita. La prima ad essere utilizzata fu la funzione di Heaviside, che confronta il valore della sommatoria con una soglia (*treshold* o *bias*) per produrre il valore di output: se il potenziale post-sinaptico è minore o uguale della soglia l'output del neurone sarà pari a zero, altrimenti sarà pari a uno.

$$y_j = \begin{cases} 0 & \text{se } net(j) \leq \theta \\ 1 & \text{altrimenti} \end{cases}$$

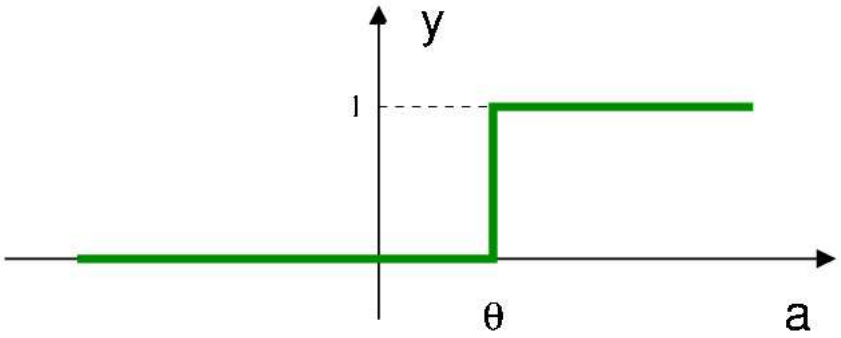

Figura 3. *Funzione di HEAVISIDE*

In questo caso consideriamo che il valore di output del neurone possa assumere valore 0 o 1 (funzione a gradino), ma si può anche stabilire questo possa assumere valore -1 o 1 (funzione segno).

$$y_j = \begin{cases} \text{-1} & \text{se } net(j) \leq \theta \\ 1 & \text{altrimenti} \end{cases}$$

Questa funzione è molto semplice da calcolare ma non è differenziabile (la differenziabilità è requisito essenziale per la convergenza dell'algoritmo Back-Propagation, si veda oltre). Oggi sono utilizzate altre funzioni, la più usata delle quali è la funzione logistica o sigmoide:

$$y_j = \frac{1}{1 + A^{net(j)-\theta}}$$

i cui valori appartengono all'intervallo $(0, 1)$ ed ha il merito di ridurre l'eventuale interferenza degli outlier. La scelta del valore A è arbitraria, ma viene generalmente fatta corrispondere ad e ($e = 2,71828283$) ed è questa la scelta che effettueremo da qui in poi. E' da notare che la scelta del valore A influenza la ripidità della funzione, che converge verso la funzione a gradino per valori di A molto grandi e verso la funzione lineare per valori di A molto piccoli. Un'altra funzione molto usata è la funzione tangente iperbolica, che permette all'output di assumere valori compresi nell'intervallo $(0, 1)$.

$$y_j = tanh\Big(net(j) - \theta\Big) = \left(\frac{e^{(net(j)-\theta)} - e^{-(net(j)-\theta)}}{e^{(net(j)-\theta)} + e^{-(net(j)-\theta)}}\right)$$

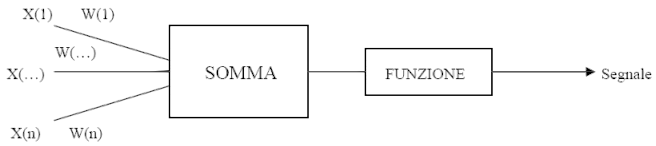

Figura 4. *Rappresentazione di un neurone artificiale*

Sebbene non esista una regola generalizzata per determinare che funzione utilizzare, si ritiene che essa debba essere continua, differenziabile e non lineare. Quest'ultimo requisito sembrerebbe escludere l'utilizzo della funzione lineare

$$y_j = a \cdot net(j) + b$$

che invece ha trovato alcune applicazioni nelle unità di output della rete, in quanto evita che il risultato tenda verso il minimo o il massimo. Il suo utilizzo nei neuroni nascosti invece risulta perlomeno inopportuno, in quanto verrebbe a determinare una connessione basata proprio sul tipo di funzione che si vuole evitare con l'utilizzo della rete neurale. Diverso è il caso della funzione a soglia lineare Clip01

$$y_j = \begin{cases} 0 & \text{se } net(j) < 0 \\ 1 & \text{se } net(j) > 1 \\ net(j) & \text{altrimenti} \end{cases}$$

che può assumere valori compresi nell'intervallo $[0, 1]$. Sono state proposte alcune funzioni che calcolano l' attivazione del neurone in base non soltanto alla funzione scelta, ma anche all'attivazione precedente del neurone. La funzione di attivazione ha lo scopo di ricondurre l'output entro un intervallo predeterminato, che è in genere $[0, 1]$ oppure $[-1, 1]$, altrimenti il suo valore potrebbe assumere valori troppo grandi. Occorre poi fare una precisazione circa la natura delle funzioni di attivazione: possiamo classificarle in funzioni deterministiche e funzioni stocastiche. Nelle prime l'attivazione del neurone è calcolata deterministicamente in funzione dei pesi delle sinapsi. dei valori di input ed eventualmente della soglia e dell'attivazione precedente. Nelle seconde l'attivazione è una funzione probabilistica dell'attivazione corrente e dei pesi delle sinapsi.

1.5 Evoluzione della ricerca sulle reti neurali

La data di nascita dello studio sulle reti neurali è convenzionalmente fatta risalire al 1943, anno in cui McCulloch e Pitts [25] pubblicarono un articolo in cui si

avanzavano delle ipotesi sul metodo di calcolo usato dal cervello. In questo studio il neurone veniva rappresentato come un elemento di decisione logica a due valori, con attivazione a soglia. Inoltre venne ipotizzata una prima architettura neurale, in cui i pesi delle sinapsi erano fissi. Seguì il lavoro di Hebb [13], che sosteneva che l'apprendimento è dovuto ad una modifica dell'efficacia trasmissiva delle sinapsi: se due neuroni sono attivi simultaneamente, l'efficacia della connessione sinaptica tra i due neuroni aumenta.

Da questa spiegazione derivano le cosiddette "regole di Hebb", utilizzate poi in diversi algoritmi di apprendimento, in cui il valore di una connessione sinaptica viene incrementato ogni volta che l'unita pre-sinaptica (cioè quella che manda il segnale) e quella post-sinaptica (quella che riceve il segnale) sono attive. Esistono due regole Hebbiane: la regola pre-sinaptica e quella post-sinaptica. In entrambe il meccanismo di incremento del valore della connessione sinaptica è quello descritto in precedenza, mentre viene gestito in modo diverso il meccanismo di decremento: mentre nella regola pre-sinaptica il peso della connessione viene decrementato se l'unità pre-sinaptica è attiva e quella post-sinaptica è inattiva, nella regola post-sinaptica questo viene decrementato se l'unità post-sinaptica è attiva e quella pre-sinaptica è inattiva.

Importantissima fu poi la ricerca condotta da Rosenblatt nel 1962 [33], in cui venne proposta una rete neurale con due strati (uno di input ed uno di output) di unità a soglia (come quelli di McCulloch-Pitts) e con sinapsi modificabili in cui ogni unità di input è connessa a tutte le unità di output. Questo lavoro dimostrò che il modello proposto (chiamato perceptron) può essere addestrato a classificare un insieme di istanze in base alle loro caratteristiche simili: fu usato per il riconoscimento di forme ed era in grado di apprendere delle relazioni casuali tra le istanze in ingresso. A Rosenblatt è dovuto il primo teorema di convergenza del perceptron: l'apprendimento sui valori di addestramento avviene tramite la convergenza dei pesi verso i valori che garantiscono la risposta corretta della rete. Simile al Perceptron era l'Adaline (Adaptative Linear Element) di Widrow e Hoff (1960) [41]. Le ricerche di questi autori furono importanti perché offrirono dei risultati concreti circa le applicazioni pratiche e la capacità di generalizzazione delle reti neurali. Presto emersero però dei grandi limiti: nel 1969 Minsky e Papert [26] dimostrarono che percettroni e adaline non erano in grado di distinguere una T da una C e che erano in grado di risolvere soltanto problemi linearmente separabili. Dimostrarono che il processo di convergenza era troppo lento, che il numero di circuiti logici necessari era a volte troppo grande ed espressero totale sfiducia circa ulteriori ricerche in quel campo. La posizione di rilievo di questi autori nel panorama scientifico internazionale congelò per un certo periodo gli studi sul settore, dovuti anche al fatto che il governo degli Stati Uniti decise di non rinnovare i finanziamenti (si veda Floreano, Nolfi [7]) . Fu necessario attendere l'inizio degli anni ottanta per arrivare a nuovi sviluppi sulle ricerche. In quegli anni gli studi nuovi impulsi grazie allo sviluppo di nuovi algoritmi di

apprendimento e di nuove strutture, uniti ad un accresciuto bagaglio di competenze teoriche.

Nel 1982 Hopfield [16] propose un modello completamente nuovo di rete neurale basata su neuroni completamente interconnessi tra di loro ma senza auto-connessioni, il cui comportamento può essere compreso paragonandolo ad un sistema dinamico. In questa rete non è corretto parlare di strato di input e di output, in quanto ogni neurone può fungere da input ed output. I pesi sono calcolati deterministicamente e non variano durante la fase di utilizzo. Quando viene presentato un pattern in ingresso, la rete determina una certa uscita che verrà ripresentata in ingresso. In questo modo la rete si evolve autonomamente, fino a raggiungere uno stato in cui l'uscita è uguale all'ingresso. Il limite di questo modello risiede nel limitato numero di pattern memorizzabili (0,138 volte il numero dei neuroni), tuttavia ha trovato discreta applicazione come memoria auto-associativa per il riconoscimento di configurazioni corrotte e il recupero di informazioni mancanti e per la risoluzione di problemi di ottimizzazione ed è stato studiato a fondo a causa della sua somiglianza con alcuni modelli di meccanica statistica. Pochi anni dopo (Hinton e Sejnowski nel 1987 [36]; Ackley, Hinton e Sejnowski nel 1988 [1]) venne proposto un modello per migliorare la rete di Hopfield introducendo dei neuroni nascosti ed una funzione stocastica di output. Questo modello è stato chiamato Macchina di Boltzmann ed è in grado di risolvere problemi non linearmente separabili. Il fatto di possedere neuroni nascosti permette alla rete di aumentare notevolmente le capacità di immagazzinamento. Questa rete dimostra inoltre di avere più flessibilità rispetto al tradizionale approccio back-propagation (vedi oltre) ma la procedura di addestramento risulta essere molto complicata e richiede tempi molto lunghi; questo ne ha limitato fortemente l'utilizzo. Contemporaneamente al lavoro di Hopfield, Kohonen (la prima pubblicazione è del 1982, seguita da altre nel 1989 [20] [21]) propose un modello di rete neurale completamente nuovo, che rappresenta uno dei più importanti contributi europei alla ricerca sulle reti neurali, e che ha il merito di essere stato il primo a proporre un sistema completo di addestramento non supervisionato (si veda oltre). Questa rete è composta da uno strato di input ed uno di output. I neuroni dei due strati sono completamente connessi tra loro, mentre i neuroni dello strato di output sono connessi in modo da essere organizzati su una linea o un piano (generalmente a matrice). In questo modo ogni neurone di output è connesso con un "vicinato" di neuroni e l'apprendimento segue una logica di tipo competitivo, con alcune particolarità che hanno reso la rete biologicamente plausibile ed adatta all'applicazione in molte aree come il controllo motorio e il riconoscimento del parlato. In poche parole, nell'apprendimento competitivo vengono modificati soltanto i pesi afferenti all'unità di output che risulta vincente. In questo caso invece i pesi afferenti ai neuroni del vicinato dell'unità di output vincente subiscono una modifica dei pesi in relazione ad una funzione a "cappello messicano".

Fondamentale per lo sviluppo e l'applicazione delle reti neurali è stato il lavoro

di Rumelhart, Hinton e Williams del 1986 [35] che propose un algoritmo di apprendimento per reti feed-forward (si veda oltre) multistrato che supera i limiti evidenziati da Minsky e Papert. Si trattava dell'algoritmo Back-Propagation, che di lì a breve si sarebbe presto affermato come il più utilizzato algoritmo di apprendimento per le reti neurali. Questo algoritmo modifica sistematicamente i pesi delle sinapsi, facendo in modo che la risposta della rete si avvicini sempre di più a quella desiderata fornita dall'utente, e si basa sulla tecnica della discesa a gradiente. Grazie a questo lavoro le attenzioni della comunità scientifica si rivolsero con rinnovato vigore verso le reti neurali, in quanto forniva un metodo facile da implementare in grado di fornire buone prestazioni. Vanno infine segnalati i lavori di Jordan (1986) [18] ed Elman (1990) [6] che hanno proposto delle varianti di una normale rete Feed-Forward, aggiungendo connessioni da uno strato superiore ad uno inferiore e auto-connessioni su uno o più nodi.

Queste reti sono chiamate reti a topologia ricorrente e sono utili in condizioni in cui la dinamica temporale contiene informazioni importanti, in quanto lo stesso input presentato in momenti diversi non produce necessariamente lo stesso output. Negli ultimi anni la ricerca ha portato allo sviluppo di nuovi algoritmi di apprendimento, tra i quali i più interessanti sono quelli che si basano sugli algoritmi genetici, e di nuovi strumenti di analisi per comprendere meglio il comportamento dei vari modelli di rete neurale. Molti ricercatori oggi fanno uso delle reti neurali nei rispettivi campi di ricerca, e sul mercato sono disponibili diverse applicazioni che si basano su di esse. Quest'ultimo fatto è dovuto anche all'attuale disponibilità di risorse di calcolo a basso costo, fenomeno che da diversi anni ha portato all'utilizzo di simulatori software di reti neurali, distogliendo l'interesse dalla loro realizzazione fisica.

1.6 Architettura e funzionamento black-box della rete

Neuroni del tipo visto in precedenza (*Processing Elements*, si veda 1.4) possono essere organizzati in differenti architetture per formare una rete neurale.

Una prima architettura propone i neuroni completamente connessi tra loro, in modo da determinare una struttura completamente connessa (ad esempio la macchina di Boltzmann vista in precedenza).

Un'altra architettura propone i neuroni raggruppati in diversi strati, concepiti come sottoinsiemi disgiunti e ordinati, a seconda della loro funzione: abbiamo quindi uno strato di input, uno o più strati nascosti (*hidden*) ed uno strato di output. Ad ogni strato sono connessi neuroni degli strati adiacenti ed, eventualmente, dello stesso strato.

Il modello più proposto ed esaminato prevede che ogni neurone sia connesso a tutti i neuroni degli strati adiacenti e che non ci siano connessioni tra neuroni dello stesso strato. I neuroni dello strato di input non hanno connessioni in ingresso e la loro attivazione consiste nel vettore (*pattern*) corrispondente ad un input del proble-

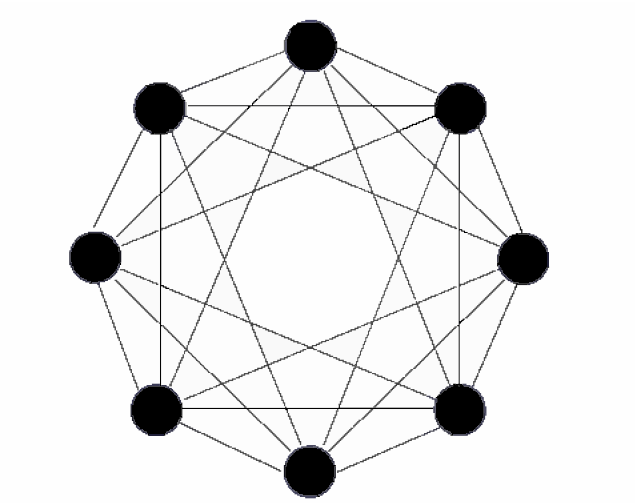

Figura 5. *Rete completamente connessa*

ma. Una funzione trasferisce il valore di attivazione senza eseguire calcoli ai neuroni dello strato nascosto, che calcolano la loro attivazione e la trasferiscono o ai neuroni di un altro strato nascosto oppure a quelli di output. L'attivazione di questi rappresenta l'output della rete. Il flusso informativo della rete è unidirezionale: i neuroni ricevono input solo dallo stato precedente e lo trasmettono solo a quello successivo e per questa caratteristica quest'architettura viene chiamata (*feed-forward*).

Esistono anche architetture a connessioni parziali, in cui ogni neurone è connesso solo con alcuni altri neuroni degli strati adiacenti.

Abbiamo già accennato alle reti ricorrenti: esse rappresentano una variante della struttura a strati appena descritta. Ne esistono diverse versioni, ma le più utilizzate sono la rete di Jordan e quella di Elman. Nella prima accanto a neuroni di input "tradizionali" sono presenti neuroni di stato; questi neuroni ricevono l'input direttamente dai neuroni di output della rete e presentano autoconnessioni. La rete di Elman invece consiste in una feed-forward a tre strati ai quali è aggiunto un ulteriore strato composto da unità (*context-units*) che ricevono input dai neuroni dello strato nascosto e trasferiscono i loro output a neuroni dello strato nascosto; anche queste unità presentano auto-connessioni.

Le altre architetture ricorrenti presenti in letteratura hanno le seguenti caratteristiche:

- la rete risultante dopo la cancellazione delle unità di contesto e dei pesi afferenti ad esse è una feed-forward senza auto-connessioni;

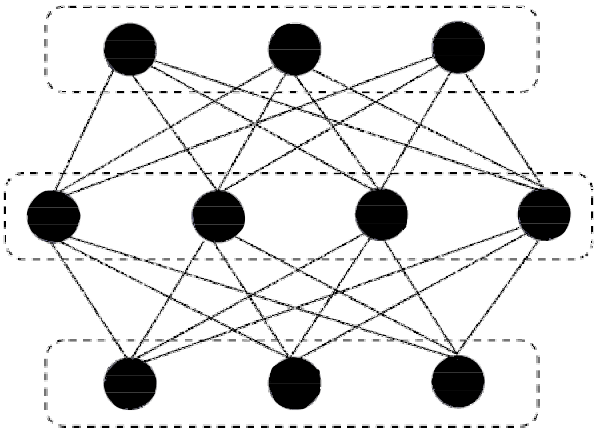

Figura 6. *Rete stratificata*

- i nodi di input non possono ricevere valori da altri nodi e quelli di output possono trasferire valori solo alle unità di contesto;
- ogni unità deve possedere almeno una connessione in entrata e per le unità di contesto questa proprietà è soddisfatta anche nel caso di una autoconnessione.

Il funzionamento di una rete neurale può a buon merito essere paragonato ad una scatola nera: essa riceve degli input che vengono trasformati in output, ma i neuroni intermedi sono nascosti nella scatola e la rete affronta il problema senza descrivere le modalità attraverso le quali perviene alla soluzione.

1.7 Apprendimento nelle reti neurali

L'apprendimento è definito come la capacità della rete di modificare il proprio comportamento in maniera da fornire le uscite giuste agli ingressi dati. La variazione del comportamento della rete avviene tramite la modifica di alcune sue componenti: abbiamo già accennato al fatto che la stragrande maggioranza degli algoritmi sottopone ad apprendimento le connessioni sinaptiche (*pesi*), per cui d'ora in poi ci riferiremo all'apprendimento come al procedimento che porta la rete ad assumere valori corretti.

Esistono tre principali tipologie di apprendimento: l'apprendimento supervisionato (supervised learning), l'apprendimento non supervisionato (*unsupervised learning*) e apprendimento con rinforzo (*reinforcement learning*).

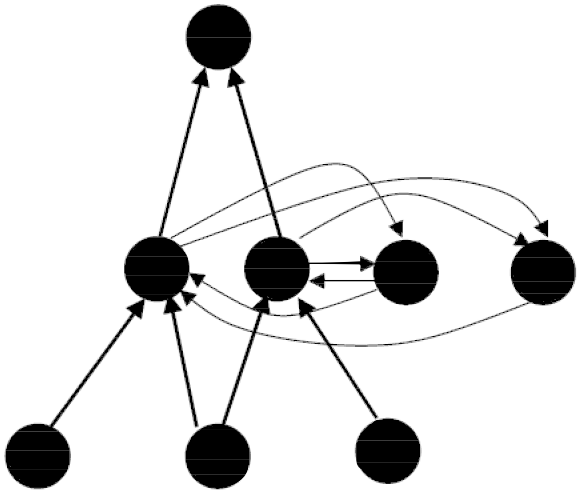

Figura 8. *Un esempio di rete ricorrente: rete di Elman*

gata quando non è possibile o risulta inutile fornire una coppia di dati in ingresso e uscita: non c'è insegnante che supervisiona il processo di apprendimento e non esistono esempi della funzione che dev'essere appresa dalla rete. L'apprendimento avviene attraverso una continua interazione con l'ambiente in modo da minimizzare una funzione *cost-to-go*, definita come l'aspettativa del costo cumulativo di una serie di azioni eseguite in sequenza. Esiste un critico che genera un segnale di rinforzo (*heuristic reinforcement signal*) a partire da segnali di rinforzo ricevuti dall'ambiente. Il sistema osserva una sequenza di segnali ricevuti dall'ambiente, gli stessi che sono utilizzati dal critico per produrre il segnale di rinforzo, in modo che questo venga interpretato come un segnale positivo o negativo sul comportamento così da poter aggiustare i parametri di conseguenza.

In questo schema può accadere che alcune azioni eseguite all'inizio della sequenza risultino determinanti sul comportamento complessivo; in relazione a questo problema un'altra componente del sistema, la *learning machine*, ha il compito di scoprire queste azioni. E' proprio la learning machine che dev'essere in grado di assegnare crediti e punizioni ad ogni azione nella sequenza, mentre il segnale di rinforzo serve soltanto a dare una valutazione del risultato globale.

Questa tipologia di a apprendimento è fortemente legata alla programmazione dinamica, che fornisce un modello matematico per i processi decisionali aventi ad oggetto sequenze di azioni ed è alla base dei modelli che studiano l'interazione di un sistema con l'ambiente.

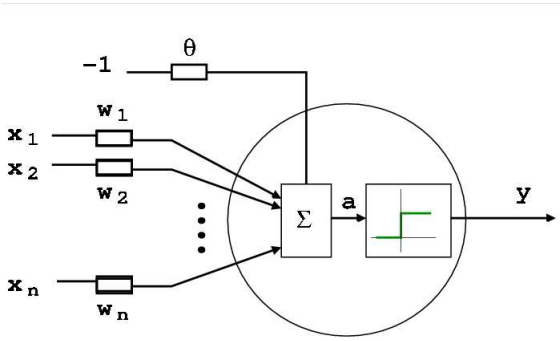

Figura 9. *Il percettrone*

1.8 Principali modelli teorici

1.8.1 Il Percettrone

Il percettrone è un modello di neurone proposto da Rosenblatt nel 1962, caratterizzato da ingressi binari, uscita binaria e funzione di attivazione di Heaviside. Tale neurone costituisce l'esempio più semplice di rete neurale ed ha trovato subito applicazione nel riconoscimento di forme offrendo dei buoni risultati. La differenza principale con il neurone di McCulloch e Pitts (si veda la sezione 1.4) sta nel fatto che i pesi sono modificabili in funzione dell'associazione input-output che si desidera ottenere dalla rete.

Il limite del percettrone è però che non riesce a risolvere problemi caratterizzati da ingressi non linearmente separabili: un problema si dice linearmente separabile se, disposti gli ingressi in uno spazio, è possibile determinare un iperpiano che divida nettamente la classe degli ingressi che portano a soluzione positiva da quelli che portano a soluzione negativa.

E' utile descrivere il funzionamento pensando ad un problema di classificazione a 2 input x_1 ed x_2. Il percettrone riceve il pattern in ingresso e pondera i suoi componenti tramite i pesi delle connessioni per determinare se il pattern appartiene o meno ad una determinata classe. Date le connessioni w_1 e w_2 è possibile separare il piano di riferimento con la retta

$$w_1 x_1 + w_2 x_2 = \theta$$

che separa il piano in due semipiani per i quali si ha:

$$w_1 x_1 + w_2 x_2 > \theta$$

e quindi $y = 1$ nel semipiano a destra della retta;

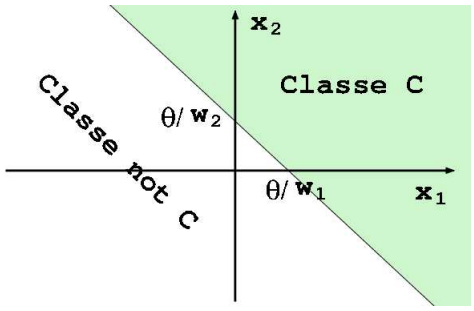

Figura 10. *Problemi linearmente separabili*

$$w_1 x_1 + w_2 x_2 \leq \theta$$

e quindi $y = 0$ nel semipiano a sinistra della retta.

E' utile introdurre in questo schema un neurone fittizio con attivazione pari a -1 e sinapsi pari al valore della soglia. L'addestramento del percettrone avviene in modo supervisionato: esiste un supervisore in grado di determinare se l'uscita è giusta o sbagliata e la rete è in grado di determinare l'errore, dato dalla differenza tra valore desiderato e valore effettivo. Se l'uscita desiderata e quella effettiva non coincidono le sinapsi verranno modificate in base alla regola[3]

$$w_i(t+1) = w_i(t) + \eta \delta x_i$$

dove

$w_i(t+1)$ è il peso della sinapsi i dopo la modifica;
$w_i(t)$ è il peso della sinapsi i prima della modifica;
η è il coefficiente di apprendimento (*learning rate*);
x_i è il valore trasmesso dalla sinapsi i;
δ è la differenza $(d - y)$tra output desiderato d ed output effettivo y.

Il significato di questa formula può essere facilmente compreso: se il neurone riceve in ingresso un pattern non appartenente alla classe oggetto dell'applicazione può restituire in uscita il valore 0, ed in questo caso l'uscita è corretta, oppure 1. In questo secondo caso si verifica che il valore dela sommatoria $w_1 x_1 + w_2 x_2$ è troppo grande, quindi dev'essere diminuita modificando il valore delle sinapsi, che risulteranno avere un nuovo valore inferiore a quello precedente. Al contrario se il

[3]Questa regola è chiamata *delta-rule*.

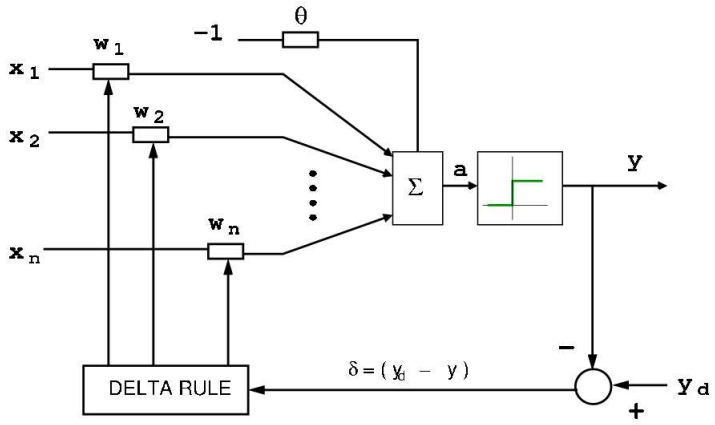

Figura 11. *Delta rule*

neurone riceve in ingresso un pattern appartenente alla classe può restituire in uscita il valore 1, ed in questo caso l'uscita è corretta, oppure 0. In questo secondo caso si verifica che il valore la sommatoria $w_1x_1 + w_2x_2$ è troppo piccola, quindi dev'essere aumentata modificando il valore delle sinapsi, che risulteranno avere un nuovo valore superiore a quello precedente. In base a quanto appena detto la modifica delle sinapsi avviene solo in caso di errata classificazione del pattern e non ha luogo se l'output del neurone pre-sinaptico è nullo. In questo esempio l'effetto dell'apprendimento è quello di modificare l'inclinazione della retta che divide il piano in due fino a realizzare la separazione. Gli algoritmi che sottopongono anche la soglia θ ad apprendimento determinano uno spostamento parallelo della retta. Il parametro η determina la velocità di apprendimento della rete: valori elevati portano a grandi modifiche delle sinapsi ad ogni passo con possibile instabilità dell'apprendimento, valori bassi portano a modifiche piccole. L'algoritmo di apprendimento può essere schematizzato come segue:

- si determina un insieme di coppie (X,D) di esempi disponibili;
- si inizializzano i pesi w con valori casuali;
- si presenta una coppia (x, d);
- si calcola la risposta y della rete;
- se il risultato effettivo y e quello desiderato D non coincidono le sinapsi vengono modificate in base alla delta rule, altrimenti rimangono inalterate;

- si presenta una nuova coppia e si procede come sopra, fino all'esaurimento degli esempi disponibili.

La delta rule fu poi generalizzata da Widrow e Hoff [41] nel 1960 che la applicarono a reti bi-strato composte da *ADALINE* (*ADAptive LINEar, ADAptive LInear NEuron, ADAptive LINear Elements*) con più unità di output, in grado così di discriminare diverse classi linearmente separabili: ad ogni unità di output corrisponde una classe e quando si presenta un pattern appartenente ad una classe k la rete determina l'attivazione del nodo o_k corrispondente a tale classe. I due autori hanno il merito di essere stati i primi a introdurre il concetto di errore, determinando che la variazione dei pesi è proporzionale al gradiente dell'errore. Si definisce *errore quadratico assoluto* relativo ad un pattern in ingresso la sommatoria dei quadrati delle differenze tra output effettivo e desiderato relative ad ogni nodo di output della rete:

$$E = \frac{1}{2}\sum_{o=1}^{m}(d_o - y_o)^2$$

Questo può essere minimizzato modificando i pesi delle connessioni. In particolare, se l'errore cresce all'aumentare dei pesi w, la derivata di E rispetto a w è di segno positivo. In questo caso i pesi devono essere diminuiti. Viceversa, se l'errore decresce all'aumentare dei pesi w, la derivata di E rispetto a w è di segno negativo. In questo caso i pesi devono essere aumentati. Più precisamente la regola di Widrow-Hoff dice che la variazione del peso dev'essere uguale all'inverso della derivata dell'errore rispetto al peso moltiplicata per il coefficiente d'apprendimento:

$$\Delta w_{ij} = -\eta\frac{\partial E}{\partial w_{ij}}$$

Applicando le regole della derivazione formale abbiamo che

$$\frac{\partial E}{\partial w_{ij}} = (y_j - d_j)\frac{\partial y_j}{\partial w_{ij}} = (y_j - d_j)\frac{\partial y_j}{\partial net_j}\frac{\partial net_j}{\partial w_{ij}}$$

Dato che

$$\frac{\partial y_j}{\partial net_j} = f'(net_j) \quad e \quad \frac{\partial net_j}{\partial w_{ij}} = x_i$$

si avrà

$$\Delta w_{ij} = -\eta(y_j - d_j)f'(net_j)x_i$$

e posto

$$\delta_j = (y_j - d_j)f'(net_j)$$

la formula finale sarà

$$\Delta w_{ij} = -\eta\delta_j x_i$$

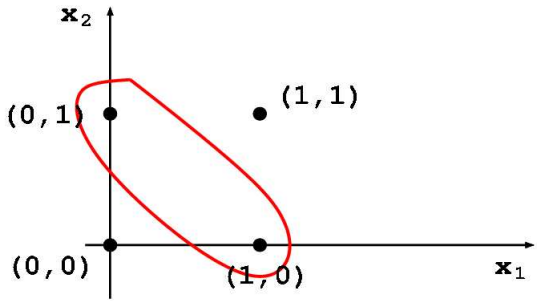

Figura 12. *Il problema dello XOR*

Questo meccanismo può essere interpretato nel seguente modo: l'errore è una funzione quadratica dei pesi della rete, e l'algoritmo di apprendimento esegue una discesa lungo la linea di massima pendenza della funzione a partire dal punto generico determinato dai pesi iniziali scelti casualmente. Il coefficiente di apprendimento rappresenta la lunghezza del passo di questa discesa. Da questa regola presero spunto Rumelhart ed altri per sviluppare l'algoritmo Back-Propagation.

1.8.2 Back-Propagation

Abbiamo visto che il Perceptron è in grado di fornire buone prestazioni ma non riesce a trattare problemi non linearmente separabili. L'esempio classico che viene presentato per spiegare questo problema è il caso della funzione XOR (or esclusivo): non è possibile disegnare sul piano una retta in modo da dividere lo spazio delle soluzioni in due categorie.

Questo problema, insieme agli altri non separabili linearmente, può essere risolto utilizzando una architettura feed-forward multi-strato che possegga uno o più strati nascosti. Questo tipo di rete è conosciuto da circa quarant'anni, e fu inizialmente chiamato MLP (*Multi-Layer-Perceptron*, termine che, sebbene ancora utilizzato in letteratura, risulta oggi improprio) ma il suo utilizzo ha trovato dei limiti nel fatto che i percettroni sono in grado di risolvere soltanto problemi linearmente separabili e nell'assenza di algoritmi di apprendimento idonei. Il primo dei due punti è già stato dimostrato in precedenza. In relazione al secondo problema, l'introduzione di strati nascosti ha impedito l'applicazione dell'algoritmo di apprendimento Widrow-Hoff perché si basa sulla modifica dei pesi in base allo scostamento tra output effettivo e desiderato per unità; dato che non si è in grado di determinare l'output desiderato delle unità intermedie, l'applicazione dell'algoritmo a questo tipo di rete risulta impossibile.

Nel 1986 Rumelhart, Hinton e Williams [35] proposero l'algoritmo Back-Propagation,

che si basa sulla propagazione all'indietro dell'errore, dalle unità di output fino a quelle di input. Il funzionamento di questo algoritmo avviene in due fasi: nella prima l'attivazione delle unità di input viene propagata in avanti tramite le funzioni di attivazione (*forward phase*), nella seconda fase vengono modificati i pesi delle connessioni tramite la tecnica della discesa del gradiente (*backward phase*), con la quale l'errore delle unità di output viene propagato all'indietro (da cui il nome dell'algoritmo) fino alle unità di input. L'errore delle unità di output è calcolato nel modo consueto, mentre quello delle unità nascoste, e qui sta la maggiore introduzione di questo algoritmo, è calcolato moltiplicando l'errore delle unità di output, ponderato dal relativo peso della connessione, per la derivata prima della funzione di output di ogni unità nascosta, .

$$\delta_j = \begin{cases} f'(net(j))(y_j - d_j) & \text{se j è unità di output} \\ f'(net(j)) \sum_k \delta_k w_{jk} & \text{se j è unità nascosta} \end{cases}$$

$$\Delta w_{ij} = -\eta \delta_j x_i$$

In questo modo i pesi sono modificati in relazione al loro contributo all'errore su quell'esempio. Infatti sappiamo che

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial net(j)} \frac{\partial net(j)}{\partial w_{ij}}$$

$$\frac{\partial y_j}{\partial net(j)} = f'(net(j))$$

$$\frac{\partial net(j)}{\partial w_{ij}} = x_i$$

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j} f'(net(j)) x_i$$

Per ottenere $\frac{\delta E}{\delta w_{ij}}$ dovremmo conoscere $\frac{\delta E}{\delta y_j}$, ma gli unici errori che conosciamo sono quelli relativi alle unità di output. Per cui il passo successivo sarà esprimere gli errori relativi ai nodi intermedi in funzione degli errori delle unità di output

$$\frac{\partial E}{\partial w_{ij}} = f'(net_j) x_i \sum_k (\delta_k w_{jk})$$

da cui

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} = -\eta \delta_j x_i$$

Questo algoritmo può essere applicato in maniera ricorsiva a reti con un qualsiasi numero di strati, anche se è dimostrato che reti con più di quattro strati non portano ad aumenti nella potenza computazionale. Particolarmente importante è la scelta della funzione di attivazione: la più utilizzata è la sigmoide, ma non esistono indicazioni codificate in tale senso. Unica regola è di non adottare funzioni di attivazione lineari su tutti i neuroni, in quanto l'utilizzo di queste funzioni ricondurrebbe una rete con un qualsiasi numero di strati al funzionamento di una rete a due strati, dato che la combinazione di funzioni lineari è essa stessa una funzione lineare.

In una rete multistrato ciascuna delle unità nascoste individua un iperpiano in grado di separare lo spazio dei pattern in due classi distinte. La combinazione di questi iperpiani permette di svolgere compiti di classificazione anche piuttosto difficili con buoni risultati nella maggior parte dei casi; si può pensare che ogni neurone nascosto può essere addestrato ad attivarsi in presenza di una determinata caratteristica nel pattern in ingresso, e questa osservazione si è rivelata utile per modellare caratteristiche neurofisiologiche e processi cognitivi. Nella regola proposta assume una particolare importanza il termine η, coefficiente di apprendimento che determina, come negli algoritmi precedenti, la velocità dell'apprendimento della rete. Un tasso di apprendimento ridotto può portare a tempi di addestramento eccessivamente lunghi, mentre un tasso grande può portare la rete a comportamenti oscillatori intorno al punto di minimo della funzione di errore. Un buon compromesso può essere quello di iniziare l'apprendimento con valori di questo coefficiente alti, per accelerare la convergenza all'inizio del processo, per poi ridurlo gradualmente, in modo da evitare oscillazioni alla fine dell'apprendimento. Un miglioramento di questa formula sta nell'aggiungere un termine, detto *momentum*, che rappresenta una proporzione dell'ultima modifica apportata al peso:

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} + \beta w_{ij}$$

Questa variante fu proposta già da Rumelhart nell'articolo in cui veniva proposto per la prima volta il Back-Propagation [35]. Possiamo pensare all'errore globale come una funzione dei pesi sinaptici (come descritto in precedenza), caratterizzata da un andamento molto irregolare a causa della non linearità dell'attivazione delle unità di output. L'algoritmo Back-Propagation esegue una ricerca del minimo in questa superficie, diminuendo ad ogni iterazione il valore dell'errore. In questo senso l'aggiunta del termine di momentum fa sì che la superficie dell'errore sia attraversata rapidamente con pochi passi in presenza di *plateaux*, mentre la dimensione dei passi diminuisca quando la superficie diventa irregolare. In questo modo può essere stabilito anche un coefficiente di apprendimento alto, ma la rete non corre il rischio di pesanti oscillazioni in quanto recupera una buona porzione dell'ultimo peso raggiunto.

Abbiamo in precedenza detto che ci saremmo occupati di descrivere il processo

di apprendimento supervisionato in relazione all'algoritmo di Back-Propagation. In questo tipo di apprendimento ad ogni iterazione viene fornita alla rete una coppia di campioni

$$X = (x_1 \dots x_n) \qquad D = (d_1 \dots d_n)$$

che rappresentano l'input e la risposta che si desidera venga prodotta dalla rete per quel determinato input. Quando alla rete viene fornito un input, il potenziale post-sinaptico dei neuroni di input verrà propagato, mediante le funzioni di attivazione e trasferimento, fino a quelli di output, che determineranno un valore di output y. In base questo le sinapsi verranno modificate in modo da minimizzare qualche misura dello scostamento degli output ottenuti (y) da quelli desiderati forniti dall'utente (d), ad esempio la sommatoria degli scostamenti (errori) al quadrato SSE.

L'apprendimento avviene su un insieme di campioni (X, D) detto *training set*. La presentazione di tutti gli elementi appartenenti all'insieme è chiamata epoca, e di solito è necessaria più di una epoca affinchè l'addestramento sia concluso. Una volta addestrata, la rete viene congelata (i pesi cioè diventano immodificabili) e può iniziare l'effettivo utilizzo della stessa su un insieme di dati non visti in precedenza, insieme ai quali non è quindi presentata nessuna uscita desiderata, il *generalisation set*. Abbiamo già visto come una delle principali potenzialità delle reti neurali è la capacità di generalizzare, cioè di operare su dati non visti in precedenza (che abbiano però le stesse caratteristiche) e fornire in relazione agli stessi dei buoni risultati. In questa ottica, minimizzare (o nel caso estremo annullare) l'errore sul *training set* potrebbe portare ad una rete che in fase di utilizzo riesca a fornire l'output desiderato per i dati incontrati durante l'apprendimento ma il cui comportamento su dati mai visti in precedenza sia caratterizzato da errori notevoli. E' questo il fenomeno dell'*over-fitting* o *over-training*, che sta ad indicare la mancanza della capacità di generalizzazione. Un modo per risolvere questo problema è quello di creare un altro insieme di campioni *(X,D)* chiamato *validation set*, usato per valutare la performance della rete su dati non usati per l'apprendimento. In pratica i dati appartenenti al training set vengono utilizzati per modificare i pesi sinaptici, ed ogni n iterazioni (n viene definito dall'utente) viene presentato alla rete un elemento del *validation set* soltanto per valutare lo scostamento dell'output dall'uscita desiderata, senza cioè che avvengano modifiche dei pesi.

Di solito si ha a disposizione un insieme di campioni (X,D) che viene partizionato in training set e validation set. Non esistono regole su come effettuare questa partizione e sul rapporto tra i due insiemi, ma generalmente si fa in modo che il numero di elementi presenti nel validation set sia circa un terzo di quelli presenti nel training set. La minimizzazione dell'errore deve avvenire sul validation set. Come abbiamo modo di notare dalla figura, l'errore sul validation set presenta un minimo assoluto, oltre il quale riprende a salire, mentre l'errore sul training set continua a decrescere.

E' questo il punto in cui l'apprendimento dovrebbe essere fermato, in quanto rappresenta il punto in cui la capacità di generalizzazione della rete è massima. Que-

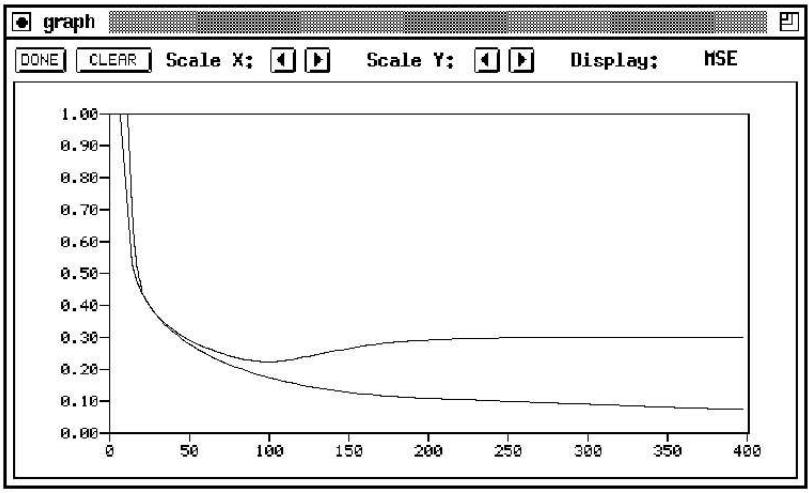

Figura 13. *Evoluzione dell´apprendimento*

sto sembra essere vero soltanto in via teorica, in quanto la scelta del validation set è arbitraria, e diversi validation set hanno funzioni di errore differenti. In passato si poteva argomentare l'incapacità di determinare quest'ipotetico punto di stop-learning in quanto la presenza di outliers oppure lo shuffling (si veda oltre) potrebbero conferire un'andamento della funzione di errore in cui siano presenti minimi locali. Queste preoccupazioni oggi possono essere evitate sia perché sono state sviluppate delle euristiche per evitare il problema dei minimi locali, sia perché gli strumenti di simulazione software permettono in ogni momento di congelare la rete e di salvarne una copia in memoria, rendendo estremamente semplice il ripristino di una configurazione precedente. I problemi che invece ancora oggi devono essere affrontati sono che tale punto non è determinabile a priori, il che non assicura la convergenza in un tempo determinato, oppure che la funzione di errore può essere caratterizzata da plateaux, che potrebbero rendere difficile determinare se il minimo è stato raggiunto. Per far fronte a questi problemi si utilizzano dei criteri di stop-learning, in cui si determina la terminazione della fase di apprendimento al verificarsi di una condizione, che può essere il raggiungimento di una soglia di errore determinata oppure di un numero di iterazioni (o epoche) senza alcun miglioramento nell'errore. Da queste regole si deduce la priorità che viene data al raggiungimento di un certo livello di generalizzabilità della rete rispetto al raggiungimento del minimo della funzione dell'errore. Un importante e semplice strumento per evitare l'over-fitting è quello di non presentare i pattern sempre nello stesso ordine temporale al variare delle epoche, usando la tecnica di *shuffling* (rimescolamento). Questo perché la rete neurale potrebbe estrapolare delle relazioni dovute esclusivamente all'ordine con cui i pattern vengono presentati,

che porterebbero ad un comportamento oscillatorio.[4]

In relazione alle modalità di aggiornamento dei pesi definiamo due tipi di algoritmo : *on-line Back-Propagation* ed *off-line Back-Propagation*.

Nella versione on-line le modifiche delle sinapsi avvengono dopo che alla rete viene presentata una coppia (x, d): si presenta alla rete un valore (o meglio, un vettore di valori) di input, si calcola l'output della rete, lo si confronta con l'uscita desiderata e si modificano le sinapsi della rete in funzione di una metrica dell'errore. Viene poi presentato un altro pattern e si continua seguendo lo stesso schema.

Nella modalità off-line (batch) invece, le modifiche dei pesi avvengono dopo aver presentato tutte le coppie che formano un'epoca: è eseguita una modifica cumulativa di tutte la variazioni di peso che sono state calcolate ad ogni presentazione della coppia. Questo tipo di apprendimento è utile in caso di implementazioni parallele caratterizzate da alti costi di comunicazione. Ad un livello intermedio tra questi due modelli si colloca il *chunkwise Back-Propagation*, in cui si definisce il numero di pattern (definito appunto chunk) da presentare alla rete prima di modificare i pesi. Questo tipo di apprendimento è utile per problemi con training set troppo ampio, in cui la versione off-line presenta tempi di convergenza troppo lunghi e quella on-line learning porta ad un comportamento instabile della rete. Osservazioni empiriche hanno mostrato che la modalità on-line converge in meno iterazioni rispetto alla off-line (chiamata back-propagation store, in quanto le modifiche da apportare cumulativamente ai pesi vengono salvati-immagazzinati in un file specifico). Tuttavia la back-propagation store sembra avere maggiori probabilità di convergere verso la configurazione ottimale, perché evita continui cambi di segno nell'aggiornamento dei pesi, riducendo la rilevanza degli outlier. Questo aspetto a favore della versione store perde di importanza quando si usa la versione on-line con momentum, in quanto la differenza è sempre meno evidente all'aumentare del termine di momentum. Tra i sistemi utilizzati per migliorare le performance dell'algoritmo di Back-Propagation vanno segnalati la *riduzione* e il *decadimento* dei pesi. Entrambi si basano su una osservazione di Rumelhart (riportata in Hanson e Pratt [12]): egli dice che in riferimento ad un insieme di dati, la rete più semplice e robusta è quella che, in media porta alla migliore generalizzazione rispetto alla popolazione da cui i dati sono stati estratti. La versione più semplice della *riduzione* consiste nel forzare periodicamente le sinapsi il cui valore risulta, in valore assoluto, inferiore ad una soglia ad assumere valore nullo. Questa soglia è generalmente definita come una proporzione del peso più piccolo all'interno della rete, ma può anche essere definita in relazione al peso più grande o alla media dei pesi. I pesi con valore nullo possono considerarsi eli-

[4]Nella figura 13 e in quelle relative alla fase sperimentale sull'asse delle ascisse sono riportate le epoche dell'apprendimento e sull'asse delle ordinate è riportata la misura dell'errore (nei nostri esperimenti utilizziamo MSE).

minati dalla rete. Il *decadimento* dei pesi invece aggiunge un termine all'errore che rappresenta una misura della complessità della rete

$$E = \frac{1}{2}\sum_{i=j}^{n}\sum_{k=1}^{r}(y_k - d_k)^2 + \lambda\sum_{ij}\frac{w_{ij}^2}{1+w_{ij}^2}$$

dato che l'obiettivo è minimizzare la complessità, un costo verrà associato ad ogni connessione. Questo risulta in una modifica della regola di apprendimento

$$\Delta w_{ij} = -\eta\frac{\partial E}{\partial w_{ij}} + \beta\Delta w_{ij} - \lambda\sum_{ij}\frac{2w_{ij}}{(1+w_{ij}^2)^2}$$

che risulta essere però molto sensibile al valore scelto per λ, in quanto un valore piccolo non ha effetti, mentre uno troppo grande porterà tutti i pesi ad assumere valore nullo. Una soluzione è quella di modificare tale valore con il procedere dell'apprendimento; a tale proposito sono state introdotte alcune regole interessanti (riportate in [5]). Oppure si può, più semplicemente, ridurre il peso di una proporzione del vecchio valore, come proposto da Werbos [40]. I pesi sono portati ad assumere valore nullo se non rinforzati dalla Back-Propagation.

$$\Delta w_{ij}(t+1) = \eta\delta_j x_i - \alpha w_{ij}(t)$$

L'algoritmo Back Propagation presenta alcuni problemi: innanzitutto, sebbene l'errore diminuisca con il proseguire delle iterazioni, non esiste nessun teorema che ne garantisca la convergenza, cioè il raggiungimento del minimo assoluto della funzione di errore, per cui non è garantito che la rete raggiunga la configurazione ottimale dei pesi. La funzione di errore presenta un andamento molto accidentato, caratterizzato da improvvisi cambi di pendenza, strette gole ed ampie vallate e per questo è probabile che l'algoritmo si diriga verso un minimo locale e vi rimanga incastrato. Sono state sviluppate diverse soluzioni per ovviare a questo problema, la più semplice delle quali consiste nell'aggiungere periodicamente del rumore casuale ai pesi in modo da provocare una certa instabilità della rete. Bisogna stare però attenti ad usare questa opportunità con estrema cautela per non provocare comportamenti fuorvianti e totalmente instabili. Altro problema è che inizializzando tutti i pesi allo stesso valore, le modifiche effettuate saranno uguali per tutte le sinapsi che portano allo stesso ramo di uscita, per cui è contemplata convenzionalmente una inizializzazione casuale dei pesi tra un limite minimo ed un massimo definiti dall'utente. Inoltre, come l'algoritmo di Widrow-Hoff, l'aggiornamento dei pesi non avviene se l'output dell'unità pre-sinaptica è nullo. Non ultima è la non-plausibilità biologica, determinata dall'utilizzo della funzione di errore e da presenza di connessioni che si comportano in maniera simmetrica in quanto trasmettono 2 flussi di informazione: l'attivazione nella prima fase e l'errore nella seconda. Fortunatamente, come già accennato, la

non plausibilità biologica non ha compromesso la ricerca in tale direzione, ed oggi possiamo dire che il Back-Propagation ha portato rinnovato interesse a questo campo (il che significa anche maggiori finanziamenti) e risultati applicativi notevoli.

Non particolarmente interessante, se non dal punto di vista teorico, appare la variante in cui lo stesso pattern viene presentato più di una volta di seguito prima di passare al pattern successivo, e le modifiche dei pesi avvengono dopo ogni singola presentazione. Il problema di questa tecnica, chiamata Back-Propagation *sequenziale*, è che la rete dimentica il pattern memorizzato quando si passa all'addestramento del pattern successivo; la funzione di errore presenta un andamento irregolare, in quanto questo diminuisce ad ogni iterazione sullo stesso pattern, per poi schizzare subito verso l'alto alla presentazione del pattern successivo.

1.8.3 Reti di Ward

Questo modello di rete è simile all'architettura feed-forward/Backpropagation, ma presenta la caratteristica di possedere, all'interno dello strato nascosto, diversi insiemi di neuroni, che generalmente possiedono una diversa funzione di attivazione a seconda dell'insieme a cui appartengono: questo è molto utile se si vogliono scoprire diverse caratteristiche nei pattern in ingresso. Considerando che tali insiemi di neuroni possono essere disposti su più di uno strato e che sono possibili connessioni dirette tra strato di input e strato di output, è possibile classificare diverse tipologie di reti di Ward. A proposito delle connessioni dirette tra unità di input e di output, c'è chi ritiene che queste possano aumentare il potere di riconoscimento di caratteristiche della rete. Alcuni autori hanno studiato questo aspetto in riferimento alla previsione one-step-ahead (si veda oltre) senza però arrivare a conclusioni soddisfacenti.

1.8.4 Reti Jump-Connection

Si tratta di una variante della classica architettura feed-forward in cui ogni strato di neuroni presenta connessioni con tutti gli strati successivi. Può essere addestrata senza problemi con un algoritmo di back propagation, ma la grande quantità di connessioni che presenta porta a tempi di addestramento notevolmente lunghi, anche se riesce ad ottenere risultati migliori.

1.8.5 Reti General-Regression

Le reti General Regression furono ideate nel 1991 da Donald Specht [37] e si sono rivelate particolarmente utili per l'approssimazione di funzioni continue, in quanto sono in grado di adattarsi a superfici multi-dimensionali. Hanno la caratteristica di possedere un solo strato nascosto, con un neurone nascosto per ogni elemento del training set, e necessitano di una sola epoca di apprendimento.

In questo tipo di reti non c'è bisogno di determinare il coefficiente di apprendimento ed il momentum, ma è necessario definire lo *smoothing factor*, un parametro che indica la capacità della rete di approssimare i dati di ingresso. L' apprendimento avviene misurando la distanza tra le osservazioni del training set ed un campione definito di dati. Un algoritmo genetico serve a determinare lo smoothing factor per ciascun input; dopo una prima fase di apprendimento la rete svolge una *sensitive analysis* per attribuire maggiore importanza agli input con smoothing factor più elevato. Tale analisi viene estesa alle variabili, in modo da poter eliminare quelle con smoothing factor basso. L'utilizzo avviene poi confrontando il pattern in ingresso con tutti i pattern del training set. L'output è ricavato da una quota proporzionale dei pattern della rete, data dalla distanza tra i neuroni dell'input e gli altri presenti nel training set. I risultati ottenuti risultano migliori di quelli ottenibili tramite Back-Propagation in diverse applicazioni, tuttavia il grande numero di neuroni nascosti di cui necessitano può portare ad un tempo di addestramento eccessivo, e questo problema ne ha compromesso fortemente l'utilizzo.

1.9 Vantaggi e svantaggi delle reti neurali

In passato le reti neurali sono state viste come degli strumenti in grado di effettuare compiti complessi con risultati stupefacenti. Infatti è stato dimostrato (si veda [17]) che una rete neurale feed-forward con un numero sufficientemente alto di neuroni nascosti e con apprendimento back-propagation è in grado di approssimare qualsiasi funzione e questa capacità di essere usate come approssimatori universali ha sempre attirato l'interesse dei ricercatori dei campi più diversi. Tuttavia, anche le reti neurali presentano pregi e limitazioni.

La capacità maggiore delle reti neurali è la proprietà di generalizzazione, cioè la capacità di operare su dati mai incontrati in precedenza, ma che hanno le caratteristiche dei dati usati per l'addestramento. E' questa la caratteristica più richiesta alle reti neurali, specie per quanto riguarda i problemi di previsione, che si rivela utile quando si ha a che fare con dati incompleti o affetti da rumore. Inoltre quando si opera con le reti neurali non si ha necessità di particolari ipotesi a priori circa le relazioni tra variabili e la distribuzione dei dati: si lascia che sia la rete stessa ad inferire la relazione eventualmente esistente tra le variabili di input e quelle di output, al contrario di quei modelli identificati come *model-driven*, rispetto ai quali le reti neurali stanno guadagnando una forte competitività. Le reti apprendono le relazioni esistenti nei dati che vengono loro presentati: imparano dall'esperienza, dall'osservazione e l'elaborazione dei dati (per questo vengono definite *data-driven*); anche quando un insieme di dati sembra avere degli input irrilevanti le reti possono essere in grado di ricavarne informazioni interessanti. Ciò è particolarmente utile quando non è presente una grande conoscenza a priori del fenomeno che si sta studiando.

A questo proposito si ritiene che le reti siano particolarmente adatte per catturare delle relazioni non lineari tra un ampio numero di variabili.

Tutto ciò fino ad ora si è dimostrato particolarmente utile per quanto riguarda diverse applicazioni economico-finanziarie in quanto ricerche empiriche hanno identificato delle interdipendenze tra variabili, ma non sono state in grado di fornire dei modelli per esse. Si è rivelata particolarmente apprezzata la capacità delle reti neurali ad operare come memorie associative. Con ciò si intende dire che la rete è in grado di risalire ad un pattern anche quando le viene presentata una versione parziale oppure corrotta di esso, e questo si è dimostrato utile ad esempio nel riconoscimento del parlato o di immagini. Una rete neurale può inoltre trattare con successo dati non numerici, quale ad esempio la posizione geografica.

Altra caratteristica è la robustezza verso il malfunzionamento tipica dei sistemi connessionisti: il malfunzionamento di qualche componente della rete non ne pregiudica il comportamento complessivo; a ciò va aggiunta una buona tolleranza verso il rumore, in quanto se questo viene aggiunto in maniera crescente ai dati in input, si nota soltanto un lento peggioramento della qualità dell'output (*gently degradation*). Questo è dovuto alla computazione distribuita attraverso le sinapsi. Inoltre l'elaborazione distribuita e parallela tipica delle reti neurali, ed il conseguente parallelismo implicito negli algoritmi di apprendimento, fanno sì che esse possano essere implementate su macchine parallele, con un sensibile miglioramento nelle performance per quanto riguarda il tempo di apprendimento. Infatti il tempo di apprendimento è cruciale specie per reti con un numero elevato di neuroni nascosti e sinapsi e con insiemi ampi di dati di input. Ciò ha portato ad esempio alla scarsa diffusione delle reti General Regression, che permettono dei risultati molto accurati, ma sono caratterizzate da un tempo di addestramento notoriamente lungo. Tra diversi metodi studiati per ridurre il tempo di addestramento, è interessante la combinazione di reti neurali e logica fuzzy ([22] [31]).

Le reti neurali non sono in grado di spiegare i risultati raggiunti in sede di addestramento e non spiegano le relazioni che generano i fenomeni. Questo deriva direttamente dalla rappresentazione sub-simbolica della conoscenza nei sistemi connessionisti, in cui la conoscenza acquisita non è memorizzata in modo simbolico, ma risulta distrribuita all'interno del sistema ed è un grave problema per chi vuole conoscere il ragionamento che porta ad ottenere i risultati finali. Studi sono stati effettuati anche per estrarre regole dalle reti neurali, senza portare però a risultati apprezzabili [38]. Manca poi una teoria riguardante il funzionamento delle reti, e ciò ha portato i ricercatori a definire una propria arte basata sull'esperienza e sul *try-and-error* nello sviluppo della rete, specie per quanto riguarda la definizione delle variabili di input e la determinazioni della topologia e dei parametri. Proprio quest'ultimo problema sembra essere il più gravoso: la scelta dei parametri (in particolare dei neuroni nascosti e dell'insieme di addestramento) si rivela cruciale in relazione al problema dell'over-fitting. Con questo termine si vuole identificare la

scarsa capacità di generalizzazione che può sopravvenire quando si progetta e si addestra la rete con il risultato che questa è in grado di operare benissimo sui dati che le sono stati presentati in fase di addestramento, mentre presenta un comportamento decisamente insufficiente su dati estranei. Tipicamente questo può avvenire quando, in una rete *feed-forward*, si utilizzano troppi neuroni nascosti, oppure quando l'insieme e il tempo di addestramento sono troppo grandi. Sono state sviluppate alcune tecniche per risolvere questo problema, ma ancora oggi le soluzioni migliori usate dai ricercatori sono dettate dall'esperienza e dall'approccio try-and-error.

2 Le reti neurali ed il rischio di credito

Come abbiamo avuto modo di osservare, le reti neurali sono state teorizzate verso la metà del novecento; tuttavia si è dovuto attendere il finire degli anni '80 del secolo concluso affinchè il mondo scientifico iniziasse a ricorrere al loro utilizzo con interesse teorico e applicazioni pratiche. Le reti neurali hanno iniziato a trovare applicazione nei campi più disparati: meccanica, robotica, applicazioni militari, diagnosi mediche, riconoscimento di immagini, eliminazione del rumore di fondo da registrazioni audio, controllo dei processi. L'utilizzo nell'area economico finanziaria è un fenomeno ancora più recente, ma oggi le reti neurali trovano applicazione in più di 90 aree dell'economia e della finanza. E' tuttavia difficile determinare lo "stato dell'arte" circa l'utilizzo delle reti neurali in ambito economico-finanziario perché generalmente le ricerche in questo campo sono finanziate da consorzi, imprese o istituzioni finanziarie che hanno l'obiettivo di trovare un modello, generalmente previsionale, che assicuri un vantaggio competitivo sui concorrenti; dato che un vantaggio competitivo rimane tale fino al momento in cui non viene imitato dai concorrenti, la ricerca conduce alla creazione di sistemi proprietari che comportano l'inaccessibilità delle ricerche, dei dati empirici e delle soluzioni trovate relativamente ad una buona parte del lavoro svolto fino ad oggi. Inoltre, molte ricerche sono state sviluppate a partire da dati empirici specifici, il che rende abbastanza difficile il confronto tra diverse soluzioni.

E' possibile classificare i lavori, sia di ricerca che applicativi, aventi ad oggetto l'applicazione delle reti neurali al campo economico-finanziario in:

- **Classificazione e discriminazione**. In questo caso la rete neurale dev'essere in grado di assegnare ogni pattern di input ad una tra un insieme di categorie. Nei problemi di classificazione questo insieme è definito dall'utente; nei problemi di discriminazione invece è la rete stessa che dev'essere in grado di creare le categorie da utilizzare (in modo simile alla cluster-analysis, si veda [11], cap. 8). Questi problemi riguardano tipicamente le applicazioni inerenti il rischio di credito, nelle fasi di determinazione della classe di rating o probabilità di default e successiva decisione di affidamento;

- **Previsione di serie temporali**, con riferimento classico ai prezzi delle azioni e di altri titoli negoziabili. E' in questo campo che si possono trovare la maggioranza dei contributi, mossi dallo scopo finale di realizzare profitti tramite attività di trading sui titoli negoziati. E' da rilevare che la maggioranza dei lavori in questo campo è stata prodotta in America Settentrionale;

- **Approssimazione di funzioni**. E' un ambito di indagine residuale, in quanto riguarda la valutazione degli strumenti in qualsiasi funzione di pricing e di risk management per i quali non esistono modelli accreditati di determinazione. A ben vedere, è possibile far rientrare in questa categoria anche le due classi di problemi descritte in precedenza. Rientrano in questo campo l'individuazione di frodi tramite carte di credito, la previsione di prezzi di opzioni di tipo americano o esotico ed una grande varietà di altre applicazioni.

L'uso delle reti neurali nei primi due principali ambiti di ricerca definiti in precedenza si è rivelato più accurato dei tradizionali modelli lineari in diverse applicazioni, soprattutto nel catturare relazioni complesse estranee all'analisi lineare. E' questo ad esempio il caso della previsione della volatilità: Hamid e Iqbal [10] hanno dimostrato che una rete feed-forward addestrata con Back-Propagation è in grado di prevedere con buona accuratezza la volatilità relativa allo *S&P Index future options*. In questo studio i risultati delle previsioni non si discostano di molto dalla volatilità effettiva, mentre il modello classico basato sul *Barone-Adesi and Whaley (BAW) American futures options pricing model* ha prodotto risultati molto discordi da quelli effettivi in relazione a due orizzonti temporali di previsione su tre.
A risultati che vanno nella stessa direzione ha portato il lavoro di Desai e Bharati [5], che hanno dimostrato come nelle previsioni finanziarie le previsioni effettuate con reti neurali sono condizionalmente efficienti rispetto ai metodi di regressione lineare, ma non è possibile affermare il contrario. Tuttavia, in uno studio sulla produzione industriale europea, è stato osservato che i modelli lineari offrono performance migliori delle reti neurali in previsioni con orizzonte temporale minore all'anno solare; le reti neurali sembrano comunque in grado di fornire previsioni più accurate in relazione alla direzione del cambiamento osservato [14].

In relazione ai diversi lavori incontrati è necessario fare alcune precisazioni. Innanzitutto si nota un certo livello di disorganicità, in quanto è facile trovare contributi di scarso interesse accanto a lavori che introducono concetti importanti e che hanno avuto rilevanza nella ricerca successiva. Inoltre molto spesso non sono riportati i dati su cui è condotto l'esperimento e non è spiegato il criterio di selezione delle variabili in ingresso, provocando grandi ostacoli alla riproducibilità dell'esperimento. Spesso non sono spiegati i motivi che hanno indotto a preferire una architettura di rete piuttosto che un'altra; a tale proposito si nota tuttavia una forte predilezione per architetture feed-forward addestrate con Back-Propagation: se da un lato questo è comprensibile, vista la facilità di implementazione e la bontà dei risultati, da

un altro lato si potrebbe verificare una convergenza dell'interesse di ricerca solo su questo tipo di architettura. A onor del vero, va detto che i risultati offerti dalla Back-Propagation, con qualche miglioramento quali il Cascade, il Pruning, il Weight Decay o il semplice momentum, si sono rivelati interessanti ed efficaci, specie se confrontati con risultati prodotti da singolari combinazioni di architetture di rete o di sistemi ibridi creati *ad hoc* per un determinato esperimento che presentano difficoltà di implementazione e interpretazione e risultati a volte inconsistenti. Comunque va detto che anche le reti ricorrenti (sempre addestrate tramite Back-Propagation) trovano uno spazio importante, in relazione ai risultati che offrono, in questo panorama composito.

Il fatto che non esistano regole precise per strutturare le reti ha portato alla generazione di diversi modelli basati su diverse regole e parametri, in cui grande ruolo investono l'abilità, l'esperienza e l'intuito del ricercatore. L'approccio principale resta il try-and-error, e l'unica regola (più che regola deve essere intesa come linea-guida e suggerimento) che sembra essere accettata con sicurezza è quella di Rumelhart: in riferimento ad un insieme di dati, la rete più semplice e robusta è quella che, in media porta alla migliore generalizzazione rispetto alla popolazione da cui i dati sono stati estratti. In altre parole è un suggerimento ad utilizzare architetture di rete semplici [12].

Tutti i lavori cercano di evitare l'over-fitting (si riscontra che le regole di stop-learning sono sensibilmente differenti da un lavoro ad un altro), ma ciò nonostante i risultati sono comunque affetti da un certo empirismo, in quanto la configurazione della rete dipende dai dati disponibili, dalla suddivisione in training e testing set, etc. E' questo un aspetto della ricerca da tenere ben presente specie quando si vuole applicare una determinata architettura ad un problema per il quale non è stata progettata: la rete neurale universale non esiste, o perlomeno non è stata ancora trovata. Non esiste poi uno standard per la misurazione della performance: risulta difficile confrontare lavori che si basano su orizzonti temporali diversi e criteri di misurazione diversi. Risulta poi impossibile includere nei termini di confronto i costi ed i tempi di sviluppo.

Ultima considerazione: in letteratura si riscontra una certa disomogeneità nell'utilizzo dei termini tecnici. Abbiamo già fatto notare l'alternanza di significato assegnata a output e attivazione, ma si può estendere la considerazione all'uso di altri termini; ad esempio per Multi Layer Perceptron si intende a volte un'architettura Feed-Forward, oppure Back Propagation può indicare erroneamente sia l'algoritmo di apprendimento, sia l'architettura feed-forward; si nota inoltre una certa confusione nelle definizioni di memoria associativa e auto-associativa (che viene a volte ricompresa nel primo tipo). E' facile notare queste difformità in quanto diversi lavori spendono molte parole sul funzionamento generale del modello, tralasciando, come detto in precedenza, aspetti considerabili più importanti dal punto di vista del

metodo scientifico[5].

Abbiamo visto che tra le proprietà più importanti delle reti neurali c'è la generalizzazione, ossia la capacità di lavorare sulla parte di una popolazione di dati non incontrata durante la fase di apprendimento. La previsione è stata per diverso tempo ad appannaggio dei sistemi lineari (che assumono cioè che le serie storiche sotto studio sono generate da processi lineari), che hanno il vantaggio di essere facili da implementare e da comprendere, ma si rivelano totalmente inefficaci se le osservazioni oggetto di studio sono generate da processi non lineari, e di fatto i sistemi osservabili nel mondo reale sono non lineari. I modelli non lineari si rivelano ancora meno efficaci, in quanto è necessario ipotizzare una relazione esplicita (tra tante disponibili) per i dati, che spesso si rivela non in grado di comprendere caratteristiche importanti. In questo quadro le reti neurali, che come abbiamo descritto rappresentano un approccio data-driven, sono in grado di cogliere relazioni non lineari tra dati senza aver bisogno della specificazione del modello sottostante e rappresentano forse l'alternativa più efficace per ricercatori e professionisti (illazione ricavata dall'incredibile aumento della ricerca in tale campo negli ultimi due decenni) anche se non si è ancora in grado di determinare i fattori del successo delle reti neurali. A titolo di esempio può essere significativo il fatto che in una "competizione" tra modelli previsionali organizzata dal Santa Fe Institute (1993), i vincitori su ogni serie di dati (di discipline quali fisica, finanza, astrofisica ed anche musica) utilizzarono modelli di reti neurali.

La prima applicazione delle reti neurali per la previsione è del 1964, quando si applicarono le teorizzazioni di Widrow-Hoff alle previsioni del tempo. Come abbiamo già accennato, l'assenza di algoritmi di apprendimento per reti multistrato ostacolò fortemente la ricerca in tale direzione, e le prime applicazioni interessanti risalgono alla fine degli anni ottanta, quando vennero ideate reti neurali per modellare sistemi non lineari e serie caotiche affetti da rumore. Si tratta però di lavori ad argomento fisico ed ingegneristico (consumo di elettricità, temperatura dell'ambiente, precipitazioni atmosferiche, fabbisogno di acqua, trasporti etc). Per lavori di carattere finanziario bisogna aspettare gli anni novanta.

La previsione temporale può avere ad oggetto soltanto un periodo successivo (one-step-ahead) oppure più di un periodo successivo (*multi-step-ahead*); in quest'ultimo caso esistono due metodi per effettuare la previsione: il metodo iterativo e quello diretto. Nel metodo iterativo il valore di output è utilizzato come input per la previsione successiva e di conseguenza è necessario soltanto un neurone di output; nel metodo diretto invece la rete determinerà i valori previsti soltanto in relazione a quelli osservati, e quindi saranno necessari diversi nodi di output. Anche se le opinioni ad oggetto sulle due forme sono discordi, si ritiene che il metodo diretto

[5]Ad esempio non vengono riportati i dati oggetto dell'analisi e non è specificato, quando si utilizza la normalizzazione, se la performance è misurata in base ai dati effettivi o quelli normalizzati, compromettendo la riproducibilità dell'esperimento.

sia migliore in quanto quello iterativo utilizza i valori previsti per generarne altri, perdendo di vista i valori effettivi man mano che la previsione va avanti, provocando quindi una diminuzione dell'accuratezza della previsione all'aumentare dell'orizzonte temporale. Sono stati anche proposti degli strumenti per combinare i due metodi descritti.

2.1 Il nostro caso: *Il Rischio di Credito*

l rischio di credito[6] è rappresentato, nell'attività delle banche e delle altre istituzioni creditizie, dalla possibilità che un debitore non adempia alle sue obbligazioni; è importante per le banche prevedere il rischio di insolvenza di un potenziale cliente in quanto una buona stima può portare ad una migliore allocazione delle risorse. L'incremento della concorrenza nel mercato del credito e la possibilità che le aziende ricorrano a propri strumenti creditizi per finanziarsi, unite al fallimento di importanti banche asiatiche hanno portato ad un aumento dell'attenzione sul problema del rischio di credito, sia tra i ricercatori che tra i professionisti. Le banche sono ormai portate ad ampliare i crediti concessi a piccole e medie imprese per non diminuire i propri volumi di attività, con crescente necessità di una corretta valutazione dei rischi di credito collegati. I circa 5000 miliardi di dollari di debiti pendenti negli Stati Uniti (2001) ci possono ben far comprendere la rilevanza della previsione di insolvenza: un miglioramento delle stime sul rischio di credito può portare a risparmiare milioni di dollari, oltre che ad aiutare a "personalizzare" il tasso d'interesse in base alla probabilità di insolvenza della controparte in quanto maggiore è il rischio di insolvenza maggiore dev'essere il tasso di interesse applicato, perché questo dovrà contenere un premio per la sopportazione del rischio. In caso di insolvenza la banca va incontro ad una perdita che dipende dalla probabilità che l'evento si verifichi e dalla parte di credito non rimborsato. Infatti lo schema logico su cui si fonda il problema del rischio di credito è il seguente:

$$p \cdot (1 + k) = 1 + i$$

dove

p = probabilità che capitale ed interessi vengano totalmente rimborsati;
k = tasso di interesse comprensivo del rischio;
i = tasso depurato dal rischio (nominale).

Dati i e p è semplice ricavare k

$$k = \frac{1 + i}{p} - 1$$

[6]Per una trattazione completa dell'argomento si vedano [28], [3].

Se si prende in considerazione il fatto che generalmente una parte del credito viene recuperata anche in caso di default, lo schema logico viene modificato nel seguente modo:

$$((1-p)\cdot rr\cdot(i+k))+p\cdot(1+k)=1+i$$

dove
rr = recovery rate, ossia la parte del credito recuperata in caso di default; $1-p$ = probabilità dell'evento di default.
Da quanto detto segue che

$$k=\frac{1+i}{(rr-(rr\cdot p)+p)}-1$$

Da queste equazioni risulta evidente quanto importante è stimare correttamente la probabilità di default e il recovery rate. In relazione alla prima variabile sono state definite diverse tecniche che verranno brevemente descritte successivamente. Il recovery rate presenta problemi di determinazione maggiori, in quanto la sua valutazione, oltre a dipendere da fattori il cui valore è facilmente determinabile (ipoteche, fideiussioni etc.), avviene in relazione a fattori immateriali di difficile valutazione economica (brevetti, marchi, catena distributiva, correttezza e continuità dei rapporti etc.); di difficile valutazione poi è il tempo di recupero del credito, anche se esistono agenzie di rating specializzate nel determinare il tempo medio di recupero per debitori insolventi appartenenti a classi di rating con diverse caratteristiche. Le eventuali conseguenze del default (liquidazione o procedure concorsuali) aggiungono ulteriore incertezza alla determinazione di questo parametro in quanto portano ad una diversa valutazione delle attività del cliente.

Il nostro lavoro sarà incentrato sulla classificazione dello stato di bonis o default delle aziende; questi stati sono convenzionalmente associati ai valori 1 (default) e 0 (bonis). Tuttavia lo strumento a nostra disposizione (la rete neurale) restituisce per ogni azienda un valore compreso tra 0 ed 1. Questo comportamento può esserci di aiuto per determinare il valore di una eventuale probabilità di default (o bonis). In tal senso possiamo dire che opereremo solo sulla determinazione della probabilità di default, tralasciando quella del recovery rate. La determinazione della probabilità di default è utile per banche e altre istituzioni finanziarie nelle seguenti decisioni operative:

- decisione di assegnamento del credito ad un cliente richiedente (fase di screening); monitoring dei crediti concessi, al fine di decidere se mantenere il prestito e/o modificarne il prezzo;
- analisi dei diversi rischi e delle eventuali correlazioni tra essi al fine di determinare le decisioni strategiche;
- determinazione del valore del credito per determinare eventuali riserve e per la redazione del bilancio.

2.2 Il modello più comune utilizzato dalle banche

Per la determinazione del rischio di credito si può procedere con un'analisi soggettiva o oggettiva. Il primo tipo di analisi si incentra su giudizi formulati da esperti o da organi collegiali circa l'impresa in considerazione, valutando la reputazione, i rapporti instaurati in precedenza ed altri parametri a forte componente soggettiva. Il secondo tipo, che ha avuto sviluppo a partire dalla fine degli anni '60 in particolar modo nei paesi anglosassoni, è da ricondurre a modelli che analizzano indicatori economici e finanziari ricavati dai bilanci delle aziende per stimare la probabilità di default ed operare una classificazione delle aziende in base a questa allo scopo di determinare quali saranno le imprese in buona salute e quelle insolventi; lo scopo di questi modelli è quello di trovare delle relazioni tra indicatori di tipo economico-finanziario ed evento di default. Il più utilizzato, anche in relazione all'applicazione delle banche, è il modello Z-SCORE proposto da Altman [2], che consiste in un modello lineare volto a separare le imprese in buona salute da quelle insolventi in base ad un numero di *cut-off* (Z) determinato in base ad una regressione lineare del tipo

$$Z = a_1 x_1 + a_2 x_2 + \ldots + a_n x_n$$

dove le variabili x rappresentano gli indicatori utilizzati. Se per l'osservazione considerata il valore di Z è minore di un *cut-off* definito dall'utente l'impresa potrà essere ritenuta potenzialmente a rischio, altrimenti in buona salute. I coefficienti vengono stimati in modo da massimizzare la differenza tra la media degli score Z tra le due classi e di minimizzarne la varianza all'interno della stessa classe. Questi coefficienti indicano la capacità discriminante della variabile considerata, ma bisogna tenere ben presente che il loro valore è influenzato dall'unità di misurazione. Il segno dei coefficienti è molto indicativo perché fornisce la direzione dell'influenza esercitata dalla variabile sullo score, ma anche in questo caso bisogna prestare attenzione perché il segno determinato dalla regressione potrebbe risultare diverso da quello che intuitivamente saremmo portati ad immaginare; questo è generalmente dovuto al fatto che in una regressione alcuni indicatori servono a bilanciare il peso di altri.

E' opportuno operare una analisi di correlazione delle variabili scelte in modo da non prendere in considerazione variabili e indicatori troppo intensamente correlati, poi è opportuno definire l'importanza relativa di ogni indicatore scelto (coefficienti di correlazione tra variabile indipendente e dipendente) per valutare quali usare o meno. Il quest'ultimo caso il segno del coefficiente di correlazione è lo stesso del coefficiente relativo nella funzione di regressione. Occorre poi stilare una tabella di assegnazione dei diversi esempi alle classi per valutare la bontà della regressione, definita in base agli errori di classificazione commessi. Questi possono essere di due tipi: il primo consiste nel classificare una impresa sana nel gruppo delle insolventi; il secondo sta, viceversa, nel classificare una impresa insolvente come sana. Per le banche il secondo tipo di errore è quello più critico, per cui sarebbe buona norma

privilegiare gli indicatori che considerati singolarmente portano a basse percentuali di errori di classificazione del secondo tipo ed i modelli che in genere offrono migliore capacità previsionale in tal senso, anche se alcuni autori sembrano preferire modelli con capacità previsionale simile per i due gruppi.

Ovviamente prima di tutto ciò deve essere determinato il campione su cui si va ad operare l'indagine in modo da confrontare imprese sane da imprese inadempienti ed è buona norma rispecchiare la differente presenza dei due tipi di imprese nell'ambiente economico considerato. E' utile inoltre non considerare nel campione delle aziende insolventi quelle per cui il default è derivato da cause non prevedibili e non imputabili, quali catastrofi naturali o morte del titolare. Il caso di aziende che presentano dati incompleti, che alcuni autori tendono ad escludere, può essere invece risolto attraverso opportune azioni di pre-processing dei dati. La bontà di questo metodo è subordinata al soddisfacimento di alcune condizioni:

- per ogni variabile esplicativa le osservazioni devono essere indipendenti;
- ogni variabile esplicativa deve avere una distribuzione normale;
- per ogni variabile esplicativa la varianza dev'essere simile per le due classi;
- per ogni coppia di variabili esplicative la covarianza dev'essere simile per le due classi.

Altri strumenti non fanno fondamento su ipotesi così restrittive. Ad esempio il modello *logit*, che consiste anch'esso in una regressione, ha come unica condizione che per ogni variabile esplicativa le osservazioni siano indipendenti. Va detto che il soddisfacimento di tali vincoli presenta comunque aspetti soggettivi e che anche quando qualcuno di questi non viene soddisfatto il modello Z-score non presenta performance estremamente peggiori.

Molti studi utilizzano i seguenti indicatori (i primi ad essere utilizzati da Altman) oppure altri derivati da questi:[7]

$$\frac{\text{capitale circolante}}{\text{totale impieghi}}$$

$$\frac{\text{riserve}}{\text{totale impieghi}}$$

[7]Per i seguenti indicatori i coefficienti stimati da Altman sono rispettivamente 1.2, 1.4, 3.3, 0.999, 0.6 e i possibili valori dei cut-off sono suddivisi in classi di insolvenza: per valori superiori a 3 l'azienda è ritenuta "sicura", per valori tra 2.7 e 2.99 l'azienda è denotata come "On Alert". Valori tra 1.8 e 2.7 indicano buone probabilità di fallimento entro 2 anni; valori inferiori a 1.8 indicano probabilità di "financial embarassment" molto alta. I valori dei cut-off variano a seconda delle pubblicazioni e dei modelli proposti.

$$\frac{\text{utile ante tasse ed oneri finanziari}}{\text{totale impieghi}}$$

$$\frac{\text{fatturato}}{\text{totale impieghi}}$$

$$\frac{\text{valore di mercato del capitale azionario}}{\text{debiti totali}}$$

Passando rapidamente in rassegna il significato di ogni indicatore possiamo dire che il primo rapporto indica la capacità dell'impresa di estinguere i suoi debiti a breve termine (il totale degli impieghi è un buon indicatore della dimensione aziendale, per cui viene spesso utilizzato come normalizzante).
Le riserve indicano gli utili ritenuti dall'azienda ed offrono una garanzia su eventuali crediti.
Il significato degli utili è abbastanza intuitivo: alti utili dovrebbero stare a significare uno stato di buona salute dell'azienda, mentre utili negativi (perdite) indicano che l'impresa perde competitività e quote di mercato, diminuendo garanzie di adempimento delle sue obbligazioni in scadenza. E' però suscettibile ad alterazioni derivate da politiche di bilancio volte ad "occultarne" alti valori a scopi fiscali, quindi va utilizzato con cura; tale problema è minore quando si indica il fatturato come indicatore dell'attività.
L'ultimo indicatore è probabilmente il meno intuitivo. Dato che l'impresa può estinguere i suoi debiti emettendo e collocando sul mercato nuove azioni, una elevata capitalizzazione indica potenzialmente una buona capacità dell'impresa di soddisfare gli interessi dei creditori.

Dato che il mercato delle azioni è ritenuto un indicatore della buona salute (e quindi potenziale solvibilità) delle imprese e viene visto anche come strumento di previsione di problemi o miglioramenti delle performance aziendali, alcuni studi (per le reti neurali si veda Atiya [4]) utilizzano indicatori derivati dal mercato azionario per ottenere benefici specie nelle previsioni a lungo periodo (lo studio considera un orizzonte di tre anni) utilizzano indicatori derivati dal mercato azionario. Per quanto possa portare a significativi incrementi delle prestazioni, il loro utilizzo è limitato soltanto alle società quotate in borsa, e quindi non è riconducibile a tutti i casi.

Il modello Z-SCORE è utilizzato con successo come segnale di allarme di potenziali crisi d'impresa (*early warning*, quindi nella fase di monitoring), ma non si rivela

particolarmente efficace per una corretta misurazione del rischio di credito e per la determinazione del prezzo. Inoltre si ritiene che la congiuntura possa influenzare la capacità di spiegazione del default degli indicatori, per cui si dovrebbero adottare delle metodologie per combinare questo modello con altri oppure usare il modello proposto come tool ausiliario alle scelte dell'esperto, anche se questo aumenterebbe la componente soggettiva e la relativa probabilità di falsificare i risultati dell'analisi. Inoltre variazioni della stessa misura proporzionale di alcuni identificatori hanno degli effetti diversi sulla previsione del rischio di credito quando dimostrano diversi stati di salute dell'impresa[8], per cui un approccio lineare puro può essere non desiderabile in molti casi ed è, in generale, un modello estremamente semplificato della realtà.

2.3 Le reti neurali ed il rischio di credito

L'approccio lineare rientra tra i modelli chiamati "strutturali": essi si propongono di individuare e definire le relazioni tra variabili che portano alla descrizione di un fenomeno (nel nostro caso la solvibilità aziendale). Tuttavia, proprio nel nostro caso specifico, la conoscenza del modello può portare a comportamenti scorretti da parte delle imprese: se esse conoscessero il modello di Z-SCORE utilizzato dalle banche (quindi indicatori utilizzati e coefficienti stimati) potrebbero operare una serie di politiche di bilancio al fine di far sembrare la loro situazione aziendale migliore di quanto sia in realtà, con i benefici che da questo possono derivare al fine di una richiesta di prestito. Per questo esse potrebbero cercare di ricostruire induttivamente il modello, facendogli perdere efficacia nel tempo. L'approccio alternativo, e vale a dire l'approccio *"black-box"*, non è sottoposto a questo tipo di inconvenienti, in quanto riesce a costruire un sistema di approssimazione, ma non permette di specificare le relazioni tra variabili. Le reti neurali sono il simbolo di questo secondo approccio, e il loro utilizzo per la previsione del rischio di credito è iniziato nei primi anni '90; anche in questo campo l'architettura più utilizzata è stata la feed-forward addestrata con back-propagation[9]. Particolare attenzione è stata rivolta alla determinazione delle variabili di input da fornire alle reti (a riguardo si veda Atiya [4]).

Li, Pang e Xu [32] utilizzano una rete feed-forward a 3 strati addestrata tramite Back-Propagation Store con 7 neuroni di input, uno strato nascosto formato da 8 neuroni e 2 neuroni di output per suddividere 120 soggetti richiedenti credito in 3 classi (buoni-solventi, medi, cattivi-insolventi). Le variabili in ingresso sono 7 indici che riflettono la capacità di estinguere il debito, la profittabilità, la struttura

[8] Ad esempio, pensiamo ad un indice che in due periodi di rilevazione assuma valori $\frac{5}{1000}$ e $\frac{10}{1000}$. La variazione proporzionale sarebbe la stessa riscontrata nel caso in cui questo assuma, in due periodi di rilevazione, valori 50000 e 100000, ma si può intuitivamente pensare che le condizioni sottostanti siamo diverse nei due casi.

[9] Per approfondimenti sull'utilizzo delle reti neurali nella previsione del rischio di credito si veda [8], [9].

del capitale e la qualità del management. Il numero di neuroni nascosti è definito tramite Cascade: si parte dall'utilizzo di un solo neurone per poi aggiungerne uno alla volta, fino a che non si riscontrano più miglioramenti nella performance della rete. Essi dimostrano che in relazione alla valutazione del rischio di credito le reti neurali offrono una accuratezza migliore rispetto al tradizionale approccio statistico dell'analisi lineare discriminante. A risultati simili è arrivato uno studio di Pang, Wang e Bai [30], che hanno utilizzato una rete feed-forward simile a quella descritta in precedenza per classificare 106 aziende cinesi in due categorie (solventi-insolventi), dimostrando un alto tasso di accuratezza della previsione, suggerendo l'uso di questo modello in futuro per prevedere stati di "cattiva salute" delle aziende. Alcuni lavori utilizzano una combinazione di reti neurali di diversa tipologia, addestrando la prima rete a prevedere una determinata serie e la seconda a prevedere l'errore della prima, ottenendo migliori risultati rispetto all'utilizzo di una sola rete o di un semplice modello lineare. C'è chi ha proposto anche l'utilizzo delle reti come strumento ausiliario rispetto ad un altro modello di previsione.

Generalmente dal confronto con i modelli lineari emerge una superiorità delle reti neurali specie perché, come esposto in precedenza, si ritiene che il modificarsi della congiuntura e l'evolversi della situazione storica portino ad un cambiamento dell'importanza esplicativa delle diverse variabili che i modelli lineari non sono in grado di determinare, mentre le reti neurali, grazie alla loro natura data-driven, riescono a fornire buone prestazioni. Tuttavia, questa superiorità è dovuta anche al modo in cui si definisce l'utilizzo dei dati per le procedure di addestramento. Ad esempio, quando invece di utilizzare la classica distinzione tra training e test set si utilizza la procedura LOO (*leave-one-out*) le reti neurali presentano risultati migliori su tutti gli orizzonti di previsione [39]. Non mancano però, anche in questo caso, lavori in cui i confronti tra modelli portano a risultati non interpretabili in termini di superiorità di uno su altri.

Un'interessante direzione di sviluppo della ricerca potrebbe essere il considerare gli indicatori macro-economici e fornirli in ingresso alla rete neurale: la congiuntura ha un impatto sulla probabilità di insolvenza delle imprese, ma questo aspetto non è stato molto preso in considerazione fino ad oggi. Per una interessante rassegna dei risultati ottenuti dall'applicazione della reti neurali al rischio di credito rimandiamo nuovamente al fondamentale lavoro di Atiya [4].

2.4 Una potenziale applicazione: L'accordo di Basilea

Lo scopo del presente lavoro è di creare una rete neurale che sia in grado di prevedere se una azienda restituirà o meno il capitale preso a prestito da una banca. Tipicamente la previsione è legata al risultato fornito in output dalla rete: il valore 0 starà a significare che l'azienda restituirà il prestito, mentre il valore 1 indicherà che l'azienda non restituirà il prestito. I risultati forniti dalla rete però saranno compresi

nell'intervallo reale $[0, 1]$, per cui bisognerà operare una operazione di interpretazione e approssimazione dei risultati. Tuttavia l'output fornito dalla rete potrebbe essere interpretato come la probabilità che si verifichi l'inadempimento da parte dell'impresa richiedente il prestito, e questa interpretazione può essere utile nel caso in cui le banche (gli istituti che cioè concedono i prestiti alle imprese) siano tenute a quantificare i requisiti patrimoniali per far fronte a possibili inadempimenti in relazione alla probabilità che i crediti concessi risultino insoluti.

Nel 1974 i governatori delle Banche Centrali dei 10 paesi più industrializzati al mondo (G10) istituirono il Comitato di Basilea, al fine di ripartire le responsabilità di vigilanza tra le autorità nazionali e garantire un'efficace supervisione delle attività bancarie a livello internazionale. Le decisioni più importanti del comitato sono volte a determinare i requisiti patrimoniali di vigilanza, ossia le quote di capitale che le banche devono accantonare in relazione al rischio assunto in conseguenza alla concessione di un credito. Il comitato opera presso la BRI (Banca dei Regolamenti Internazionali con sede a Basilea, Svizzera) e attualmente è composto da membri provenienti da Belgio, Canada, Francia, Germania, Giappone, Italia, Lussemburgo, Paesi Bassi, Regno Unito, Spagna, Stati Uniti, Svezia, Svizzera e le conclusioni a cui perviene non hanno efficacia legale ma sono redatte nell'ottica che le banche di tutti i paesi (e soprattutto le autorità nazionali dei paesi aderenti e non) possano disporre regole operative che, pur tenendo conto delle diversità dei singoli stati, si ispirino a esse.

Il primo accordo di Basilea è del 1988 ed in esso (a cui hanno aderito oltre 100 stati) si definisce la quota di capitale da accantonare nella misura dell'8% del capitale erogato. Il chiaro limite di quest'accordo è che non tiene conto della qualità del prenditore e delle eventuali realtà aziendali: una tale percentuale può essere considerata eccessiva per un debitore affidabile ed insufficiente per eventuali debitori rischiosi. La misura quindi non è sensibile al problema del rischio di credito e l'emergere di questo problema portò alla definizione di un nuovo accordo nel 2001 (la cui versione definitiva è stata pubblicata nel giugno 2004). In realtà, sebbene l'attuazione dell'accordo sia prevista per la fine del 2006, le banche dovranno dimostrare di aver operato per almeno 3 anni in conformità alle disposizioni per poter usufruire dei particolari strumenti operativi previsti dall'accordo (che si rivelano meno onerosi per le banche).

Nel nuovo accordo si definiscono tre campi d'azione (pilastri): i requisiti patrimoniali minimi per le banche, il potere di controllo delle Banche Centrali e le regole di trasparenza per l'informazione al pubblico. Il primo pilastro è quello che più interessa ai fini del nostro lavoro. Per la determinazione della probabilità di default (inadempimento) le banche possono utilizzare la metodologia che più ritengono opportuna, quindi fare riferimento a sistemi lineari, neurali, giudizio di esperti, utilizzando ad esempio una delle metodologie viste in precedenza. Si noti però che il secondo pilastro affida all'autorità delle Banche Centrali la verifica dell'affidabilità degli strumenti di valutazione delle banche, anche con la possibilità di imporre requisiti patrimoniali

maggiori a seconda del profilo di rischio della banca. Le disposizioni del terzo pilastro infine intendono garantire la trasparenza in merito al profilo di rischio degli istituti di credito e in merito agli strumenti utilizzati per determinare i coefficienti di rischio.

Il fatto che le banche debbano classificare i propri clienti in base alla propria rischiosità ha portato diversi studiosi ed opinionisti alla preoccupazione che, considerando la relazione tra maggior rischio e maggiori accantonamenti-costi per le banche, l'applicazione dell'accordo possa portare a tassi d'interesse più elevati per le imprese medio-piccole. Per questo le imprese in questione dovrebbero attuare una serie di politiche di bilancio e gestionali per diminuire la loro rischiosità percepita e valutata dalle banche. Il problema delle piccole imprese è stato evidenziato anche dal fatto che le autorità italiane e tedesche (Banca d'Italia e Bundesbank) sono intervenute sul problema chiedendo delle aggiunte e modifiche all'accordo appunto per una maggiore tutela delle piccole e medio-imprese sulle quali si basa la fetta più grande del sistema economico dei due paesi. Inoltre le banche più piccole non potranno utilizzare le metodologie più avanzate e questo le porterà a sopportare requisiti patrimoniali maggiori rispetto alle banche più grandi. A ciò va aggiunto che in periodi di crisi economica il provvedimento avrebbe come effetto quello di ridurre gli impieghi, con la conseguenza di aumentare la crisi stessa. Proprio in base a queste ultime considerazioni il primo pilastro dell'accordo offre alle banche 3 metodi per determinare l'accantonamento a fronte del rischio di credito: lo Standard Approach, il metodo IRB (Internal Rating Based approach) e il metodo IRB avanzato.

Il primo metodo riprende le linee generali del primo accordo di Basilea, in pratica viene confermato l'accantonamento dell'8% a fronte di ogni posizione. Tuttavia l'ammontare così determinato dev'essere ponderato in base a valutazioni esterne della qualità del credito (quindi l'appartenenza a classi di rating), maggiorando in tal senso gli accantonamenti per crediti verso aziende con rating scarsi e riducendo quelli verso aziende con rating buoni secondo i seguenti coefficienti per classi di credito. L'approccio IRB prevede che le banche assegnino un rating interno ed una probabilità di default (pd) ai singoli clienti, mentre i fattori di rischio addizionali (esposizione al momento dell'inadempienza ead, perdita in caso di inadempienza lgd, scadenza effettiva m) sono stimati in ossequio a parametri prudenziali definiti dalla Banca Centrale. Con il metodo IRB avanzato anche questi ultimi parametri possono essere stimati internamente dalle banche. Come abbiamo accennato, l'utilizzo degli ultimi due metodi è consentito solo alle banche che dimostrino di aver operato secondo le metodologie da essi suggeriti da almeno 3 anni. Tuttavia sono importanti al fine delle scelte riguardanti il modello da utilizzare le direttive della Banca Centrale, che nel nostro caso (Banca d'Italia) sembra aver suggerito, almeno per le banche di grandi dimensioni, l'utilizzo degli approcci basati su modelli interni. L'accordo definisce comunque i requisiti minimi che le banche devono utilizzare per impiegare il sistema IRB, e anche se le banche possono avvalersi di strumenti interni per stimare le componenti di rischio, in alcuni casi esse potrebbero essere invitate a impiegare valori

prudenziali in luogo delle loro stime interne. Il metodo si basa su misure delle perdite inattese e delle perdite attese, ma dalle formule proposte si ricavano i requisiti patrimoniali a fronte delle perdite inattese (quelle attese vengono trattate separatamente). Ai fini della distinzione tra metodo di base ed avanzato le banche devono ripartire i crediti concessi nelle seguenti classi di esposizioni (in linea comunque con le classificazioni già adottate dalle banche) :

- esposizioni verso imprese;
- esposizioni verso soggetti sovrani;
- esposizioni verso banche;
- esposizioni azionarie;
- esposizioni al dettaglio.

All'interno della prima classe sono definiti 5 sottogruppi:

- finanziamento di progetti;
- finanziamento di attività materiali a destinazione specifica;
- finanziamento su merci;
- finanziamento di immobili da investimento;
- finanziamento di immobili commerciali ad alta volatilità.

Bisogna notare che per quanto riguarda le esposizioni inferiori a 1000000 Euro le esposizioni verso piccole imprese possono essere equiparate a crediti al dettaglio se vengono amministrate, all'interno della banca, secondo gli stessi criteri applicati alle altre esposizioni al dettaglio e se è gestita come parte di un pool di attività con caratteristiche simili al fine della valutazione del rischio. La normativa (più che normativa sono delle direttive indirizzate alle Banche Centrali) prevede, fatte le dovute eccezioni, il seguente accantonamento per crediti verso imprese, soggetti sovrani e banche[10]:

$$k = \{lgd \cdot N[(1-r)^{-\frac{1}{2}} G(pd) + (\frac{r}{1-r})^{\frac{1}{2}} \cdot G(0,999)] - pd \cdot lgd\} \cdot$$

$$\cdot (1 - 1,5 \cdot b)^{-1} \cdot [1 + (m - 2,5) \cdot b]$$

dove r indica la correlazione e si calcola nel seguente modo:

$$r = 0,12 \cdot \frac{(1 - e^{-50pd})}{1 - e^{-50}} + 0,24 \cdot \frac{1 - (1 - e^{-50pd})}{1 - e^{-50}}$$

[10] k = Requisito Patrimoniale; N indica la funzione normale e G la sua inversa.

e b indica l'aggiustamento in funzione della scadenza e si calcola nel seguente modo:

$$b = [0,11852 - 0,05478 \cdot \ln(pd)]^2$$

In questo caso sono previsti aggiustamenti per le piccole e medie imprese (società facenti parte di un gruppo consolidato il cui fatturato è inferiore a 50000000 Euro): è previsto un aggiustamento specifico nel calcolo della correlazione (S rappresenta il fatturato annuo in milioni di Euro, anche se le autorità nazionali di vigilanza possono usare il totale delle attività consolidate del gruppo in cui il fatturato non sia una misura significativa della dimensione aziendale).

$$r = 0,12 \cdot \frac{(1 - e^{-50pd})}{1 - e^{-50}} + 0,24 \cdot \frac{1 - (1 - e^{-50pd})}{1 - e^{-50}} - 0,04 \cdot \left(1 - \frac{S - 5}{45}\right)$$

Le banche che non possiedono i requisiti per l'utilizzo del metodo IRB devono applicare, verso imprese, l'accantonamento dell'8 % ponderato dai seguenti coefficienti in relazione all'appartenenza alle seguenti classi di rating:

Forte	0,7
Buono	0,9
Sufficiente	1,15
Debole	2,5
Inadempiente	non applicabile

per le seguenti tipologie:

- finanziamento di progetti;
- finanziamento di attività materiali a destinazione specifica;
- finanziamento su merci;
- finanziamento di immobili da investimento.

Per quanto riguarda il finanziamento di immobili commerciali ad alta volatilità invece i coefficienti in relazione alle classi di rating sono i seguenti:

Forte	0,95
Buono	1,2
Sufficiente	1,4
Debole	2,5
Inadempiente	non applicabile

Per le esposizioni verso banche e imprese la pd può essere determinata in base alla metodologia interna dell'azienda (per la categoria "inadempiente" $pd = 1$) ma non può essere minore di $0,0003$.

Anche per la perdita in caso di inadempienza (lgd) sono possibili 2 metodi: il metodo base e quello avanzato. Nell'ambito del metodo base essa è pari a $0,45$ per i crediti non subordinati ("senior") e $0,75$ per quelli subordinati (la definizione di subordinazione è abbastanza generica e rimessa all'autorità delle Banche Centrali). In caso l'operazione sia coperta da garanzia la perdita effettiva è calcolata nel seguente modo:

$$lgd = lgd(\text{senza garanzia}) \cdot (\overline{E} - E)$$

Il primo fattore è uguale a lgd in presenza di posizione non garantita per crediti non subordinati $(0,45)$, E è pari al valore corrente dell'esposizione e $\overline{E}$ è pari al valore dell'esposizione al netto dell'effetto della garanzia Se si utilizza il metodo IRB invece si utilizza il seguente schema: in presenza di garanzia le esposizioni per cui il rapporto tra valore della garanzia C e valore dell'esposizione E è inferiore ad una soglia $\overline{C}$ stabilita per ogni garanzia si applica la lgd pari a $0,45$. Le esposizioni per le quali il rapporto tra C ed E supera una seconda soglia $\overline{C}_1$ si applica lgd in base alla seguente tabella, in relazione al tipo di garanzia:

Garanzia	lgd minima	$\overline{C}$	$\overline{C}_1$
Garanzia finanziaria idonea	0	0	n.a.
Crediti commerciali	$0,35$	0	$1,25$
CRE e RRE	$0,3$	0	$1,25$
Altre garanzie reali	$0,4$	$0,3$	$1,4$

Le esposizioni senior vanno suddivise in una quota interamente garantita e una non garantita; alla prima ($\frac{C}{\overline{C}_1}$) è applicata la lgd prevista dalla tabella, la parte restante è considerata non garantita e si applica lgd pari a $0,45$. Se la banca possiede i requisiti per accedere al metodo IRB avanzato può utilizzare stime interne di lgd, misurata in percentuale di *ead*. Per quanto riguarda l'esposizione al momento dell'inadempienza, questa non dovrebbe essere inferiore alla somma tra l'ammontare della riduzione del patrimonio regolamentare che avrebbe luogo in seguito a svalutazione totale ed eventuali accantonamenti specifici e svalutazioni parziali.

Per quanto riguarda la scadenza effettiva (m) questa, nelle esposizioni verso imprese, sarà pari a 2,5 anni. Per le banche che utilizzano il metodo IRB avanzato la stima può essere realizzata con strumenti interni, ma essa non può essere inferiore ad un anno ne superiore a 5 anni e deve rappresentare la vita residua espressa in anni, a meno che non si tratti di esposizioni a breve termine con scadenza inferiore ad un anno (ad esempio le transazioni sui mercati finanziari) la cui definizione spetta alle autorità di vigilanza. Tuttavia le autorità nazionali possono prevedere che, nei riguardi delle piccole imprese (quelle con fatturato dichiarato e patrimonio del gruppo

consolidato di cui fanno parte inferiore a 500000000 Euro) si proceda a determinare la scadenza effettiva come nel metodo base (2,5 anni)[11].

3 Pre-processing dei dati

In qualsiasi lavoro avente ad oggetto delle operazioni su un insieme di dati è opportuno fare le seguenti considerazioni prima di iniziare la fase operativa vera e propria:

- Qualsiasi insieme di dati può contenere dei dati mancanti. Le cause possono essere le più diverse: i dati possono essere stati considerati irrilevanti e non registrati, possono essere stati perduti, i valori possono essere non noti oppure non calcolabili. Bisogna capire se l'assenza di questi dati è casuale oppure può rivestire un certo significato[12].
- Qualsiasi insieme di dati può contenere dei dati inaccurati a causa di errori di trascrizione\rilevazione[13], errori arbitrari (trucchi contabili), ma anche semplicemente al fatto che spesso i dati non sono stati creati per essere analizzati successivamente e ciò comporta una minore attenzione nell'immagazzinamento.
- Per questi motivi bisogna prestare molta attenzione alla natura dei dati ed eliminare eventuali inconsistenze, ma per gli stessi motivi bisogna sempre tenere in considerazione l'ipotesi che il dato apparentemente corretto su cui si sta lavorando potrebbe essere errato.

3.1 Outlier

Un *outlier* può essere definito come un dato completamente differente o inconsistente dall'insieme di dati a cui appartiene. Come si può notare tale definizione è alquanto generica, per cui la ricerca degli outliers è una pratica estremamente soggettiva[14].

La presenza di outlier in un insieme di dati può avere origini differenti: può essere dovuto ad errori ed inaccuratezze (ne abbiamo parlato precedentemente), alla

[11] Per maggiori approfondimenti si consulti il documento ufficiale del nuovo accordo di Basilea [29].

[12] Nella nostra analisi ciò è utile quando si nota l'assenza di dati dovuta al fatto che il dato in questione è il risultato di una divisione per 0.

[13] Nella nostra analisi ciò accade quando il valore di una istanza in un attributo non rientra nel suo intervallo teorico di variazione.

[14] Si può dare una definizione più rigorosa dicendo che una istanza o è outlier con parametri p e d in un insieme S se almeno il $p\%$ delle istanze dell'insieme è lontano almeno d da o, ma anche in questa definizione c'è l'elemento soggettivo dettato dalla determinazione di p e d. Esiste una branca del *data-mining* chiamata *outlier mining* che si occupa della ricerca degli outlier; non è questo lo scopo del presente lavoro per cui rimandiamo alla letteratura specializzata per eventuali approfondimenti.

variabilità intrinseca nei dati oggetto dell'analisi, oppure a situazioni "patologiche". Quest'ultimo caso è da ricondurre ad azioni deliberate compiute da chi ha creato il dato in discussione ed è il tipico caso delle politiche di bilancio che vengono attuate per motivi fiscali. Quando ci si trova in presenza di potenziali outlier si possono seguire due strade: eliminare l'outlier dall'insieme di dati oppure conservarlo; la scelta della strada da seguire dipende dalle finalità. Se si vuole creare un modello in grado di riconoscere gli esempi a disposizione, o comunque che sia in grado di lavorare su dati ritenuti normali (ad esempio nella previsione di serie storiche future in cui si vuole riconoscere il pattern della serie storica in condizioni normali) è conveniente eliminare gli outlier, ad esempio applicando ai dati una distribuzione normale ed eliminando i dati relativi alle code, utilizzando una opportuna misura della deviazione standard. Se invece si vuole operare anche in presenza di valori eccentrici, in quanto questi possono essere significativi del fenomeno oggetto dell'osservazione, è conveniente conservare gli outlier nella base di dati. Questo è il caso del problema oggetto della nostra osservazione. Nei dati a nostra disposizione abbiamo modo di osservare valori di bilancio che in alcuni casi risultano essere molto differenti dagli altri. Nei dati relativi al rapporto tra *Capitale Circolante Netto* e *Totale Attivo* abbiamo notato valori pari a -78,75, rispetto ad una media di $-0,06$ ed una deviazione standard di $0,28$. Questo valore può stare a significare che le passività a breve dell'impresa sono molto maggiori rispetto all'attivo circolante e possono essere significative rispetto all'analisi delle condizioni che possono influenzare la mancata restituzione del credito, considerando che è molto più probabile una mancata restituzione in presenza di condizioni patologiche dell'attività aziendale rispetto a condizioni di normale operatività aziendale. Conservare questi valori può significare però falsare il significato degli altri nel caso in cui si operi una normalizzazione, a causa dell'accuratezza degli strumenti a disposizione: la media dell'attributo *Vendite/Magazzino* è pari a $77,95$; tuttavia una istanza presenta un valore molto più alto rispetto alla media (1877), che indica che probabilmente l'azienda ha operato una grande operazione di dismissione del magazzino (liquidazione). Questa operazione può essere da ricollegare ad una situazione di crisi appena precedente alla chiusura dell'attività (non a caso l'azienda in questione non ha onorato il debito). La situazione quindi è degna di nota e può essere indicativa di futuri default, ma includerla nei dati a disposizione con una normalizzazione min-max (si veda oltre) ha portato diversi campi ad assumere valore pari o molto prossimi a zero. Ancora più significativo è il caso dell'attributo *Passività a breve/fatturato*. Il range di variazione è $[0, 196.5]$. Il valore 0 denoterà assenza di passività a breve e probabilmente una condizione di buona operatività aziendale; viceversa il valore $196,5$ indica passività a breve che non possono essere estinte grazie al normale flusso operativo determinato dalle vendite. In questo attributo la media è pari a $3,26$ e l'utilizzo della normalizzazione min-max porta ad un incredibile appiattimento verso lo 0 di tutti gli altri dati. Cosa fare in questi casi? Cambiare normalizzazione degli attributi? Eliminare il dato? Sono questi i problemi

che si presentano nella fase di pre-processing dei dati.

3.2 Normalizzazione dei dati

Con la normalizzazione si vogliono uniformare i dati in modo che ricadano in un intervallo determinato. Questo risulta utile quando gli attributi che compongono i dati oggetto dell'analisi utilizzano misure diverse, oppure quando utilizzano la stessa misura che però, dovendo descrivere caratteristiche diverse, assume intervalli di variabilità differente a seconda dell'attributo utilizzato. In questi casi la normalizzazione porta i diversi attributi ad avere lo stesso peso (cioè al fatto che ognuno di essi non conti proporzionalmente più degli altri).

Nell'utilizzo delle reti neurali questo aspetto è più evidenziato dal fatto che generalmente si utilizzano funzioni di attivazione che riconducono l'output della rete ad un intervallo specifico ([0,1] oppure [-1,1]), per cui, utilizzando l'apprendimento supervisionato bisogna avere la cura di trasformare gli output desiderati per ricondurli all'intervallo in cui andranno a ricadere gli output effettivi della rete; inoltre anche quando non si utilizzano tali funzioni di attivazione è utile normalizzare i dati per evitare problemi computazionali, per soddisfare i requisiti di alcuni algoritmi e per facilitare l'apprendimento.

Esistono diversi modi per effettuare la normalizzazione dei dati. Uno dei metodi più diffusi è la *Zero-score normalization*, con cui i diversi dati vengono normalizzati secondo la seguente formula:

$$z_{ia} = \frac{x_{ia} - m_a}{s_a}$$

dove

z_{ia} = valore normalizzato dell'i-sima istanza dell'attributo a;
x_{ia} = valore non normalizzato dell'i-sima istanza dell'attributo a;
m_a = media dell'attributo a;
s_a = scarto assoluto medio dell'attributo a.

In alternativa si può utilizzare lo scarto quadratico medio dell'attributo in luogo di quello assoluto, ottenendo un attributo con media nulla e scarto quadratico medio unitario, ma ciò risulta essere molto sensibile agli outliers.
Con questo metodo non si può però ricondurre il campo di variazione ad un intervallo specifico, in quanto i valori assunti dipendono appunto da media, scarto assoluto e valore attuale dell'istanza. Questo problema è risolto se si utilizza la normalizzazione *Min-Max*[15]:

[15] La forma generale di questo tipo di normalizzazione è

$$z_{ia} = a + (b - a) \cdot \frac{x_{ia} - min_a}{max_a - min_a}$$

$$z_{ia} = \frac{x_{ia} - min_a}{max_a - min_a}$$

dove

z_{ia} = valore normalizzato dell'i-sima istanza dell'attributo a;
x_{ia} = valore non normalizzato dell'i-sima istanza dell'attributo a;
min_a = valore minimo assunto dall'attributo a;
max_a = valore massimo assunto dall'attributo a.

Questa normalizzazione risulta essere ancora più sensibile agli outliers. La funzione che permette di trattare meglio gli outliers è la logaritmica, che può essere definita dall'utente, permettendo quindi un grado di elasticità molto alto[16]. Un'altra forma di normalizzazione è quella semplice, ottenuta dividendo il valore dell'istanza su un determinato attributo per il massimo valore assunto da quell'attributo:

$$z_{ia} = \frac{x_{ia}}{max_a}$$

Studi sulla normalizzazione dei dati applicati alla fase di pre-utilizzo delle reti neurali (riportati in [44]) hanno dimostrato che, applicata a problemi di classificazione, la normalizzazione dei dati ha effetti benefici sulla percentuale di classificazioni corrette e sull'errore, ma rallenta l'apprendimento della rete ed i benefici riscontrati diminuiscono all'aumentare della dimensione della rete. Inoltre alcuni studi riconducono i dati ad intervalli quali $[0.1, 0.9]$ oppure $[0.2, 0.8]$ argomentando che le funzioni di attivazione non lineari dei neuroni raggiungono i valori 0 e 1 asintoticamente. A nostro avviso comunque, dato che l'accuratezza degli strumenti di calcolo fa in modo che tali valori vengano comunque raggiunti per argomenti della funzione con modulo molto grande[17], è preferibile non ricorrere a tali artificiosità e limitarsi agli intervalli di variazione standard.

Bisogna però tenere conto che non sempre si vogliono normalizzare i dati: la scelta dipende dalla natura dell'attributo e dall'individuazione di eventuali outliers, ma anche dagli scopi che si vogliono ottenere. Anche quando si desidera normalizzare può essere conveniente dare ad un attributo un peso maggiore che ad un altro, e su attributi diversi si possono operare normalizzazioni basate su funzioni diverse.

Come ultima considerazione vogliamo far notare che se si utilizzano dati normalizzati, l'output della rete sarà compreso nell'intervallo normalizzato, per cui si

che permette ai dati di assumere valori compresi nell'intervallo $[a, b]$.

[16] Unica accortezza dev'essere quella di fare in modo che l'argomento della funzione assuma valore ≥ 1.

[17] Abbiamo avuto modo di osservare questo comportamento già dalle prime prove effettuate sul nostro data-set e sul modello di rete.

dovrà operare un'operazione inversa per riportare gli output ai risultati voluti. Ciò può portare confusione nella determinazione della misura di performance: in molti studi non è indicato se la misura di performance è relativa a dati normalizzati o a quelli effettivi.

3.3 Descrizione dei dati a disposizione

I dati che abbiamo a disposizione nel nostro data set sono reali e si riferiscono al triennio 2001-2003. Sono relativi ad un insieme di piccole e medie imprese (PMI) individuate da un numero progressivo e suddivise in due categorie: quelle che hanno restituito il credito concesso dalla banca e che per questo sono state donotate con la dicitura *bonis* e quelle che non hanno restituito il credito e perciò denotate con *default*. Ogni azienda è contrassegnata da:

- un numero progressivo;
- l'indicazione della forma giuridica dell'impresa;
- il codice *RAE*, consistente in una codifica a tre cifre definita dalla Banca d'Italia che identifica il ramo dell'attività economica dell'azienda;
- il codice *SAE*, consistente in un'altra codifica a tre cifre definita dalla Banca d'Italia definita per classificare le imprese per settore dell'attività economica (quindi in relazione a beni e servizi prodotti)[18].

Per ogni azienda ci sono stati forniti i principali indicatori relativi al modello di bilancio, al modello Centrale dei Rischi ed a quello andamentale. Sono gli indicatori che sono ritenuti maggiormente predittivi dalla banca in questione e che perciò vengono utilizzati nei loro modelli di previsione del rischio di credito. Uno degli scopi iniziali del nostro lavoro era quello di confrontare il modello di rischio di credito da noi elaborato con il classico modello Z-SCORE di Altman descritto in precedenza. Purtroppo questo confronto si è reso impossibile perché i dati a nostra disposizione non includono tutti gli indicatori utilizzati da Altman. Gli indicatori di bilancio presenti nel nostro data-set sono i seguenti:

- $\frac{\text{Cash Flow}}{\text{Debiti Totali}}$,
  dove il Cash flow è determinato dalla sommatoria tra utile ed ammortamenti di immobilizzazioni materiali ed immateriali, alla quale devono essere sottratti gli accantonamenti e la svalutazione del capitale circolante[19];

[18] Questa codifica è stata elaborata in seguito alle raccomandazioni dell'Eurostat nel 1995 definite per uniformare le diverse catalogazioni esistenti in differenti stati.

[19] In senso lato indica la variazione delle liquidità immediate determinata dalla gestione, ma nel nostro data set si considera il cash flow operazionale, che rappresenta il flusso monetario derivante

- $\frac{\text{Fatturato}}{\text{Valore del magazzino}}$
  Questo indicatore può essere soggetto a cattive interpretazioni, in quanto il fatturato può essere molto differente a seconda dei settori di appartenenza economica e i dati a disposizione non ci consentono di sapere con che metodologia è stato contabilizzato il magazzino;

- $\frac{\text{Passività correnti}}{\text{Fatturato}}$
  Questo indicatore misura la capacità di generare flusso di cassa per estinguere i debiti correnti;

- $\frac{\text{Patrimonio netto}}{\text{Totale attività}}$,
  dove il patrimonio netto è rappresentato dalla differenza tra attività e passività e rappresenta la fonte di finanziamento interna dell'impresa;

- $\frac{\text{Oneri finanziari}}{\text{Debiti Totali}}$,
  che indicano il peso degli interessi passivi e delle spese, connessi ai finanziamenti esterni ricevuti dall'azienda, su quelli totali (in cui rientrano le posizioni debitorie originate dalla gestione caratteristica);

- $\frac{\text{Capitale circolante netto}}{\text{Totale attivo}}$,
  dove il Capitale circolante netto è dato dalla differenza tra attivo circolante[20] ed il passivo corrente;

- $\frac{\text{Crediti verso clienti}}{\text{Fatturato}}$,
  che indica se le operazioni di vendita hanno portato ad un flusso di cassa oppure al sorgere di posizioni creditorie. Valori superiori ad 1 mostrano aziende che non riescono a riscuotere i crediti originati dalla gestione;

- $\frac{\text{Valore Aggiunto}}{\text{Totale attivo}}$,
  dove il valore aggiunto rappresenta la nuova ricchezza prodotta dall'azienda, cioè il contributo alla valorizzazione delle risorse acquistate presso terzi, ed è dato dalla sommatoria delle remunerazioni dei fattori produttivi utilizzati dall'azienda[21].

dalle operazioni di esercizio, con esclusione quindi da flussi derivati da investimenti, finanziamenti e distribuzione di utili, rimborsi di capitale ed aumenti di capitale a pagamento.

[20] Beni destinati alla vendita o al rapido utilizzo nel processo produttivo.

[21] Può essere calcolato anche come differenza tra il valore delle risorse che l'azienda cede a terzi ed il valore delle risorse che acquista da terzi.

Per ogni azienda sono poi presenti 4 campi relativi al modello "Centrale dei Rischi". Questa è un sistema informativo gestito dalla Banca d'Italia contenente le informazioni sui crediti concessi dalle Banche ai clienti. Secondo disposizioni della Banca d'Italia[22] gli intermediari finanziari sono tenuti a comunicare alla Banca d'Italia i dati relativi alle posizioni creditorie nei confronti della clientela, in modo da creare un Sistema Informativo (la Centrale dei Rischi appunto) per fornire alle banche stesse delle informazioni sulla clientela per una eventuale valutazione del rischio di credito al fine di migliorare la gestione degli impieghi e di perseguire la stabilità del sistema creditizio e finanziario. Per ogni cliente viene definita quindi una situazione di indebitamento complessivo verso il sistema creditizio[23]. I campi presenti nel nostro data-base sono:

- $\frac{\text{Sconfinamento a breve termine}}{\text{Fido accordato a breve termine}}$;
- $\frac{\text{Sconfinamento a medio-lungo termine}}{\text{Fido accordato a medio-lungo termine}}$;
- $\frac{\text{Fido utilizzato a breve termine}}{\text{Fido accordato a breve termine}}$;
- $\frac{\text{Fido utilizzato a medio-lungo termine}}{\text{Fido accordato a medio-lungo termine}}$.

Il fido utilizzato consiste nel credito erogato o alle garanzie prestate al cliente; quello accordato consiste nel fido che gli organi competenti dell'intermediario hanno deciso di concedere al cliente; lo sconfinamento è pari alla differenza tra i due termini. In base alla normativa i dati di sconfinamento, utilizzato e accordato non sono relativi all'istituto da cui provengono i dati ma al sistema complessivo.

Sono presenti poi i dati relativi all'andamento del rapporto con la banca (da cui il nome di modello andamentale), con i seguenti campi:

- $\frac{\text{Fido utilizzato}}{\text{Fido accordato}}$;
- $\frac{\text{Quantità insoluti}}{\text{Quantità presentazioni salvo buon fine}}$
  rappresenta il numero di ricevute bancarie (e/o di altri effetti) presentate dal cliente e ritornate insolute divise per il numero totale delle presentazioni;
- $\frac{\text{Valore insoluti}}{\text{Valore presentazioni salvo buon fine}}$.
  rappresenta il valore delle ricevute bancarie (e/o di altri effetti) presentate dal cliente e ritornate insolute divise per il valore totale delle presentazioni.

[22] Circolare n.140 del 11 febbraio 1991.

[23] Questa posizione potrebbe essere inferiore alla esposizione complessiva dell'azienda rispetto al sistema creditizio complessivo in quanto per alcune banche di piccole dimensioni non sussiste alcun obbligo di comunicazione.

3.4 Il trattamento dei dati mancanti

Per quanto riguarda i dati mancanti abbiamo deciso di trattarli ad operazioni differenti a seconda della natura dell'assenza: in caso di assenza derivante da mancata rilevazione oppure da inconsistenza dei dati abbiamo deciso di utilizzare, per il dato in questione la media; in caso di assenza derivante da errore di calcolo (divisione per zero) abbiamo deciso, una volta normalizzato l'attributo, di sostituire il dato con uno degli estremi dell'intervallo di variazione, a seconda del tipo di normalizzazione utilizzata.

Dopo aver deciso queste linee guida abbiamo dovuto vagliare diverse ipotesi per il trattamento dei dati mancanti: in caso di assenza derivante da errore di calcolo non si prensentano problemi, mentre più complessa è apparsa la gestione dell'assenza derivante da mancata rilevazione oppure da inconsistenza dei dati. In questo caso infatti abbiamo detto che l'approccio generale consiste nell'utilizzare, per il dato in questione la media[24]. Questo ci ha portato a scegliere tra tre opzioni di sostituzione:

- sostituire il valore mancante per l'azienda in questione con la media dei valori assunti dall'azienda per gli anni in cui la rilevazione è presente (*completamento per riga*);
- per ogni anno sostituire i valori mancanti con la media dei valori assunti dall'attributo in questione in quello specifico anno (*completamento per colonna*);
- per ogni attributo sostituire i valori mancanti con la media dei valori assunti dall'attributo in questione in tutti gli anni (anche questo è un *completamento per colonna*).

Una prima osservazione dei dati ci ha portato a scartare il primo modo di procedere in quanto abbiamo visto che generalmente se una azienda presenta dei dati mancanti li presenta per tutti e tre gli anni. Tra le due opzioni rimaste abbiamo scelto di utilizzare l'ultima perchè abbiamo preferito associare un valore unico per ogni attributo alla situazione di mancanza di informazione: facendo affidamento sulla capacità di *feature detection* della rete, potremmo pensare che la rete apprenda che quello specifico valore è derivante da mancanza di informazione. Probabilmente la rete non è in grado di ricavare questo tipo di informazione, ma ricordiamo che questo procedimento è adottato solo per preservare potenziali informazioni utili nascoste negli altri attributi consentendo l'elaborazione.

Una volta deciso questo, per il procedere del trattamento dei dati, siamo di fronte a tre possibilità. Sappiamo che i dati devono essere normalizzati per essere forniti in ingresso alla rete. Questo porta all'utilizzazione congiunta della sostituzione dei dati mancanti e della normalizzazione. Le due operazioni possono essere integrate nei seguenti modi:

[24] Parliamo di media aritmetica.

1. sostituire prima i valori mancanti per inconsistenza o rilevazione assente con la media, poi normalizzare e, se sono presenti dati mancanti a causa di divisione per zero, sostituire il dato con uno degli estremi dell'intervallo di variazione;

2. normalizzare i dati presenti (quindi non trattare ancora i dati mancanti a causa di divisione per zero), poi sostituire i valori mancanti per inconsistenza o rilevazione assente con la media dell'attributo normalizzato. Procedere poi al trattamento dei dati mancanti a causa di divisione per zero, sostituendoli con uno degli estremi dell'intervallo di variazione a seconda della normalizzazione utilizzata;

3. prima normalizzare i dati presenti, questa volta trattando però anche i dati mancanti a causa di divisione per zero sostituendoli con uno degli estremi dell'intervallo di variazione, poi sostituire i valori mancanti per inconsistenza o rilevazione assente con la media dell'attributo normalizzato *considerando per il calcolo anche i valori corrispondenti ad uno degli estremi derivati dalla precedente sostituzione.*

In realtà l'esiguità del database a disposizione non ci aiuta nella scelta della operazione da affettuare. Abbiamo effettuato delle prove addestrando delle reti con le tre tipologie di sostituzione proposte, ma i risultati a cui queste pervengono sono gli stessi. Probabilmente disponendo di data-base più ampi sarebbe proponibile uno studio volto a valutare le prestazioni al variare dei tre modi di procedere. Per i nostri esperimenti però scegliamo di procedere nel primo dei tre modi proposti.

3.5 Considerazioni sui dati in esame

La prima considerazione effettuata in merito ai dati in esame riguarda la completezza delle osservazioni periodiche per ogni azienda: abbiamo a disposizione per tutte le aziende le 3 rilevazioni relative agli anni in considerazione: si tratta di una considerazione positiva in quanto elimina in partenza l'eventuale problema di addestrare la rete con esempi diversi dal punto di vista della dinamica temporale[25].

Le cose invece non vanno così bene se prendiamo in considerazione i singoli attributi. Questi verranno ora esaminati e saranno messe in evidenza alcune situazioni particolari.

- L'attributo $\frac{\text{Cash Flow}}{\text{Debiti totali}}$ non ha teoricamente limiti superiore ed inferiore, ma ci risulta difficile pensare che una azienda abbia un flusso di cassa maggiore dei debiti totali (e quindi $\frac{\text{Cash Flow}}{\text{Debiti totali}} > 1$), in quanto il flusso di cassa è il maggior determinante delle disponibilità finanziarie che serviranno per estinguere

[25] Tale problema si potrebbe presentare quando disponiamo soltanto dei dati relativi a 1 o 2 anni, perché l'azienda potrebbe essere fallita oppure perché l'apertura è recente.

le posizioni debitorie (si potrebbe ipotizzare in un caso simile che l'azienda in questione abbia per la maggior parte debiti a lungo termine e per questa ragione non utilizzi le disponibilità determinate dal flusso di cassa per farvi fronte). Anche se non sono presenti casi del genere nel nostro data-set, proponiamo, in vista di utilizzi futuri, una operazione di *data-smoothing* su dati che dovessero presentare questa particolarità, considerandone soltanto la parte intera 1. D'altra parte questo attributo presenta valori negativi[26] per la maggior parte in corrispondenza dei casi di default. Sono pochi i casi in cui il valore negativo risulta essere inferiore a -1. Riteniamo che già un flusso di cassa negativo sia sintomatico di crisi, per cui i valori inferiori a -1 possono essere smussati alla loro parte intera (-1). Sebbene questa operazione restringa l'intervallo di variabilità dei valori, questi non risultano essere ancora adeguati allo scopo di essere forniti alla rete, in quanto decidiamo da ora di utilizzare neuroni con ingressi compresi tra 0 e 1.

- Nell'attributo $\frac{\text{Vendite}}{\text{Magazzino}}$ i dati a disposizione presentano spesso una divisione per 0. Precedenti lavori considerano questa situazione un errore ed evitano di considerare il valore prodotto, eliminando addirittura in alcuni casi l'attributo in considerazione. Noi proponiamo, invece, di conservare questo attributo: il fatto che il magazzino sia nullo può derivare sia da una situazione in cui l'intero magazzino è stato dismesso (ad esempio in caso di liquidazione aziendale precedente alla cessazione dell'attività) oppure da particolari politiche di magazzino adottate dall'impresa (*Just in Time*), oppure ancora dall'assenza del magazzino in imprese per la prestazione di servizi (quindi dal settore e dal ramo dell'attività economica). Il fatto che in ultima analisi questo fenomeno si distribuisce in maniera abbastanza equilibrata tra le aziende che si rivelano essere solventi e non considerando l'intera dinamica del rapporto (20 contro 17) può essere significativo delle diverse motivazioni che portano al suo verificarsi. Per cui proponiamo di normalizzare i dati in modo da considerare tali situazioni e di assegnare loro uno dei limiti dell'intervallo proposto. Il campo di variazione di questo attributo è molto ampio ($[0, 638.4]$) e la media delle osservazioni è molto spostata verso il limite inferiore ($70,2$). Questo porta all'impossibilità dell'utilizzo della normalizzazione Min-Max, in quanto abbiamo riscontrato un incredibile appiattimento dei valori verso lo 0 in caso di utilizzo di questo tipo di normalizzazione. Abbiamo proceduto operando una trasformazione logaritmica dell'attributo, eliminando il valore più alto (considerandolo outlier e operando come se si trattasse del caso della divisione per 0) e procedendo a normalizzare l'attributo con una funzione logaritmica a base[27] 600, avendo

[26] I debiti totali non possono essere mai negativi, per cui l'unica causa di una tale situazione è da ricondurre a flusso di cassa negativo.

[27] La base del logaritmo è scelta in tutti gli attributi in modo da essere vicina al massimo attuale.

cura di sommare 1 all'argomento della funzione, in modo che il risultato sia sempre ≥ 0. La formula utilizzata è la seguente:

$$\overline{x} = log_{600}(x+1)$$

- Anche nell'attributo $\frac{\text{Passività a breve}}{\text{Fatturato}}$ sono presenti dei campi in cui è indicato che l'operazione relativa è impossibile in quanto consiste in una divisione per 0. Anche in questo caso decidiamo di conservare questi dati e di considerarli pari al valore massimo dell'intervallo normalizzato di riferimento (1). Anche in questo caso abbiamo deciso di utilizzare una trasformazione logaritmica. Il problema che si è posto è simile al precedente: alta variabilità dell'attributo con massimo molto elevato e media vicina all'estremo inferiore di variazione. Abbiamo utilizzato una normalizzazione simile alla precedente, utilizzando un logaritmo a base 1000 e riportando ad 1 i valori superiori a tale limite:

$$\overline{x} = log_{1000}(x+1)$$

- Il limite superiore dell'attributo $\frac{\text{Patrimonio Netto}}{\text{Totale attivo}}$ è 1, tuttavia nei dati relativi ad un cliente (71) si riscontrano per i 3 anni valori superiori a tale limite, per cui questi dati sono da considerare errati. In questi casi l'idea più intuitiva sarebbe quella di eliminare dal data-set tutte le osservazioni relative a quel cliente. Tuttavia, considerando anche che non disponiamo di un data-set di dimensioni grandissime, abbiamo preferito non eliminarle, in quanto potrebbero andare perdute informazioni rilevanti rispetto agli altri attributi, per cui abbiamo semplicemente sostituito i valori inconsistenti con la media, in modo da non alterare l'analisi. Anche in questo attributo sono presenti campi in cui la rilavazione dell'indicatore porta ad una divisione per 0; anche in questo caso abbiamo trasformato questi dati in modo che una volta normalizzati assumessero valore uguale ad un estremo[28]. Abbiamo inoltre riscontrato una certa corrispondenza tra default e valore negativo su quest'attributo: valori negativi con modulo molto alto sono sempre associati a default, mentre valori negativi con modulo relativamente basso hanno un comportamento leggermente meno uniforme (la maggioranza dei casi è comunque sempre da ricondurre al default). Questo attributo presenta un limite superiore (1) ma non è limitato inferiormente. Abbiamo operato in questo caso una trasformazione speculare della funzione, prima di procedere alla trasformazione logaritmica (base 10). Per cui abbiamo utilizzato la seguente funzione:

$$\overline{y} = log_{10}(1+1-y)$$

[28]Inizialmente si è pensata una normalizzazione Min-max ma l'applicazione successiva di una funzione speculare ci ha portati a considerare tali valori come corrispondenti al valore minimo 0.

L'aggiunta di una unità (-1 è il limite inferiore della negazione della funzione) serve a delimitare il quadrante in cui si sviluppa la funzione. L'aggiunta di una ulteriore unità è necessaria affinchè l'argomento della funzione non assuma mai valore ≤ 1.

- L'attributo $\frac{\text{Oneri finanziari}}{\text{Debiti Totali}}$ non necessita di normalizzazione in quanto il suo intervallo di variazione è $[0, 1]$, per cui i dati forniti sono già pronti per essere utilizzati dalla rete neurale.

- Considerazioni analoghe ad attributi precedenti valgono per l'attributo $\frac{\text{Capitale circ netto}}{\text{Totale attivo}}$: anche qui esistono campi in cui è impossibile determinare il valore dell'attributo perché è il risultato di una divisione per 0, anche qui il limite superiore è pari ad 1 e quello inferiore (almeno in via teorica) a $-\infty$ ed anche qui operiamo una trasformazione logaritmica.

- L'attributo $\frac{\text{Crediti verso clienti}}{\text{Fatturato}}$ presenta limite inferiore pari a 0 ed operiamo una trasformazione logaritmica:

$$\overline{y} = log_{15}(1 + y)$$

in modo da far variare i valori assunti tra 0 e 1.

- Per l'attributo $\frac{\text{Valore Aggiunto}}{\text{Totale attivo}}$ abbiamo operato una trasformazione speculare della funzione, applicando poi una normalizzazione logaritmica a base 3 dopo aver fatto in modo che la funzione si estendesse soltanto nel primo quadrante in analogia con il metodo usato per gli attributi prima menzionati.

- L'attributo $\frac{\text{Sconfinamento a breve termine}}{\text{Fido accordato a breve termine}}$ presenta dei campi con valori negativi. Questo è impossibile per la natura stessa dell'attributo, in quanto entrambi i termini del rapporto non possono essere negativi. Abbiamo deciso di eliminare questi valori ritenuti inconsistenti e li abbiamo sostituiti con la media trattandoli come i dati mancanti. Lo stesso procedimento è stato riservato all'attributo
$\frac{\text{Fido utilizzato a breve termine}}{\text{Fido accordato a breve termine}}$.

Se per gli indicatori di bilancio sono state necessarie alcune correzioni solo per individuare dati non corretti perché non rientranti nell'intervallo teorico di variabilità dell'attributo e per eliminare l'incidenza di particolari outlier troppo diversi dal resto della popolazione, per gli indicatori relativi al modello *Centrale dei Rischi* e al modello *andamentale* vanno fatte alcune considerazioni. Si ha subito modo di osservare che il numero di osservazioni per ogni attributo è sensibilmente inferiore non solo rispetto al totale di casi osservati (318), ma anche rispetto al numero di rilevazioni corrette per ogni altro attributo relativo

al modello di bilancio. Infatti, per i 7 attributi relativi a questi due modelli, il numero di osservazioni annue è pari a 198, 238, 199, 238, 291, 39 e 39. Nelle osservazioni relative al modello di bilancio abbiamo evitato di eliminare i dati relativi a osservazioni incomplete sostituendo il dato mancante a causa di errori di rilevazione o di mancata comunicazione con la media delle osservazioni relative a quell'attributo. Se operassimo tale modo di procedere con questi attributi potremmo falsare le conclusioni ottenute dalla nostra rete neurale. A tal proposito menzioniamo che generalmente la suddivisione tra training set e testing set[29] viene effettuata secondo alcune proporzioni, la più comune delle quali è $[75\%, 25\%]$. Per gli attributi rientranti in questi modelli che presentano un totale delle osservazioni inferiore ad $\frac{1}{4}$ del totale delle osservazioni, sostituire le informazioni mancanti con la media potrebbe voler significare, nella più malaugurata delle ipotesi, far rientrare tutte le osservazioni corrette in uno dei due sottoinsiemi utilizzati, con conseguenze disastrose dal punto di vista della generalizzabilità della rete e della valutazione della performance. Per questo motivo decidiamo di non utilizzare gli attributi che presentano un numero di osservazioni inferiore al 75% di quelle totali, quindi eliminiamo dalla nostra analisi gli attributi

1. $\frac{\text{Sconfinamento a breve termine}}{\text{Fido accordato a breve termine}}$
2. $\frac{\text{Fido utilizzato a breve termine}}{\text{Fido accordato a breve termine}}$
3. $\frac{\text{Quantità insoluti}}{\text{Quantità presentazioni salvo buon fine}}$
4. $\frac{\text{Valore insoluti}}{\text{Valore presentazioni salvo buon fine}}$

- Per l'attributo $\frac{\text{Fido utilizzato}}{\text{Fido accordato}}$ (l'unico attributo considerato del modello andamentale) abbiamo operato una trasformazione logaritmica

$$\overline{y} = log_{14}(1 + y)$$

anche in questo caso abbiamo deciso di prendere in considerazione i campi vuoti derivanti da errori a causa di divisione per 0 e vi abbiamo assegnato valore 1 dopo la normalizzazione.

3.6 Ipotesi di utilizzo degli attributi RAE e SAE

Abbiamo già detto che le codifiche *RAE* e *SAE* sono utilizzate per indicare il ramo di appartenenza economica e il settore di appartenenza economica. L'utilizzo arbitrario di questi attributi può essere pericoloso, se non addirittura fuorviante. Queste

[29] A volte viene utilizzato un terzo insieme, chiamato *validation set* per evitare l'overfitting e per decidere il punto di stop-learning.

codifiche servono per tradurre una "descrizione" dell'appartenenza dell'azienda in questione in un numero[30]. Possiamo ben capire che il criterio di scelta del numero utilizzato segue nient'altro che il criterio progressivo della numerazione, per cui la normalizzazione dei valori assunti da questi attributi non sembrerebbe rispondere a nessuna aspettativa o esigenza[31] e l'utilizzo di questa procedura porterebbe soltanto a delle ulteriori inconsistenza nei dati in esame.

Piuttosto che normalizzare i dati relativi a questi attributi al pari di tutti gli altri attributi esaminati fino ad ora si potrebbe utilizzare un neurone per ogni possibile valore assunto da questi attributi in modo che per ogni istanza presentata in ingresso alla rete si attivino (assumano valore pari a 1) soltanto i neuroni corrispondenti al RAE ed al SAE in questione, in maniera molto simile a quanto avviene per le variabili *dummy* in econometria[32]. In questo modo saremmo in grado di incorporare l'informazione qualitativa presente nel data-base nella nostra analisi; tuttavia un simile modo di operare ci ha posto di fronte dei problemi:

- l'attributo *RAE* assume nei dati in esame circa 50 valori diversi. Inoltre pochi sono i valori che si riscontrano in più di una o due aziende. Considerando che il nostro data-set si compone di 106 aziende, è facile comprendere come non è possibile produrre un buon comportamento della rete neurale utilizzando il comportamento pseudo-dummy con queste premesse: i neuroni di input sono troppi e gli esempi a disposizione sono troppo pochi per avere un apprendimento adeguato, mentre la fase di generalizzazione non porta a risultati utilizzabili per i valori del RAE poco "presenti" nel data-set.

- l'attributo *SAE* assume nei dati in esame 7 valori diversi. Questa situazione appare più virtuosa di quella dell'attributo precedente: aggiungere 7 neuroni in ingresso alla rete è una operazione facile da implementare e fornisce miglioramenti nella fase di addestramento della rete. L'inconveniente cruciale si presenta nella fase di utilizzo della rete, quando si fornisce in input una istanza caratterizzata da un codice *SAE* non incontrato durante la fase di addestramento. In questo caso il dato sarebbe inconsistente e non utilizzabile dalla rete.

- la scelta di come trattare gli attributi in esame deve essere rapportata alle finalità che si intende perseguire con l'utilizzo della rete neurale. Se si vuole creare

[30]Sappiamo che, almeno in linea teorica, la rete neurale è in grado di trattare con dati non numerici.

[31]Su questo tipo di dati non è possibile confrontare i valori con gli operatori relazionali $<$, $>$, $\leq$, $\geq$, non è possibile definire un ordine, non esiste il concetto di distanza; l'unica operazione possibile è data dal confronto per determinare se 2 valori sono uguali.

[32]In econometria le variabili *dummy* sono variabili binarie utilizzate per descrivere le informazioni qualitative. Per maggiori informazioni si veda [42], cap 7.

una rete in grado soltanto di riconoscere gli esempi con i quali viene "alimentata" si può tranquillamente optare per la scelta operata al punto precedente[33]. Se invece si vuole creare una rete in grado di operare su dati mai incontrati in precedenza bisogna introdurre un neuroni di input *per ogni possibile valore o comunque per il sottoinsieme di valori tipici del campo di analisi del problema in oggetto* e questo porta all'esigenza di avere un data-set sufficientemente ampio.

Per questi motivi decidiamo di non utilizzare i due attributi *RAE* e *SAE*.

3.7 Anomalia nella serie storica

Da una analisi approfondita dei dati abbiamo osservato che, in relazione ai dati derivanti dal modello di bilancio, diverse aziende[34] presentano dati identici negli ultimi due anni; le aziende 6 e 16 presentano addirittura dati identici nei tre anni considerati. Questa situazione può derivare o da errori di rilevazione da parte della banca[35] oppure da errori contabili più o meno volontari da parte delle aziende, anche se ci sembra strano che 30 aziende abbiano deciso di perseguire la stessa "politica di bilancio" in due anni. Tutto ciò, unitamente alle considerazioni fatte in precedenza circa la non plausibilità di alcuni valori assunti da particolari attributi, pone dei problemi riguardo alla correttezza dei dati: non siamo ancora in grado di trarre delle conclusioni, ma i dati che ci sono stati forniti appaiono inconsistenti. E se siamo in grado di dire che alcuni valori sono errati perché non rientrano nell'intervallo di variabilità di un attributo, probabilmente ce ne saranno altri errati nel data-set che non ci è possibile individuare perché invece vi rientrano, ma sono derivati egualmente da errori di rilevazione o di calcolo.

Abbiamo bisogno dei risultati forniti dalla rete neurale per vedere se le nostre illazioni sono corrette o meno[36].

3.8 Analisi di correlazione delle variabili

Iniziamo la fase sperimentale calcolando la correlazione tra le variabili: come detto in precedenza, conviene eliminare dall'analisi le variabili che presentano una correlazione molto intensa tra loro. Per questa analisi è stato usato il software *gretl* (GNU

[33] Questa finalità ci sembra interessante solo per scopi divulgativi o teorici.

[34] Ne abbiamo rilevate 30, e precisamente quelle contrassegnate dai numeri 8, 13, 14, 15, 27, 28, 30, 33, 39, 40, 42, 47, 51, 52, 57, 58, 61, 66, 67, 68, 69, 70, 71, 72, 73, 74, 76, 77, 78, 79.

[35] Ci sembra quantomeno improbabile che 30 aziende presentino valori di bilancio identici in due anni consecutivi.

[36] Una delle caratteristiche che vengono più spesso celebrate, ma non sempre valutate correttamente, delle reti neurali è quella di poter operare con dati incompleti o affetti da rumore: date le premesse, con il nostro lavoro abbiamo la possibilità di valutare se questa caratteristica è efficace o meno.

Regression, Econometric and Time-series Library). La prima correlazione è stata effettuata sui dati grezzi, non analizzati e non normalizzati ed i risultati sono esposti nelle tabelle successive[37]:

	C. F. / Deb Tot	Vendite / Magazzino	Pass corr / Fatturato	Pat netto / Tot att
Cash Flow / Debiti Tot	1,0000	-0,0005	-0,0684	0,1567
Vendite / Magazzino		1,0000	-0,0352	0,0848
Pass corr / Fatturato			1,0000	-0,1105
Pat net / Tot att				1,0000

	Oneri fin / Debiti Tot	C.C. netto / Tot attivo	Cr cl / Fatt	V. Agg. / Tot. att.	Sconf br / Fido br
Cash Flow / Deb Tot	-0,0896	0,0160	-0,0343	0,3857	-0,0492
Vendite / Magaz	0,0234	-0,3408	-0,0574	0,0497	0,0402
Pass corr / Fatt	0,0381	-0,0022	-0,0102	0,1041	-0,0628
Pat netto / Tot att	-0,0024	-0,0811	0,0199	0,1547	0,0804
Oneri fin / Deb Tot	1,0000	0,0623	0,0343	0,1419	-0,0228
Cc netto / Tot att		1,0000	0,0252	0,1111	0,0101
Cr cl / Fatt			1,0000	-0,0978	0,0007
Val Agg / Tot att				1,0000	-0,0539
Sconf br / Fido br					1,0000

[37] I dati mancanti sono stati saltati.

	Sconf ML / Acc ML	Util B / Acc B	Util ML / Acc ML	Util / Acc	Q insoluti / Q sbf
Cash Flow / Debiti Totali	-0,1343	-0,0598	-0,0165	-0,0734	-0,2213
Vendite / Magazzino	-0,0190	0,0152	-0,0112	0,0077	0,9583
Pass corr / Fatturato	0,2536	-0,0267	0,1283	0,0376	-0,3215
Pat netto / Tot att	-0,0014	-0,0093	0,0182	-0,0821	0,3532
Oneri fin / Debiti Tot	-0,0196	-0,0372	-0,0921	0,0105	0,6972
Cc netto / Tot att	0,0206	-0,0226	-0,0389	-0,0356	0,2593
Cr cl / Fatt	-0,0307	0,0250	-0,1322	0,0116	-0,1193
V. Agg. / Tot att	-0,0716	-0,0710	-0,0284	-0,0584	0,1864
Sconf br / Fido br	0,0523	0,9594	0,1095	0,0186	0,0654
Sconf ML / Fido acc ML	1,0000	0,2372	0,6411	0,0924	0,0613
util B / acc B		1,0000	0,2756	0,0436	-0,1826
util ML / acc ML			1,0000	0,0055	0,1757
util / acc				1,0000	0,2417
Q insoluti / Q sbf					1,0000

	$\frac{\text{Valore insoluti}}{\text{Valore sbf}}$
$\frac{\text{Cash Flow}}{\text{Debiti Totali}}$	-0,2167
$\frac{\text{Vendite}}{\text{Magazzino}}$	0,9562
$\frac{\text{Passività correnti}}{\text{Fatturato}}$	-0,3545
$\frac{\text{Patrimonio netto}}{\text{Totale attività}}$	0,3447
$\frac{\text{Oneri finanziari}}{\text{Debiti Totali}}$	0,6605
$\frac{\text{Cap circ netto}}{\text{Totale attivo}}$	0,3053
$\frac{\text{Crediti c\o clienti}}{\text{Fatturato}}$	-0,0836
$\frac{\text{V. Agg.}}{\text{Tot att}}$	0,2080
$\frac{\text{Sconf br}}{\text{Fido br}}$	-0,0479
$\frac{\text{Sconf ML}}{\text{Fido acc ML}}$	-0,0402
$\frac{\text{util B}}{\text{acc B}}$	-0,1967
$\frac{\text{util ML}}{\text{acc ML}}$	0,1802
$\frac{\text{util}}{\text{acc}}$	0,2115
$\frac{\text{Q insoluti}}{\text{Q sbf}}$	0,9535
$\frac{\text{V insoluti}}{\text{V sbf}}$	1,0000

L'analisi dimostra una intensa correlazione tra le variabili

1. $\frac{\text{Sconfinamento a breve termine}}{\text{Fido accordato a breve termine}}$
2. $\frac{\text{Fido utilizzato a breve termine}}{\text{Fido accordato a breve termine}}$.

Inoltre risulta che le variabili

1. $\frac{\text{Fatturato}}{\text{Valore del magazzino}}$,
2. $\frac{\text{Quantità insoluti}}{\text{Quantità presentazioni salvo buon fine}}$
3. $\frac{\text{Valore insoluti}}{\text{Valore presentazioni salvo buon fine}}$

sono fortemente correlate a due a due tra di loro.

Tuttavia le operazioni di pre-processing ci hanno portato a non considerare 4 su 5 delle variabili appena menzionate, per cui l'informazione ottenuta appare ridondante per la nostra analisi. La ripetizione dell'analisi dopo il pre-processing dei dati ci porta ai seguenti risultati:

	Cash Flow / Debiti Totali	Vendite / Magazzino	Passività correnti / Fatturato
Cash Flow / Debiti Totali	1,0000	0,0617	-0,1796
Vendite / Magazzino		1,0000	-0,0339
Passività correnti / Fatturato			1,0000

	Pat netto / Tot att	Oneri fin / Deb Tot	C.C. netto / Tot att	Cr. cl / Fatt	V. Agg. / Tot. att.
Cash Flow / Deb Tot	-0,1626	-0,0443	-0,1754	-0,0621	0,2311
Vendite / Magaz	-0,2350	-0,0062	-0,0154	0,0881	0,2663
Pass cor / Fatt	0,0716	-0,0939	-0,0351	0,5859	0,1720
Pat netto / Tot att	1,0000	0,0835	0,7752	-0,0602	-0,4900
Oneri fin / Deb Tot		1,0000	0,1178	-0,0254	-0,0540
C. C. netto / Tot att			1,0000	-0,1451	-0,6715
Cr clienti / Fatt				1,0000	0,1593
V. Agg. / Tot att					1,0000

	Sconf ML / Fido acc ML	util ML / acc ML	util / acc
Cash Flow / Deb Tot	0,0013	0,0901	-0,0714
Vendite / Magaz	0,1067	0,1158	0,0725
Pass. corr. / Fatt	0,1013	-0,0178	0,0618
Pat netto / Tot att	-0,0817	-0,0397	0,0913
Oneri fin / Deb Tot	0,0734	0,0044	0,0295
C. C. netto / Tot att	-0,0638	0,0389	0,0862
Cr. c\o cl / Fatt	0,1561	0,0459	0,0826
V. Agg. / Tot att	0,0724	-0,0033	0,0246
Sconf ML / Fido acc ML	1,0000	0,7924	0,2378
util ML / acc ML		1,0000	0,1158
util / acc			1,0000

Secondo l'analisi condotta le correlazioni più intense si riscontrano tra le coppie di attributi

- $\frac{\text{Patrimonio netto}}{\text{Totale attivo}}$ e $\frac{\text{Capitale circolante netto}}{\text{Totale attivo}}$[38];

[38] La correlazione tra questi due attributi è pari a 0,7752.

- $\frac{\text{Sconfinamento a m.l.termine}}{\text{Fido accordato a m.l. termine}}$ e $\frac{\text{Fido utilizzato a m. l. termine}}{\text{Fido accordato a m.l. termine}}$[39].

Tuttavia il valore della correlazione per queste due coppie di attributi non ci sembra abbastanza alto per giustificare l'eliminazione di alcune tra le variabili analizzate. Per questo decidiamo di proseguire i nostri esperimenti con tutte le variabili prese in considerazione dopo le precedenti operazioni di pre-processing.

4 Criteri per la costruzione di una Rete Neurale per la classificazione dell'insolvenza

Il lavoro di costruzione di una rete neurale è più un'arte che una scienza, come è stato sottolineato in diversi lavori, in quanto non esiste uno standard di riferimento in grado di funzionare ed offrire buone prestazioni in tutte le possibili applicazioni. L'architettura ottimale è da definire in base al problema da affrontare, e quindi la selezione dei parametri fondamentali deve avvenire in base al problema oggetto di studio.

I parametri fondamentali nel definire l'architettura della rete sono:

- il numero di neuroni di input;
- il numero di neuroni di output;

e, assumendo di utilizzare una rete feed-forward (o ricorrente) addestrata con Back-Propagation come nei nostri esperimenti,

- il numero di strati nascosti e dei neuroni appartenenti ad ogni strato.

Una delle proprietà delle reti neurali, che invero è stata alquanto sottovalutata, è la possibilità di sottoporre ad apprendimento anche l'architettura della rete: esistono dei metodi per determinare dinamicamente la migliore architettura. Alcuni di questi partono da una rete piccola ed aggiungono componenti fino a quando non si verificano ulteriori miglioramenti nelle prestazioni (l'esempio classico è il Cascade), altri suggeriscono di partire da una rete grande ed eliminare progressivamente delle componenti (Weight-Decay, Pruning) fino a che non si ottengono miglioramenti nelle prestazioni.

Tra gli ultimi ritrovati per definire una struttura ottimale ci sono gli algoritmi genetici [27], sviluppati originariamente da John Holland negli anni '70 allo scopo di capire i processi evolutivi dei sistemi naturali e di disegnare dei sistemi artificiali robusti e ben funzionanti [15]. Essi si basano sulle teorie evolutive di Darwin e di Lamark ed offrono metodi di soluzione simili a quelli impiegati dalla natura attraverso il processo evolutivo, attraverso la valutazione della bontà di una soluzione (fitness) e

[39] La correlazione tra questi due attributi è pari a 0,7924.

gli operatori genetici di mutazione, ricombinazione, selezione e sostituzione. Furono applicati da subito alla risoluzione di problemi, all'ottimizzazione di funzioni, al disegno di sistemi adattivi ed alla simulazione. Si può subito notare che i quattro ambiti di applicazione hanno tutti qualcosa a che vedere con le reti neurali, quindi la loro utilizzazione congiunta è stata una semplice e prevedibile conseguenza. E' da notare purtroppo che neanche quest'ultimo strumento evoluto è in grado di trovare la soluzione ottimale, per cui il maggior strumento in mano agli sviluppatori rimane l'esperienza.

4.1 Il numero dei neuroni di input

Per la scelta del numero di neuroni di input sono stati proposti degli strumenti per determinare la scelta migliore. Esistono però anche delle regole intuitive o empiriche, specie per la previsione di serie temporali, nelle quali i nodi di input rappresentano il numero di osservazioni a partire dalle quali si vuole effettuare la previsione. In questi problemi possibili regole sono utilizzare 12 unità per dati a rilevazione mensile, 7 per dati a rilevazione giornaliera e via discorrendo. Modellare una previsione soltanto su osservazioni passate dell'oggetto della previsione invece che su una serie di indici e rapporti è tipico quando ci si basa su ipotesi che prevedono una regolarità intrinseca nella serie, ad esempio una natura frattale, che spesso è stata ipotizzata in relazione ai mercati mobiliari [24]. Abbondare nel numero dei neuroni in ingresso spesso può portare a buoni risultati nella previsione multi-step, ma anche a deterioramento nelle prestazioni nella previsione single-step-ahead. Generalmente per gli altri scopi si usa il numero delle variabili osservate per ogni pattern, dopo una analisi preliminare volta a selezionare quelle più significative ed eventualmente ad operare una loro combinazione. Quanto detto finora può risultare più complesso se si introducono pattern di dimensione inferiore al numero dei neuroni di input, la cui definizione spetta all'utente che deve anche controllarne il funzionamento durante l'apprendimento, in quanto essi sono più difficili da gestire. Di norma la dimensione di questi pattern è pari ad un divisore del numero dei neuroni di input, ed essi sono usati ad esempio nella previsione di serie storiche di più variabili congiunte: abbiamo detto che in questi problemi i nodi di input stanno a rappresentare il numero di osservazioni a partire dalle quali si vuole effettuare la previsione, e se vogliamo effettuare la previsione su più variabili n in un orizzonte temporale di m mesi, il numero di neuroni di input sarà pari ad nm. Il caso opposto si ha quando introduciamo pattern di dimensione maggiore del totale dei neuroni di input. In questo caso un pattern dovrà essere diviso in più sub-pattern e questi ultimi dovranno essere processati uno alla volta. La dimensione del subpattern dovrà essere uguale al numero di neuroni di input, ma il pattern originario può avere una dimensione arbitraria e pattern appartenenti allo stesso insieme possono anche avere dimensioni diverse. La gestione

dei subpattern è ancora più complessa e necessita molta attenzione nella definizione e nell'esecuzione.

4.2 Il numero di strati nascosti e di neuroni per strato

È lo strato nascosto il garante della corretta funzionalità della rete, in quanto i neuroni nascosti sono gli strumenti in grado di catturare delle caratteristiche dei dati in ingresso e di trovare relazioni "nascoste" tra essi. Il numero di strati nascosti dipende dalla complessità del problema che andiamo a prendere in considerazione. Per i problemi finanziari è ormai accettato che un solo strato nascosto è sufficiente a garantire buoni risultati e per questo molti studi utilizzano un solo strato nascosto. Tuttavia un solo strato nascosto può portare ad un elevato numero di neuroni, con conseguente influenza negativa sul tempo di addestramento e sulla capacità di generalizzazione della rete. Hornik [17] ha dimostrato che un solo strato nascosto è in grado di approssimare qualsiasi funzione continua, ma ci sono alcuni studi (in disaccordo con altri) che sembrano indicare migliori performance di reti a due strati nascosti (in particolare per le previsioni finanziarie di dati ad alta frequenza), mentre aggiungere ulteriori strati è universalmente riconosciuto come uno spreco di risorse, in quanto non aumenta la capacità di generalizzazione e porta ad un aumento del tempo di addestramento . Per voler ricorrere ad una similitudine grafica, uno strato nascosto è in grado di separare superfici convesse nello spazio delle soluzioni, mentre l'aggiunta di un ulteriore strato riesce a separare superfici qualsiasi. A ben vedere, una rete neurale senza strati nascosti e con funzione di attivazione lineare può essere a ragione paragonata ad una regressione lineare (era questo il limite delle reti di perceptron) e le prestazioni offerte non sarebbero superiori a quest'ultimo strumento. Il numero di neuroni negli strati nascosti è particolarmente cruciale: un piccolo numero di neuroni può non rivelarsi in grado di cogliere correttamente le relazioni presenti tra variabili mentre un numero alto porta all'aumento del potere computazionale, a buoni risultati sui dati incontrati, ma anche alla memorizzazione di caratteristiche inutili e erronee e complessivamente dimostra scarsa capacità di generalizzazione. Sebbene esistano alcuni studi in cui si afferma che il numero dei nodi nascosti non influisce sulle prestazioni, esistono diversi suggerimenti sul numero di neuroni nascosti da utilizzare per ottimizzare il comportamento della rete. Diversi studi hanno indicato come miglior modello, parlando di un solo strato nascosto, quello con il numero di neuroni nascosti pari a quelli di input, ma esistono altre regole, ovviamente non valide per tutti i problemi, come le seguenti:[40]

- $n_{hidden} = 2n_{input} + 1$
- $n_{hidden} = n_{input}$

[40] $n_{training}$ indica il numero di esempi contenuti nel training set.

- $n_{hidden} = \frac{n_{input}}{2}$
- $n_{hidden} = \frac{n_{input}+n_{output}}{2} + \sqrt{n_{training}}$

Ne segue che il numero dei neuroni nascosti non può essere determinato a priori.

4.3 Il numero dei neuroni di output

È generalmente semplice determinare il numero di neuroni di output, in quanto è in stretta dipendenza con il problema oggetto di studio, ma sono comunque possibili diverse situazioni: nel caso generale in cui l'output sia composto da variabili, i neuroni di output dovranno essere generalmente pari al numero di variabili desiderate. Se il problema ha come scopo l'appartenenza ad una classe (ad es. classi di rating) si può utilizzare un neurone di output per ogni possibile classe, facendo in modo che l'output del neurone corrispondente alla classe desiderata assuma attivazione pari ad 1 e tutti gli altri assumano attivazione nulla. Si può in alternativa sfruttare la codifica binaria per utilizzare un numero di neuroni inferiore a quello definito in precedenza (tipicamente offre buone prestazioni solo nel caso in cui siano presenti due classi, come nel nostro lavoro). Nella previsione di serie temporali l'output rappresenta l'orizzonte temporale di previsione, ferma restando la distinzione tra metodo iterativo e metodo diretto fatta in precedenza.

4.4 Il coefficiente di apprendimento

Abbiamo già discusso in precedenza del coefficiente di apprendimento e di eventuali modifiche dell'algoritmo di apprendimento che portano modifiche alla regola di modifica dei pesi. Vogliamo solo ricordare che durante l'apprendimento il tasso di apprendimento dovrebbe essere continuamente sottoposto a modifica (generalmente se ne fa diminuire il valore, ma se si utilizzano strumenti di monitoraggio è possibile avere una migliore gestione che ne prevede anche l'aumento), per ottimizzare il processo di apprendimento. Tuttavia per alcuni autori [10] non è un parametro fondamentale e può essere lasciato al valore di default proposto.

4.5 La funzione di attivazione

Abbiamo già discusso in precedenza della funzione di attivazione. Anche se in teoria ogni funzione continua e differenziabile può determinare una funzione di attivazione, in pratica sono poche quelle ad essere utilizzate: si tratta di funzioni limitate, monotone crescenti e differenziabili[41]. E' da ricordare che in una rete possono esserci neuroni con funzione di attivazione differenti anche nello stesso strato, anche se la maggior parte dei ricercatori utilizza la stessa funzione di attivazione quanto

[41] Abbiamo presentato le principali funzioni di attivazione in precedenza.

meno tra nodi di output e nodi nascosti. Non esiste una regola unanime su quale funzione di attivazione utilizzare, anche se quella più usata è la logistica. Si può riscontrare una interessante preferenza da parte di alcuni importanti autori (ad es. Rumelhart) verso l'utilizzo della funzione lineare nei nodi di output, anche se questa non si dimostra in grado di cogliere i trend eventualmente presenti quando si effettuano previsioni su serie. Inoltre non sono stati sviluppati dei lavori comparativi tra funzioni lineari e non lineari per determinare quella preferibile. E' importante sottolineare che qualsiasi funzione venga utilizzata, è bene normalizzare l'output globale in un range di $(0, 1)$ o $(-1, 1)$. A tale proposito può essere utile fare affidamento ad una funzione di remapping degli output.

4.6 Remapping function

Queste funzioni servono a cambiare la funzione di determinazione del valore di output della rete e vengono di solito usate durante il funzionamento della rete per vedere come variano le performance al variare del tipo di output, in particolare quando i valori di output devono essere re-inseriti nella rete per produrre nuovi output (tipicamente previsione). Le funzioni prendono come argomento l'output determinato da un pattern e producono un nuovo valore di output lasciando inalterata la configurazione degli input. Le funzioni più utilizzate sono le seguenti:

- Funzione binaria: se l'argomento è maggiore di un valore soglia (di solito $0,5$) l'output assume valore 1, altrimenti assume valore nullo. E' usata generalmente per la classificazione binaria;
- Funzione inversa: è usata in problemi di classificazione binaria; i valori 0 vengono trasformati in 1 e vice-versa. Può anche essere usata con valori continui ed il risultato sarà di trasformare i valori maggiori di un valore soglia (di solito $0,5$) in 0 e quelli minori dello stesso valore soglia in 1;
- Funzione lineare: esegue una trasformazione lineare basandosi su parametri specificati dall'utente secondo la nota formula

$$y_r = a \cdot y_o + b$$

- Funzione *Clip*: vengono definite una soglia inferiore e una superiore, di solito $(0, 1)$ oppure $(-1, 1)$, ed i valori maggiori della soglia superiore vengono ricondotti al valore soglia superiore, mentre quelli minori della soglia inferiore vengono ricondotti al valore soglia inferiore. Gli altri rimangono invariati;
- Funzione a soglie: è la più flessibile di tutte. Vengono definiti due valori soglia in modo da determinare un intervallo. I valori all'interno dell'intervallo verranno tutti ricondotti ad un valore determinato dall'utente, quelli all'esterno verranno ricondotti ad un secondo valore, sempre determinato dall'utente.

4.7 Training set e test set

Per decidere in che modo suddividere i dati a disposizione in training set e test set bisogna prendere in considerazione diversi aspetti quali le caratteristiche del problema in esame, la natura dei dati disponibili ed il loro numero. Sono state proposte diverse proporzioni in modo da suddividere i dati a disposizione secondo delle percentuali stabilite nei seguenti modi:

- $[75\%, 25\%]$
- $[70\%, 30\%]$
- $[80\%, 20\%]$
- $[90\%, 10\%]$

Qualsiasi sia la proporzione scelta, è importante però che l'insieme sia rappresentativo della popolazione. In altre parole, se in un problema di classificazione binaria gli esempi che portano a risultato 1 sono il 55% del totale, tale proporzione dovrà essere rispettata sia nell'insieme di addestramento che in quello di verifica. A volte può essere introdotto un terzo insieme, il *validation set*: in questo modo il training set viene utilizzato per l'addestramento, il validation set viene utilizzato per evitare l'overfitting e per determinare il momento di stop-learning e il test set per verificare la validità del modello creato. Generalmente però, sopratutto per quanto riguarda insiemi di esempi non molto numerosi, si usa un solo insieme per determinare la fine dell'addestramento e per *testare* la rete.

Un'altra variabile critica è il numero dei dati da fornire alla rete. Anche qui non sono presenti regole universalmente accettate, anche se molti ricercatori sono concordi nell'affermare che le performance della rete migliorano all'aumentare dei dati forniti, ma spesso il numero dei dati da fornire alla rete è vincolato dal numero degli esempi a disposizione.

5 La costruzione della rete neurale per la classificazione dell'insolvenza

Siamo ora giunti al momento della costruzione della nostra rete neurale per la classificazione dell'insolvenza.

5.1 Gli esempi a disposizione

La prima operazione effettuata è stata l'eliminazione dei dati ritenuti errati. Nel capitolo precedente abbiamo evidenziato che alcune aziende presentano valori identici negli ultimi due anni delle rilevazioni. Decidiamo di non utilizzare questi dati, in

quanto, non disponendo di informazioni necessarie (la presa in visione del bilancio delle aziende in esame e la consulenza di chi ha raccolto i dati) pensiamo che questa anomalia sia derivata da errori di rilevazione.

Inoltre abbiamo effettuato i primi esperimenti a partire dai soli dati di bilancio (utilizzando quindi 8 attributi) e considerando come target soltanto lo stato di solvenza-insolvenza relativo all'ultimo anno.

5.1.1 Training set e Test set

Per i nostri esperimenti abbiamo diviso i dati a disposizione in *training set* e *test set*, utilizzando la ripartizione percentuale $[70\%, 30\%]$.

Degli esempi a disposizione 48 si rivelano solventi e 28 insolventi. Secondo questa ripartizione i due sotto-insiemi devono contenere il $63,16\%$ di esempi solventi e il $36,84\%$ di esempi insolventi.

In base a questo schema gli esempi del training set sono 53, di cui 33 solventi e 20 insolventi, e quelli del test set sono 23, di cui 15 solventi e 8 insolventi.

5.2 JavaNNS

Per i nostri esperimenti abbiamo utilizzato il tool *JavaNeuralNetSimulator*, un software sviluppato dall'Università di Tübingen[42](D) , che ci permette di simulare reti neurali e di operare con diversi algoritmi di apprendimento dandoci la possibilità di definire i diversi parametri degli esperimenti (tasso di apprendimento, numero di epoche etc.). Per fornire i pattern è stato necessario creare due file (uno per il training set ed uno per il test set) contenenti gli esempi da fornire alla rete seguendo una specifica sintassi:

SNNS pattern definition file V3.2
generated at Tue May 31 22:14:44 2005

No. of patterns : 23
No. of input units : 33
No. of output units : 1

 # Input pattern 1:
0.521234582795 23 0.518507094386181 0.516666666666667
0.307502290030991 0.289417999652023 0.220710434898748
0.064046112423296 4 0.0715881391909024 0.0820936335679637

[42] UNIVERSITY OF TÜBINGEN
WILHELM-SCHICKARD-INSTITUTE FOR COMPUTER SCIENCE
Department of Computer Architecture.

0.291163787680071 0.292423663652957 0.2905970034949
0.0291782446232768 0.0333127698951265 0.05
0.369461207369609 0.377654045076886 0.399456933535681
0.182690260790231 0.163924766547313 0.150703008454323
0.516383656837432 0.481914893617022 0.500127616130679
1 0.065355916551247 5 0.065355916551247 5
1 0.343819189619274 0.343819189619274
0.336326366944396 0.37183380049603 0.394922119213304
 # Output pattern 1:
0

in cui:

No. of patterns : 23

indica il numero dei pattern contenuti nel pattern set;

No. of input units : 33

indica il numero dei neuroni di input;

No. of output units : 1

indica il numero di neuroni di output.

I valori successivi indicano i diversi valori da fornire alla rete in input (quelli seguenti la stringa # *Input pattern* 1[43], anche se viene utilizzata solo per motivi di comprensibilità in quanto il carattere "#" è usato per indicare i commenti) e l'output desiderato (quelli seguenti la stringa # *Output pattern* 1). A riguardo dell'output desiderato, dobbiamo ricordare che disponiamo dei valori (0 e 1) relativi a 3 anni. Abbiamo deciso, per la nostra analisi, di utilizzare come output desiderato il valore oggetto della terza osservazione. In tal modo si spera che la rete riesca ad apprendere la dinamica delle osservazioni relative ai tre anni per fornire la previsione relativa all'ultimo anno. In relazione agli input dobbiamo dire che i valori sono stati organizzati, per ogni pattern, nel seguente modo: il pattern può essere raggruppato in sequenze di tre valori: questi gruppi di tre valori rappresentano le osservazioni relative ad un attributo; per ogni attributo le osservazioni sono state disposte in ordine cronologico.

Nell'esempio riportato sopra i valori

[43] Il valore 1 indica il numero del pattern, anche se non è necessario che l'ordine sia progressivo.

0.52123458279523 0.518507094386181 0.516666666666667

sono relativi all'attributo $\frac{\text{Cash Flow}}{\text{Debiti Totali}}$; la riga successiva

0.307502290030991 0.289417999652023 0.220710434898748

è relativa all'attributo $\frac{\text{Vendite}}{\text{Magazzino}}$ e così via.

5.3 La classificazione

Come sappiamo le reti neurali sono costituzionalmente *fuzzy*, per cui i risultati che offrono sono approssimati. Nel nostro caso ciò vuol dire che una rete addestrata bene, quando dovrà effettuare la previsione su un esempio che noi sappiamo essere in default, e quindi etichettato dal numero 1, non restituirà necessariamente il valore 1, ma un numero reale *abbastanza vicino* a esso. Lo stesso discorso vale per gli esempi in bonis, che sono etichettati con il valore 0 e per i quali la rete restituirà un numero reale *abbastanza vicino* a 0. Riteniamo necessario soffermarci un attimo sul significato dell'espressione "*abbastanza vicino*". Ricordiamo che lo scopo del lavoro è la classificazione, quindi addestrare la rete in modo che sia in grado di dirci se il pattern di dati che le forniamo in ingresso è riconducibile ad un cliente in bonis o in default. In un primo momento avevamo pensato di determinare due valori "soglia" compresi tra 0 ed 1: per ogni pattern presentato in ingresso, se la rete avesse restituito in uscita un valore minore od uguale alla soglia di modulo minore l'esempio sarebbe stato classificato in "bonis"; se la rete avesse restituito un valore maggiore o uguale alla soglia di modulo maggiore l'esempio sarebbe stato in "default"; altrimenti non saremmo stati in grado di assegnare l'esempio ad una di queste due classi. I valori soglia erano stati fissati a $0,3$ e $0,7$. Questa scelta si basava sulla consapevolezza che sarebbe stato necessario dividere lo spazio dei possibili risultati della rete in due segmenti complementari in modo da determinare l'appartenenza ad una classe, ma sarebbe stato difficile determinare a priori un valore soglia adeguato, in quanto non saremmo stati in grado di determinare un valore per cui i valori maggiori di esso avrebbero denotato situazioni di default[44]. La scelta più ovvia era di scegliere il valore $0,5$, in modo da tagliare lo spazio delle soluzioni in maniera simmetrica al fine di determinare l'appartenenza ad una delle due classi, ma inizialmente le informazioni a nostra disposizione non erano in grado di dirci se tale scelta avrebbe costituito una soluzione adeguata, in quanto non eravamo in grado di determinare a priori se i 2 segmenti dello spazio delle soluzioni sarebbero dovuti risultare simmetrici o meno (il che avrebbe comportato la scelta di un valore soglia diverso, ad esempio $0,4$ oppure $0,8$).

[44]E viceversa, per cui valori minori di esso avrebbero denotato situazioni di bonis.

Il proseguire degli esperimenti però ci ha portati a fare delle interessanti considerazioni:

- i casi in cui i la rete restituisce valori molto prossimi a $0,5$, cosa che potrebbe portare a dubbi sulla valutazione nello schema a soglia unica, sono in realtà abbastanza rari;
- gli errori di classificazione sono generalmente molto "evidenti": in caso di errata classificazione di clienti in default spesso la rete restituisce valori vicini allo 0, e viceversa in caso di errata classificazione di clienti in bonis la rete restituisce valori vicini a 1;
- confrontando gli errori di classificazione relativi allo schema a due soglie e quelli relativi allo schema ad una sola soglia, notiamo che l'adozione della soglia singola porta ad un aumento degli errori (che è relativo ai casi ritenuti "non-classificabili" nello schema a due soglie) abbastanza distribuito tra le due classi. L'incremento degli errori inoltre, almeno per le reti che forniscono buoni risultati nello schema a due soglie, appare abbastanza modesto.

Per questi motivi abbiamo successivamente deciso di utilizzare lo schema a soglia unica fissata a $0,5$: i pattern per i quali la rete restituirà valori maggiori o uguali di $0,5$ saranno approssimati ad una situazione di default; quelli per i quali la rete restituirà valori minori di $0,5$ saranno approssimati ad una situazione di bonis, in modo da eliminare la "zona grigia" e fornire una classe di appartenenza per ogni pattern in ingresso.

5.4 Esperimenti con 8 attributi

Abbiamo iniziato i nostri esperimenti utilizzando soltanto gli indicatori del modello di bilancio: essendo indicativi dello stato dell'impresa in un determinato momento è auspicabile che la rete, partendo da essi, sia in grado di ricavare informazioni utili ai fini della previsione del default.

Per gli esperimenti ci siamo serviti di due architetture differenti di rete in modo da poter confrontare i risultati: la prima è una architettura feed-forward completamente connessa; la seconda invece consiste in una rete "cablata" con due strati nascosti in cui i neuroni di input sono connessi a tre a tre con un neurone del primo strato nascosto, mentre il primo strato nascosto appare completamente connesso con il secondo, così come quest'ultimo con lo strato di output. In questo modo per ogni attributo, le osservazioni relative ai tre anni vengono sintetizzate dal neurone associato nel primo strato nascosto in una sorta di operazione di "pre-processing"[45].

[45] D'ora in poi ci riferiremo al primo modello di rete utilizzando il termine "standard", mentre utilizzeremo il termine "cablato" per riferirci al secondo modello.

5.4.1 Esperimenti con la rete cablata

I primi esperimenti sono stati effettuati sulla rete cablata. Essa presenta 24 neuroni di input, due strati nascosti di 8 neuroni ciascuno ed un neurone di output. Per ogni neurone la funzione di attivazione utilizzata è la logistica.

Primi esperimenti con Back-Propagation Nei primi esperimenti abbiamo confrontato l'algoritmo Back-Propagation ($\eta = 0,2$) con la sua variante Back-Propagation-Momentum ($\eta = 0,2$ e $\beta = 0,5$), ed abbiamo trovato conferma riguardo a ciò che abbiamo prima esposto in via teorica: nonostante l'aggiunta del termine di Momentum faccia aumentare il tempo di addestramento, le prestazioni della rete migliorano.

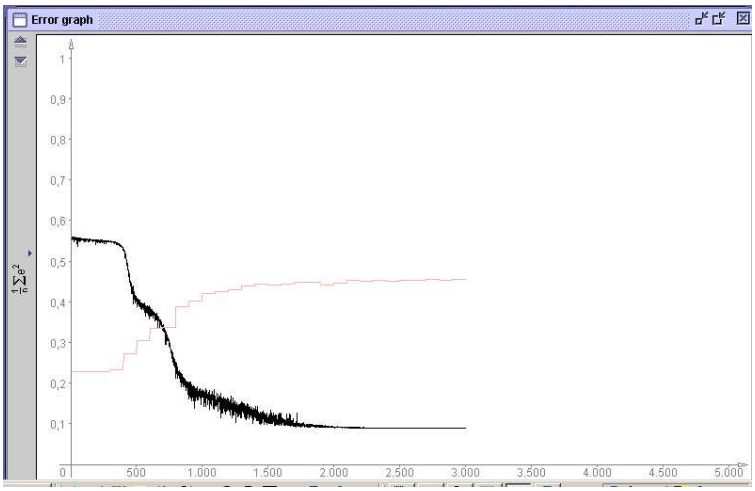

Figura 14. *Back-Propagation, inizializzazione* $[-0.3, 0.3]$

Abbiamo utilizzato nei nostri esperimenti l'inizializzazione casuale dei pesi tra due estremi forniti dall'utente ed abbiamo notato che l'inizializzazione influisce in maniera determinante sull'errore, e quindi sulle performance della rete. L'inizializzazione che sembra portare ai risultati migliori è quella in cui i valori sono compresi nell'intervallo $[-1, 1]$.

A tale proposito è utile confrontare i risultati ottenuti variando gli estremi dell'inizializzazione: la figura 16 ci mostra la dinamica dell'addestramento di una rete addestrata con Back-Propagation-Momentum (con i parametri sopra esposti) con i pesi inizializzati casualmente con valori compresi nell'intervallo $[-1, 1]$.

La figura 17 ci mostra invece la dinamica della stessa rete, inizializzata con valori compresi nell'intervallo $[-0.3, 0.3]$.Come si può osservare, in quest'ultimo caso le prestazioni sono peggiori del caso precedente a causa dell'inizializzazione a pesi più bassi.

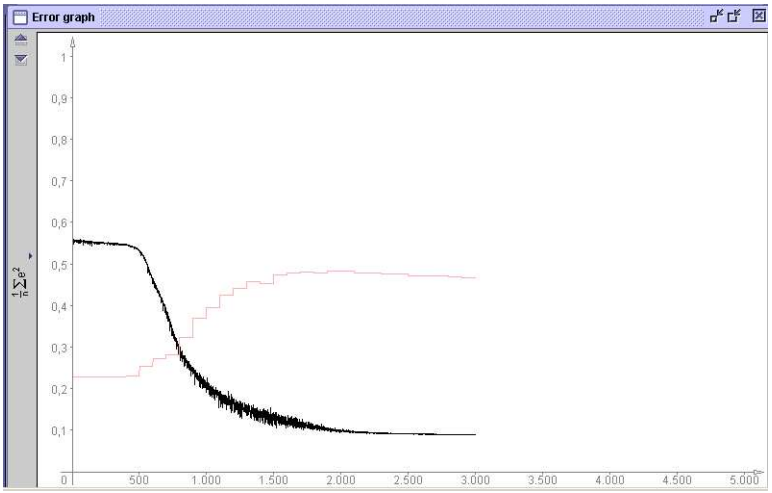

Figura 15. *Back-Propagation, inizializzazione* $[-1, 1]$

Già dai risultati ottenuti dai primi esperimenti siamo in grado di trarre alcune interessanti considerazioni. Innanzitutto emerge da un primo sguardo alle illustrazioni che l'addestramento porta ad una progressiva diminuzione dell'errore sul training set, mentre l'errore sul test-set appare assumere un andamento meno prevedibile, ma solitamente maggiore del precedente. A tal proposito valuteremo le prestazioni delle reti considerate in relazione alle classificazioni errate sul test-set.

La rete della figura 16 ha portato ad avere 5 errori sul test set (su un totale di 23 osservazioni, per cui il $21,7\%$), dei quali 4 sono relativi a errate classificazioni dei clienti in default (su un totale di 8 esempi di default, per cui il 50%), e 1 è relativo ad una errata classificazione di un cliente in bonis (su un totale di 15 esempi di bonis, per cui il $6,6\%$); sul training-set non si sono riscontrati errori[46]. Nelle tabelle seguenti "η" indica il coefficiente di apprendimento, "β" il termine di momentum, "misbo su tr" il rapporto percentuale tra classificazioni errate di clienti in bonis e clienti in bonis totali sul training-set, "misdef su tr" il rapporto percentuale tra classificazioni errate di clienti in default e clienti in default totali sul training-set, "errore su tr" il rapporto percentuale tra classificazioni errate e totale degli esempi a disposizione sul training-set; "misbo su ts" il rapporto percentuale tra classificazioni errate di clienti in bonis e clienti in bonis totali sul test-set, "misdef su ts" il rapporto in percentuale tra classificazioni errate di clienti in default e clienti in default totali sul test-set,

[46]D'ora in poi le errate classificazioni dei clienti in default verranno indicate con il termine "misdefault" mentre le errate classificazioni dei clienti in bonis verranno indicate con il termine "misbonis". In relazione a questi primi risultati dobbiamo subito dire che per le banche è più grave incorrere in un errore di misdefault che in uno di misbonis, per cui la rete non rappresenta una buona soluzione per la misurazione del rischio di credito.

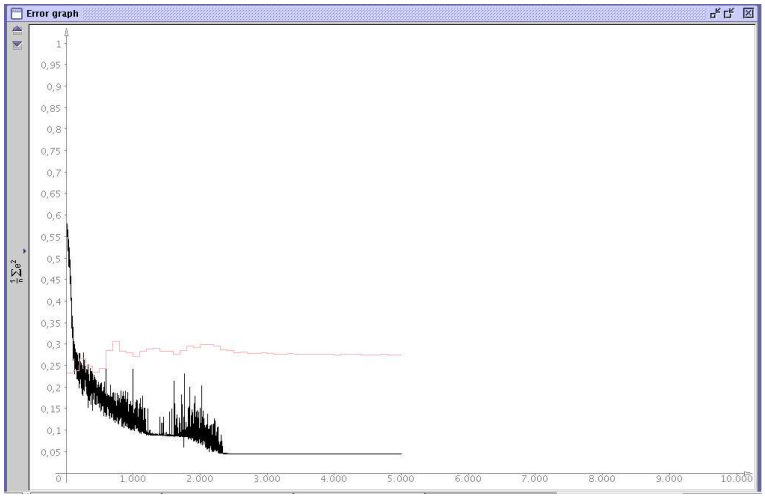

Figura 16. *Back-Propagation Momentum, inizializzazione* $[-1, 1]$ *(1)*

"errore su ts" il rapporto percentuale tra classificazioni errate e totale degli esempi a disposizione sul test-set; "id errori" indica i numeri associati ai clienti per i quali la rete produce errori di classificazione. La tabella successiva è relativa alla rete il cui addestramento è raffigurato nella figura 16.

η	β	misbo su tr	misdef su tr	errore sul tr	misbo su ts	misdef su ts	errore sul ts	id errori
$0,2$	$0,5$	0	0	0	$6,6$	50	$21,7$	54 83, 84, 103, 104

Tabella 1. *Back-Propagation Momentum, inizializzazione* $[-1, 1]$ *(1)*

Tuttavia abbiamo notato che, anche utilizzando sempre gli stessi estremi, diverse inizializzazioni portano a risultati anche abbastanza diversi tra di loro. Ciò dimostra una certa casualità nella fase di inizializzazione dei pesi, per cui conviene ripetere l'esperimento programmato con diverse inizializzazioni casuali, per poi concentrare l'attenzione su quella che ha fornito la performance migliore della rete. Abbiamo proceduto in questa maniera, anche considerando che gli strumenti a nostra disposizione ci permettono di addestrare la rete in un tempo generalmente inferiore al minuto.

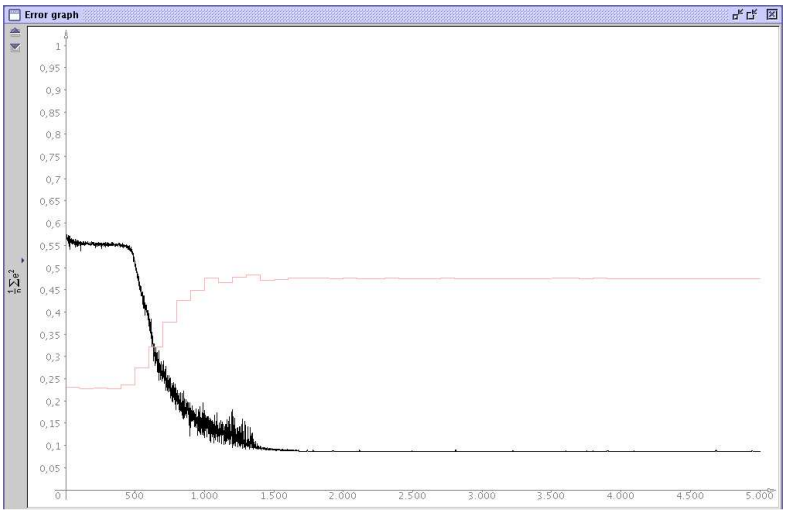

Figura 17. *Back-Propagation Momentum, inizializzazione* $[-0.3, 0.3]$ *(1)*

Esperimenti con differenti inizializzazioni Le figure seguenti intendono mostrare come cambia l'evoluzione dell'addestramento della rete quando si eseguono diverse inizializzazioni. I grafici e la tabella seguenti sono relativi a diverse inizializzazioni casuali con valori compresi nell'intervallo $[-0.3, 0.3]$ (reti addestrate con Back-Propagation-Momentum,$\eta = 0,2$ e $\beta = 0,5$).

Una delle reti prodotte ha fornito risultati simili a quelli descritti in precedenza nella tabella 1 riguardo ai primi esperimenti (5 errori, di cui 4 relativi a misdefault ed 1 relativo a misbonis; inoltre confrontando i pattern in cui la rete incorre a classificazioni errate 4 pattern sono classificati erroneamente anche dalla rete precedente); le altre hanno prodotto risultati non incoraggianti, in quanto il numero degli errori sul test-set varia da 9 a 11 (quindi dal $39,1\%$ al $47,8\%$) ed inoltre si riscontrano in quasi tutti i casi errori anche sul training-set.

La tabella ed i grafici seguenti invece sono relativi a diverse inizializzazioni casuali con valori compresi nell'intervallo $[-1, 1]$ (reti addestrate con Back-Propagation-Momentum, $\eta = 0,2$ e $\beta = 0,5$).

Anche in questi casi abbiamo una osservazione (27) in cui la rete ha fornito risultati simili a quelli forniti dalla migliore rete esaminata in precedenza (tabella 1): 5 errori sul test set (su un totale di 23 osservazioni, per cui il $21,7\%$), dei quali 4 sono relativi a errate classificazioni dei clienti in default (su un totale di 8 esempi di default, per cui il 50%), ed 1 è relativo ad una errata classificazione di un cliente in bonis (su un totale di 15 esempi di bonis, per cui il $6,6\%$); negli altri casi invece il numero

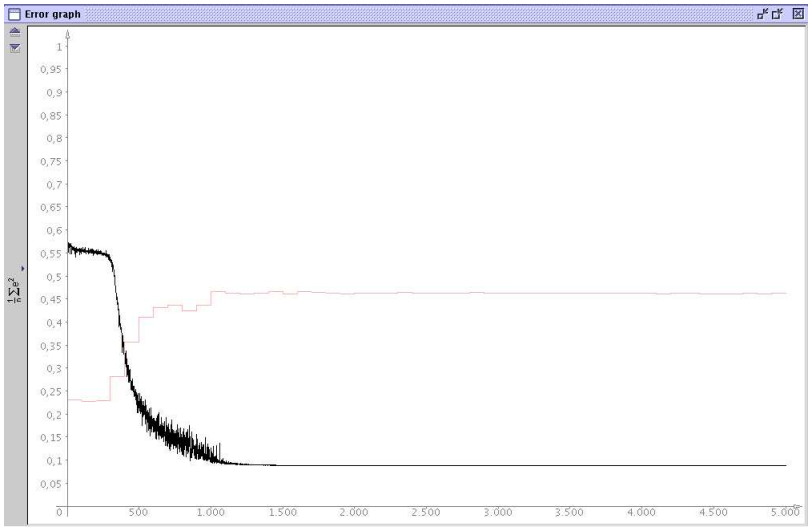

Figura 18. *Back-Propagation-Momentum, inizializzazione* $[-0.3, 0.3]$ *(2)*

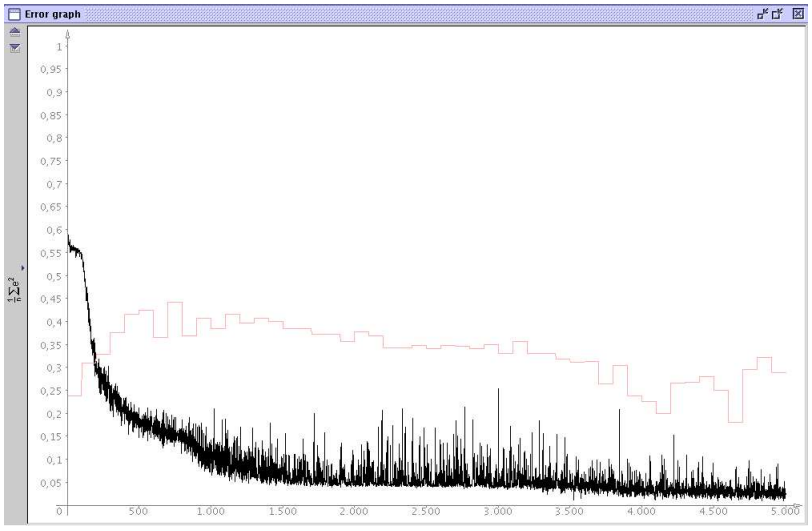

Figura 19. *Back-Propagation-Momentum, inizializzazione* $[-0.3, 0.3]$ *(3)*

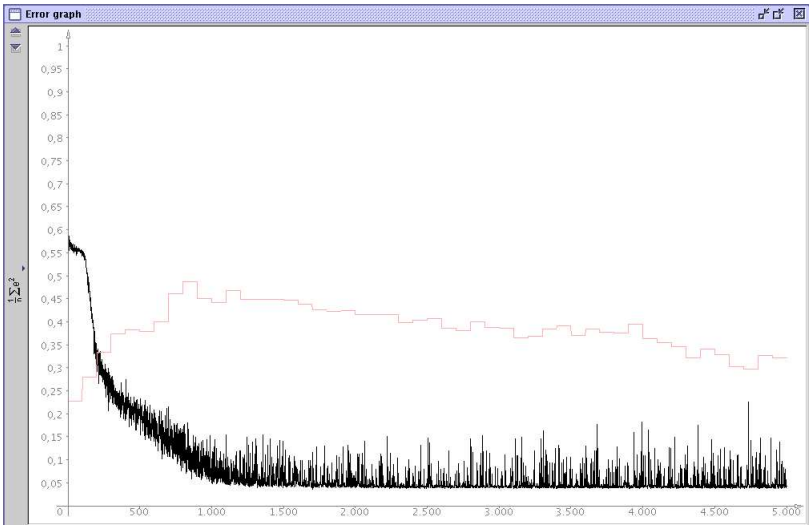

Figura 20. *Back-Propagation-Momentum, inizializzazione* $[-0.3, 0.3]$ *(4)*

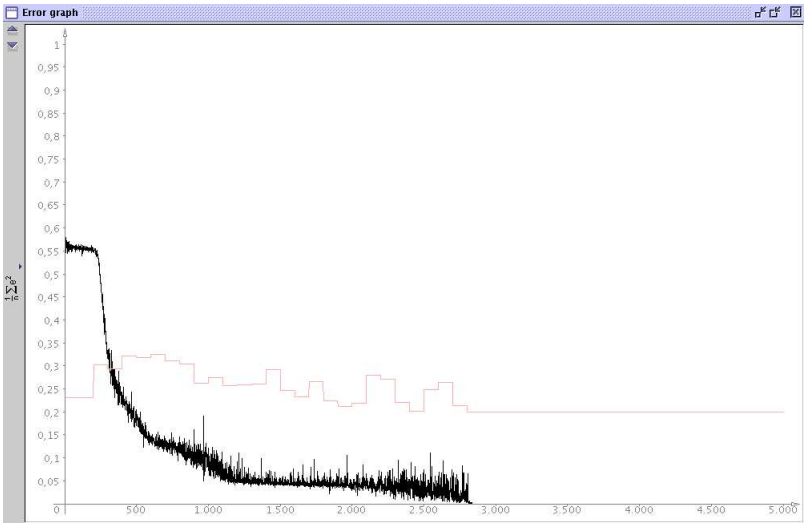

Figura 21. *Back-Propagation-Momentum, inizializzazione* $[-0.3, 0.3]$ *(5)*

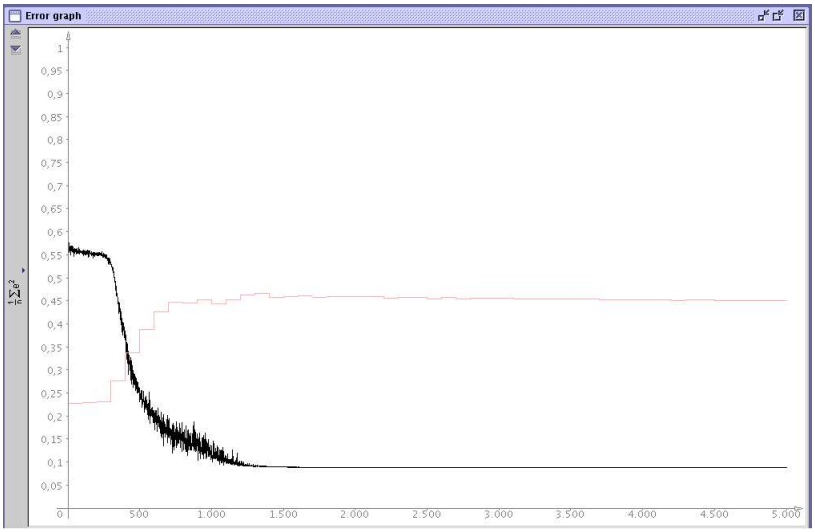

Figura 22. *Back-Propagation-Momentum, inizializzazione* $[-0.3, 0.3]$ *(6)*

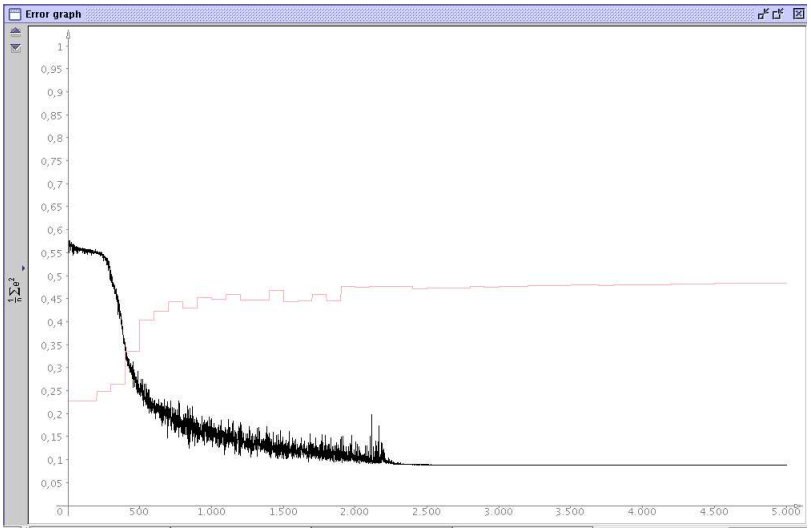

Figura 23. *Back-Propagation-Momentum, inizializzazione* $[-0.3, 0.3]$ *(7)*

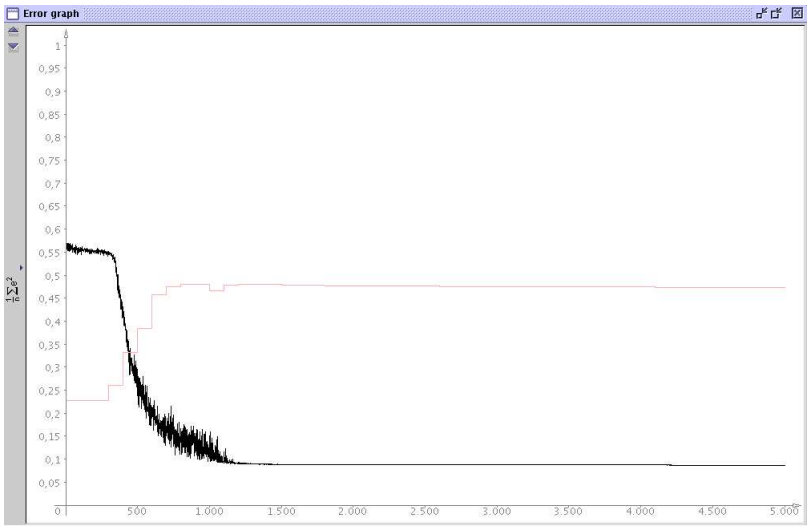

Figura 24. *Back-Propagation-Momentum, inizializzazione* $[-0.3, 0.3]$ *(8)*

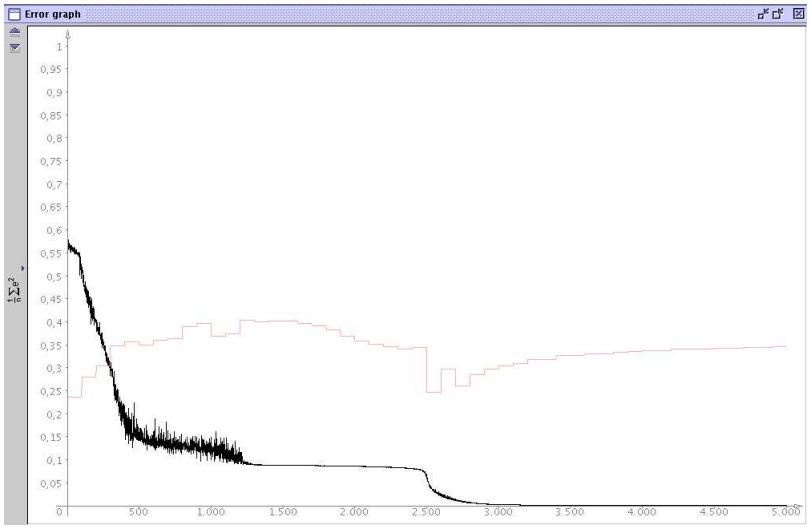

Figura 25. *Back-Propagation-Momentum, inizializzazione* $[-1, 1]$ *(2)*

ID	mb tr	md tr	errore tr	mb ts	md ts	errore ts	id errori
2	0	10	6	26, 6	87, 5	47, 8	2, 10, 31, 54 *91,106*, 83, 84, 100, 101, 102, 103, 104
3	0	10	6	13, 3	75	34, 7	10, 24 *75,106*, 83, 84, 100, 103, 104, 105
4	0	5	3	26, 6	62, 5	39, 1	2, 10, 24, 54 *106*, 83, 84, 100, 101, 105
5	0	0	0	6, 6	50	21, 7	54 83, 84, 101, 103
6	0	10	6	26, 6	87, 5	47, 8	2, 10, 31, 54 *91,106*, 83, 84, 100, 101, 102, 103, 104
7	0	10	6	26, 6	87, 5	47, 8	2, 10, 31, 54 *91,106*, 83, 84, 100, 101, 102, 103, 104
8	0	10	6	26, 6	75	43, 4	2, 10, 31, 54 *91,106*, 83, 84, 100, 102, 103, 104

Tabella 2. *Back-Propagation-Momentum, inizializzazione* $[-0.3, 0.3]$

ID	misbo su tr	misdef su tr	errore sul tr	misbo su ts	misdef su ts	errore sul ts	id errori
2	0	0	0	20	87, 5	43, 4	2, 10, 54 83, 84, 100, 101, 103, 104, 105
3	0	10	3, 7	20	87, 5	43, 4	2, 10, 54 *91,106*, 83, 84, 100, 101, 102, 103, 104
4	0	0	0	6, 6	50	21, 7	10 83, 84, 103, 105

Tabella 3. *Back-Propagation-Momentum, inizializzazione* $[-1, 1]$ *(2)*

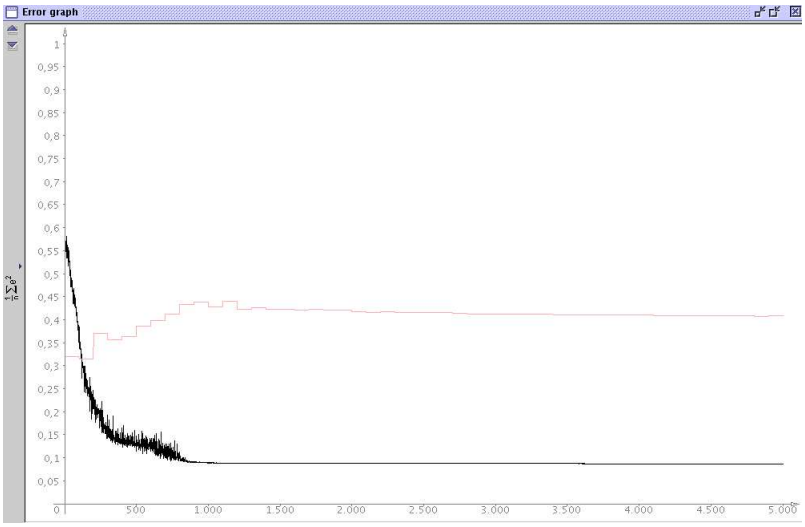

Figura 26. *Back-Propagation-Momentum, inizializzazione* $[-1, 1]$ *(3)*

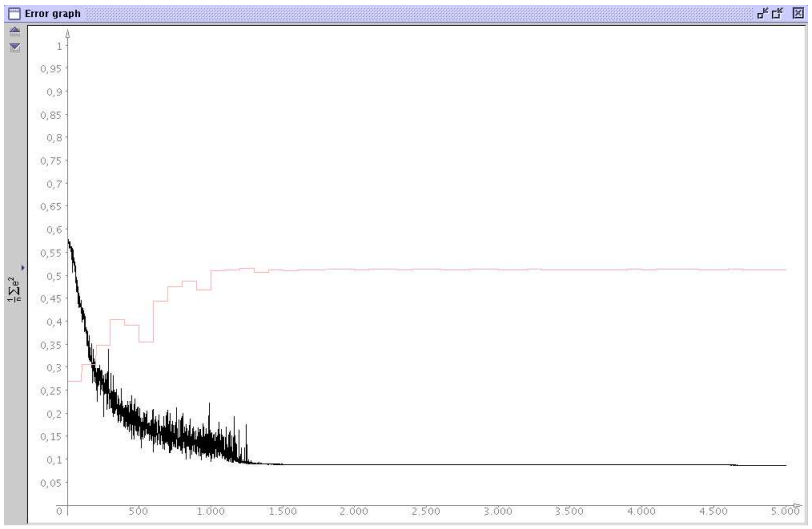

Figura 27. *Back-Propagation-Momentum, inizializzazione* $[-1, 1]$ *(4)*

degli errori sul test-set è 10. I risultati sopra esposti ci confermano che il parametro migliore per valutare le prestazioni di una rete è la percentuale di errate classificazioni sul test-set, in quanto affidarsi alla valutazione dell'errore (MSE) potrebbe portare a risultati inconsistenti. Si noti a tal proposito che utilizzando MSE come parametro di valutazione, la rete 21 è da considerarsi migliore di quella 27, mentre sono da considerarsi equivalenti in relazione agli errori di classificazione. Questi ultimi sono oltretutto i risultati ai quali sono interessate le banche.

A titolo di completezza riportiamo anche i grafici e le tabelle relativi alla dinamica dell'addestramento con inizializzazione con valori compresi nell'intervallo $[-0.5, 0.5]$.

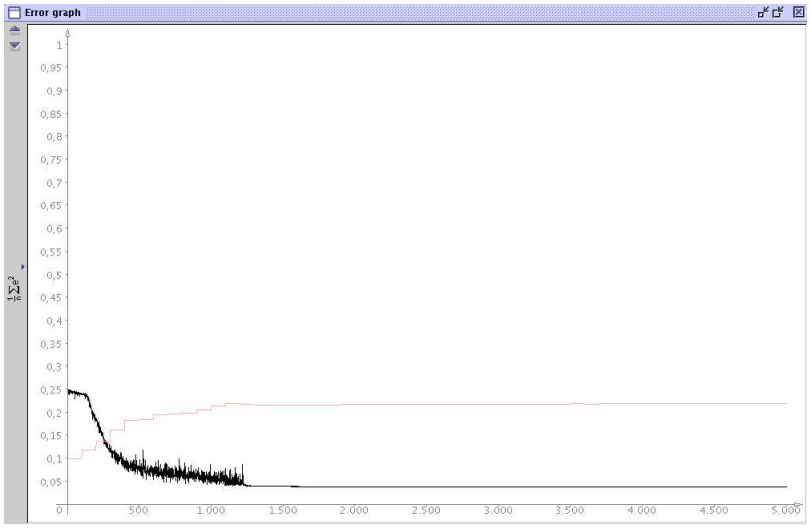

Figura 28. *Back-Propagation-Momentum, inizializzazione* $[-0.5, 0.5]$ *(1)*

Anche se il valore del MSE non sembra essere molto elevato, i risultati ottenuti in termini di errate classificazioni non sembrano particolarmente interessanti: le reti non riescono ad apprendere il concetto di default, e le errate classificazioni degli esempi in default sul test-set sono 8 (100% degli esempi in default) nei primi 2 casi e 7 (87,5%) nel terzo, mentre per quanto riguarda il bonis la rete sembra comportarsi meglio: 3 errori nel secondo caso (20%), 4 nel primo e terzo (26,6%).

Esperimenti condotti variando il numero di neuroni nel secondo strato nascosto Abbiamo poi provato a variare il numero di neuroni nel secondo strato nascosto, ma non abbiamo riscontrato miglioramenti significativi nella classificazione. Riportiamo i risultati conseguiti con una rete con 16 neuroni nel secondo strato nascosto. Abbiamo sempre utilizzato l'inizializzazione casuale compresa tra -1 ed 1, variando gli altri parametri dell'apprendimento.

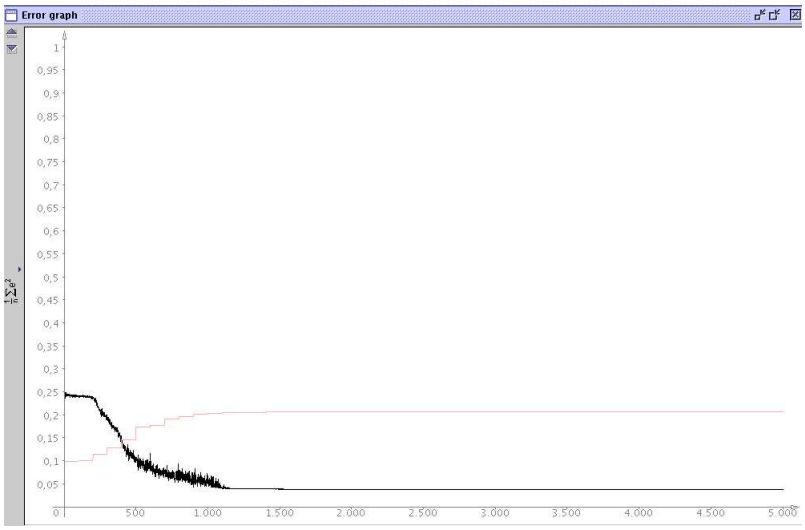

Figura 29. *Back-Propagation-Momentum, inizializzazione* $[-0.5, 0.5]$ *(2)*

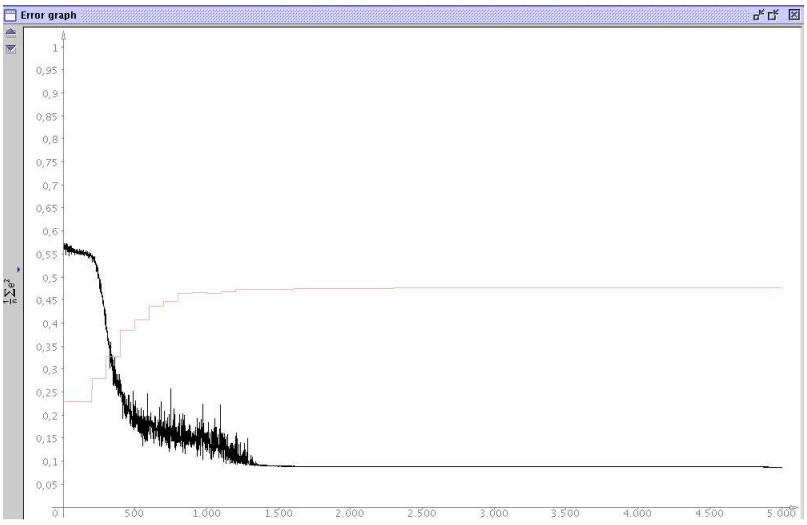

Figura 30. *Back-Propagation-Momentum, inizializzazione* $[-0.5, 0.5]$ *(3)*

ID	misbo su tr	misdef su tr	errore sul tr	misbo su ts	misdef su ts	errore sul ts	id errori
1	0	10	3,7	26,6	100	52,1	2, 10, 31, 54 *91,106*, 83 84, 100, 101, 102, 103, 104 105
2	0	10	3,7	20	100	47,8	2, 10, 54 *91,106*, 83 84, 100, 101, 102, 103, 104 105
3	0	10	3,7	26,6	87,5	47,8	2, 10, 31, 54 *91,106*, 83 84, 100, 101, 102, 103, 104

Tabella 4. *Back-Propagation-Momentum, inizializzazione* $[-0.5, 0.5]$

Le due figure seguenti sono relative a due reti addestrate con $\eta = 0,2$ e $\beta = 0,5$.

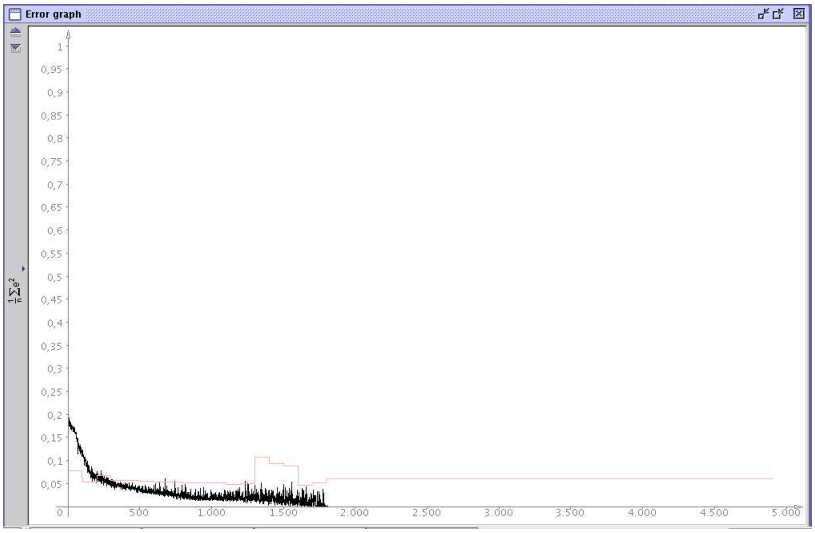

Figura 32. *Back-Propagation-Momentum, 16 neuroni nascosti nel secondo strato (2)*

Anche questi due esempi sono utili per chiarire la bontà del criterio di valutazione della performance utilizzato. Se si guarda la dinamica del MSE la prima rete appare peggiore della seconda, ma la prima ha riportato 9 errori sul test-set (39, 1%), di cui 7 misdefault (87, 5%) e 2 misbonis (13, 3%), mentre la seconda ne ha riportati 11 (47, 8%), di cui 7 misdefault (87, 5%) e 4 misbonis (26, 6%) (tabella 5).

ID	misbo su tr	misdef su tr	errore sul tr	misbo su ts	misdef su ts	errore sul ts	id errori
1	0	5	1, 8	13, 3	87, 5	39, 1	10, 54 *106*, 83, 84, 101, 102, 103, 104, 105
2	0	0	0	26, 6	87, 5	47, 8	8, 36, 41 83, 84, 100, 101, 102, 103, 105

Tabella 5. *Back-Propagation-Momentum, 16 neuroni nascosti nel secondo strato (1)*

Lo stesso discorso vale per le altre due reti: la rete "3" è stata addestrata con

$\eta = 0,5$ e $\beta = 0,5$, mentre la "4" con $\eta = 0.8$ e $\beta = 0,5\%$. La seconda ha riportato 4 errori($17,3\%$) di cui 1 misbonis ($6,6\%$) e 3 misdefault ($37,5\%$); la prima ne ha riportati 12 ($52,1\%$)di cui 8 misdefault (100%) e 4 misbonis ($26,6\%$) (tabella 6).

ID	misbo su tr	misdef su tr	errore sul tr	misbo su ts	misdef su ts	errore sul ts	id errori
3	0	10	3,7	26,6	100	52,1	2, 10, 31, 54 *91,106*, 83, 84, 100, 101, 102, 103, 104, 105
4	0	0	0	6,6	37,5	17,3	10 83, 84, 105

Tabella 6. *Back-Propagation-Momentum, 16 neuroni nascosti nel secondo strato (2)*

Esperimenti con errore non propagato Una interessante opportunità che ci viene offerta dallo strumento a nostra disposizione è di impostare l'errore che non verrà propagato indietro durante l'addestramento. Semplificando ed esemplificando in termini di errore assoluto, se impostiamo l'errore non propagabile (δ_{max}) a $0,2$, i casi in cui la rete restituisca, per pattern che sappiamo essere corrispondenti ad una situazione di default, valori maggiori (o uguali) a $0,8$ non determineranno modifiche dei pesi. Lo stesso vale per i pattern in bonis in cui la rete restituisca valori inferiori (o uguali) a $0,2$[47].

Nei primi esperimenti con l'errore non propagato abbiamo impostato i parametri $\eta = 0,8$, $\beta = 0,5$ e $\delta_{max} = 0,1$. I risultati sono riportati nella seguente tabella:

Come si può osservare le ultime due reti offrono degli ottimi risultati. A dire il vero l'inizializzazione con questi parametri è stata ripetuta altre 5 volte, con risultati simili ai primi due esperimenti riportati, per cui pensiamo che i buoni risultati ottenuti siano piuttosto casuali.

Abbiamo poi effettuato degli esperimenti provando a variare i parametri η e δ_{max}[48]. D'ora in poi riportemo soltanto i risultati migliori tra le diverse inizializzazioni effettuate.

[47] In quanto la differenza tra 1 (valore associato al default) ed un numero maggiore o uguale di $0,8$ è minore o uguale di $0,2$, così come la differenza tra un numero minore o uguale di $0,2$ e 0 (valore associato al bonis).

[48] $\beta = 0,5$

ID	misbo su tr	misdef su tr	errore sul tr	misbo su ts	misdef su ts	errore sul ts	id errori
1	0	0	0	13,3	37,5	21,7	18,24 83,84,105
2	6	0	3,7	6,6	37,5	17,3	48,56,10 101,104,105
3	0	0	0	13,3	75	34,7	10,54 83,84,100, 101,103,104
4	0	0	0	6,6	62,5	26	54 83,84,101, 103,104
5	0	5	1,8	6,6	25	13	54 85,83,104
6	0	0	0	6,6	25	13	54 84,104

Tabella 7. *Back-Propagation-Momentum, errore non propagato (1)*

In tutti questi casi i risultati non appaiono pienamente soddisfacenti, anche se fino a questo momento abbiamo comunque ottenuto alcune reti (5 e 6 della tabella 7) in grado di offrire buone prestazioni. Utilizzando questa architettura di rete possiamo dire che ricorrere alla mancata propagazione dell'errore non sembra avere effetti determinanti sulla classificazione e che le reti hanno in genere difficoltà ad apprendere il concetto di default.

5.4.2 Esperimenti con la rete standard

Gli esperimenti seguenti sono stati condotti utilizzando l'architettura feed-forward convenzionale. Per ogni neurone la funzione di attivazione utilizzata è la logistica.

Esperimenti con 8 neuroni nascosti In un primo momento abbiamo pensato che un solo strato nascosto fosse adeguato per gli scopi del nostro lavoro, ed abbiamo pensato di introdurre 8 neuroni nascosti. Su questa architettura abbiamo provato ad utilizzare diversi valori di η, β e δ_{max} per vedere se questi parametri hanno una certa influenza sulle prestazioni. I risultati sono esposti nelle tabelle successive:

ID	η	δ_{max}	mb tr	md tr	errore tr	mb ts	md ts	errore ts	id errori
7	0,8	0,2	0	0	0	0	50	17,3	83,84 103,105
8	0,2	0,1	0	20	7,5	20	62,5	34,7	2,10,24 85,86,99, 106,83,84, 102,103,105
9	0,2	0,2	0	0	0	13,3	50	26	31,10 83,84, 103,105
10	0,5	0,1	0	0	0	13,3	37,5	21,7	10,24 83,84,105
11	0,5	0,2	0	0	0	6,6	62,5	26	54 83,84,101, 103,104

Tabella 8. *Back-Propagation-Momentum, errore non propagato (2)*

ID	η	β	δ_{max}	misbo su tr	misdef su tr	errore sul tr	misbo su ts	misdef su ts	errore sul ts	id errori
1	0,2	0,5	0	0	0	0	13,3	37,5	21,7	10,54 83,84,103
2	0,2	0,2	0	0	0	0	13,3	37,5	21,7	10,54 83,84,103
3	0,2	0,8	0	0	0	0	0	62,5	21,7	83,84,103, 104,105
4	0,2	0,2	0,1	0	0	0	6,6	62,5	26	10 83,84,103, 104,105
5	0,2	0,5	0,1	0	0	0	6,6	37,5	17,3	10 83,84,103
6	0,2	0,8	0,1	0	0	0	6,6	37,5	17,3	10 83,84,103
7	0,2	0,2	0,2	0	0	0	6,6	50	21,7	10 83,84, 103,105
8	0,2	0,5	0,2	0	0	0	6,6	50	21,7	10 83,84, 103,105
9	0,2	0,8	0,2	0	0	0	6,6	50	21,7	54 83,84, 103,105

ID	η	β	δ_{max}	misbo su tr	misdef su tr	errore sul tr	misbo su ts	misdef su ts	errore sul ts	id errori
10	0,5	0,2	0	0	0	0	13,3	37,5	21,7	10,54 83,84,103
11	0,5	0,5	0	0	0	0	13,3	37,5	21,7	10,54 83,84,103
12	0,5	0,8	0	0	0	0	6,6	37,5	17,3	54 83,84,103
13	0,5	0,2	0,1	0	0	0	0	62,5	26	10 83,84,101, 103,105
14	0,5	0,5	0,1	0	0	0	0	50	21,7	10 83,84 103,105
15	0,5	0,8	0,1	0	0	0	0	50	17,3	83,84, 103,104
16	0,5	0,2	0,2	0	0	0	13,3	50	26	10,54 83,84 103,105
17	0,5	0,5	0,2	0	0	0	13,3	50	26	10,54 83,84, 103,105
18	0,5	0,8	0,2	0	0	0	6,6	50	21,7	54 83,84, 103,105
19	0,8	0,2	0	0	0	0	6,6	37,5	17,3	54 83,84,103
20	0,8	0,5	0	0	0	0	0	50	17,3	83,84, 101,103
21	0,8	0,8	0	0	0	0	6,6	62,5	26	54 83,84,103, 104,105
22	0,8	0,2	0,1	0	0	0	13,3	37,5	21,7	10,54 83,84,103,
23	0,8	0,5	0,1	0	0	0	6,6	37,5	17,3	54 83,84,103
24	0,8	0,8	0,1	0	0	0	6,6	62,5	26	54 83,84,103, 104,105
25	0,8	0,2	0,2	0	0	0	13,3	50	26	10,54 83,84, 103,105
26	0,8	0,5	0,2	0	0	0	0	37,5	13	83,84,103
27	0,8	0,8	0,2	0	0	0	0	75	26	83,84,101, 103,104,105

La penultima rete prodotta (numero 26) ha portato a risultati notevoli, ma dobbiamo constatare che su diverse inizializzazioni effettuate soltanto una volta i risultati sono stati di tale portata. Utilizzando invece altre combinazioni diverse inizializzazioni hanno portato a risultati più simili tra loro; ad esempio utilizzando $\eta = 0,8$, $\beta = 0,2$ e $\delta_{max} = 0$ (rete 19), anche se con risultati lievemente peggiori. Constatiamo comunque che sia nel caso dei parametri $\eta = 0,8$, $\beta = 0,5$ e $\delta_{max} = 0,2$ (rete 26) che nel caso dei parametri $\eta = 0,8$, $\beta = 0,2$ e $\delta_{max} = 0$ (rete 19) la percentuale di misdefault sul test-set è la stessa, per cui preferiamo utilizzare in seguito quest'ultima combinazione, che garantisce, eseguendo inizializzazioni diverse, prestazioni più

simili tra loro. Probabilmente è l'aggiunta dell'errore non propagato, unitamente allo *shuffling* che determina una certa variabilità nell'addestramento.

Esperimenti condotti variando il numero di neuroni dello strato nascosto Utilizzando la combinazione $\eta = 0,8$, $\beta = 0,2$ e $\delta_{max} = 0$ abbiamo poi provato a variare il numero di neuroni nello strato nascosto:

neuroni nascosti	misbo su tr	misdef su tr	errore sul tr	misbo su ts	misdef su ts	errore sul ts	id errori
9	0	0	0	13,3	37,5	21,7	10,54 83,84,103
10	0	0	0	13,3	50	26	10,54 83,84, 101,103
11	0	0	0	0	62,5	21,7	83,84,103, 104,105
12	0	0	0	6,6	50	21,7	54 83,84, 103,104

Abbiamo notato che l'aumento dei neuroni nello strato nascosto fa diminuire progressivamente il numero di epoche necessarie all'addestramento, ma non sembra comportare un miglioramento nelle prestazioni. Possiamo anche notare come gli errori in cui incorre la rete sono sempre relativi agli stessi clienti e che gli errori si verificano per la maggior parte nei casi di aziende in default. Questo potrebbe far sorgere dei dubbi circa l'attendibilità dei dati a nostra disposizione: un'azienda operante in maniera fraudolenta potrebbe non estinguere il debito, pur mostrando le condizioni adatte per farle, mentre potrebbe sembrare meno veritiero il caso in cui un azienda estingua il debito in assenza delle condizioni necessarie. Tuttavia, anche ammettendo questa possibilità, gli errori sembrano ancora troppi, per cui abbiamo proceduto ad ulteriori esperimenti per migliorare le nostre stime.

Esperimenti con 2 strati nascosti Abbiamo aggiunto uno strato nascosto in modo da aumentare le capacità computazionali della rete, ma neanche così siamo riusciti a migliorare le prestazioni. Abbiamo provato diverse combinazioni ma usato in maniera più attenta soltanto due modelli di rete neurali a due strati nascosti (tabella 10):

- un modello con due strati nascosti di 8 neuroni ciascuno;

neuroni nascosti	misbo su tr	misdef su tr	errore sul tr	misbo su ts	misdef su ts	errore sul ts	id errori
13	0	0	0	6,6	50	21,7	54 83,84, 103,104
14	0	0	0	6,6	50	21,7	10 84,101, 103,105
15	0	0	0	6,6	50	21,7	10 83,84, 103,105
16	0	0	0	0	62,5	21,7	83,84,101, 103,105
17	0	0	0	13,3	37,5	21,7	10,54 83,84,103
18	0	0	0	6,6	37,5	17,3	10,54 83,84,103
19	0	0	0	0	37,5	13	83,84,103
20	0	0	0	0	37,5	13	83,84,103
21	0	0	0	13,3	50	26	10,54 83,84, 103,105
22	0	0	0	6,6	62,5	26	10 83,84,101, 103,105
23	0	0	0	6,6	50	21,7	54 83,84, 103,105
24	0	0	0	6,6	50	21,7	10 83,84, 101,103

Tabella 9. *Rete standard, variazione del numero di neuroni nascosti*

neuroni nascosti	misbo su tr	misdef su tr	errore sul tr	misbo su ts	misdef su ts	errore sul ts	id errori
16	0	0	0	13,3	50	26	10,54 83,84, 103,105
48	0	0	0	6,6	37,5	17,3	54 83,84,105

Tabella 10. *Rete standard, 2 strati nascosti*

- un modello con due strati nascosti di 24 neuroni ciascuno.

Tra questi due modelli (tabella 10) il secondo offre prestazioni migliori. Tuttavia, alla fine degli esperimenti condotti utilizzando 8 attributi possiamo dire che i migliori risultati sono stati conseguiti utilizzando reti diverse tra loro, per cui le reti si sono dimostrate in grado di sviluppare delle capacità di generalizzazione a partire dai dati ricevuti e il numero di neuroni nascosti non è sembrato un parametro fondamentale, mentre più incisiva è stata la scelta dell'architettura: la rete cablata ci è sembrata migliore non solo per i risultati ottenuti, ma anche perchè questi sono stati ottenuti dopo un numero di inizializzazioni minori. Il limite non è rappresentato dalla rete, ma dal data-base: a partire da esso la rete non riesce a generalizzare correttamente, e non siamo in grado di dire se questo è dovuto all'esiguità, all'eventuale inconsistenza o alla mancanza di informazioni rilevanti. Troppo importante ci è parsa l'inizializzazione dei pesi: mantenendo gli stessi estremi, diverse inizializzazioni portano a risultati troppo differenti tra loro. Probabilmente questo è un segnale di mancanza di informazioni adeguate nei dati che stiamo utilizzando. Gli esperimenti successivi hanno cercato di ovviare a questa situazione.

L'opzione Cascade Una interessante opportunità che ci viene offerta dallo strumento a nostra disposizione è data dall'opzione *Cascade*. Questa ci permette di definire la nostra rete neurale senza preoccuparci della la topologia della rete. Il funzionamento di base può essere spiegato nel secondo modo: viene definita una rete neurale minimale formata da un neurone di input ed uno di output alla quale viene aggiunto un neurone nascosto alla volta fino a che l'errore non scende al di sotto di una soglia prefissata (o finchè non si verifica un'altra condizione di stop-learning). Con questo procedimento, al quale sono state apportate diverse varianti, l'apprendimento risulta generalmente più rapido e tale approccio è stato utilizzato in alcune pubblicazioni.

Abbiamo provato ad utilizzare questa opzione nei nostri esperimenti, ma i risultati a cui siamo pervenuti non sono stati soddisfacenti.

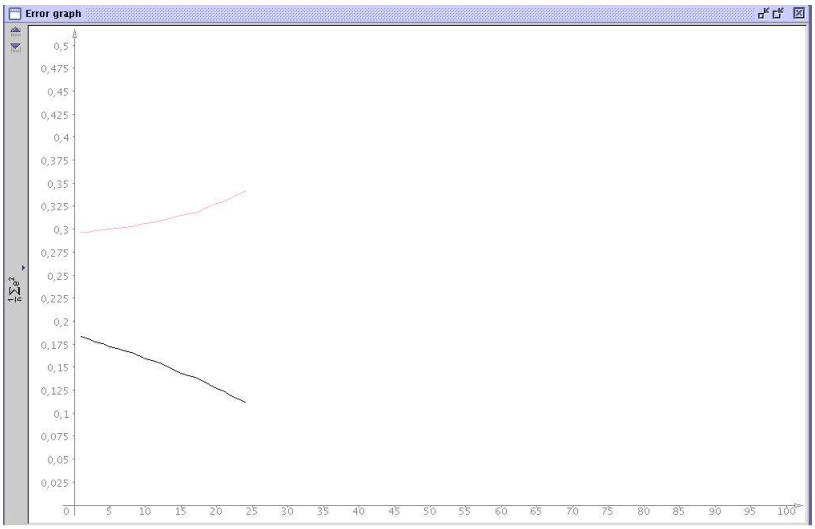

Figura 33. *Cascade*

Si può vedere infatti che la rete si adatta progressivamente ai dati del training-set, mentre l'errore sul test-set assume un andamento crescente all'aumentare delle epoche, comunque troppo superiore rispetto agli esperimenti condotti in precedenza. Per questo motivo abbiamo deciso di preoccuparci personalmente della definizione della topologia della rete, abbandonando il proposito di automatizzzare questa scelta.

5.5 Esperimenti con 11 attributi

Il passo successivo negli esperimenti è stato quello di utilizzare altri tre attributi derivati dal modello *Centrale dei Rischi* ed *Andamentale*, eliminando dalla analisi quelli che presentavano un numero di osservazioni corrette troppo scarso per evitare di falsare l'analisi. Gli attributi aggiunti sono stati:

- $\frac{\text{Sconfinamento a medio-lungo termine}}{\text{Fido accordato a medio-lungo termine}}$;
- $\frac{\text{Fido utilizzato a medio-lungo termine}}{\text{Fido accordato a medio-lungo termine}}$;
- $\frac{\text{Fido utilizzato}}{\text{Fido accordato}}$.

La scelta è stata determinata dal fatto che questi attributi riguardano direttamente l'evoluzione dei rapporti creditizi tra banca e cliente, per cui la rete può ricavare delle informazioni rilevanti per l'individuazione del default.

5.5.1 Esperimenti con la rete cablata

Per questi esperimenti abbiamo ritenuto adeguato non utilizzare l'errore non propagato (δ_{max}) e utilizzare solo gli altri due parametri ($\eta = 0,8$ e $\beta = 0,2$). Abbiamo utilizzato una rete cablata con 33 neuroni di input, 2 strati nascosti da 11 neuroni ciascuno ed un neurone di output. Per ogni neurone la funzione di attivazione utilizzata è la logistica.

L'utilizzo delle informazioni aggiuntive derivanti dall'utilizzo degli ulteriori tre attributi ha prodotto dei risultati sorprendenti, come riportato in tabella:

misbo su tr	misdef su tr	errore sul tr	misbo su ts	misdef su ts	errore sul ts	id errori
0	5	1,8	0	12,5	4,3	34,69

Tabella 11. *11 attributi: rete cablata*

Figura 34. *Esperimenti con 11 attributi*

Questa rete riporta un solo errore relativo ad un cliente in default sul test-set ed un solo errore, relativo anch'esso ad un cliente in default, sul training-set. Anche con questa architettura possiamo dire che il concetto di default è il più difficile da apprendere, ma i risultati che questa architettura ci fornisce sono pienamente soddisfacenti. Inoltre provando diverse inizializzazioni i risultati non cambiano: la rete

riporta errate classificazioni sempre sugli stessi clienti (contrassegnati dai numeri 34 e 69). Abbiamo inoltre, come ulteriore verifica della bontà dei risultati ottenuti, ridefinito da capo due nuovi insiemi per il training ed il test. Anche operando in questo modo gli errori di classificazione prodotti dalla rete sono stati uno sul training-set ed uno sul test-set; inoltre l'errore sul training-set è relativo allo stesso cliente per entrambe le suddivisioni del pattern-set determinate, mentre l'errore sul test-set è relativo a clienti diversi nei due casi.

Siamo riusciti ad ottenere in questo modo una rete neurale in grado di offrire delle buone capacità di generalizzazione con delle prestazioni degne di riguardo relativamente ai dati a disposizione utilizzando una architettura che si presenta come una semplice variante di nostra invenzione (non ci risulta che nelle pubblicazioni sia contemplata una rete con la medesima topologia) della feed-forward tradizionale, a dimostrazione della bontà delle reti neurali per i fini della classificazione e predizione.

Esperimenti condotti scambiando Training e Test set Abbiamo provato a scambiare training-set e test-set per vedere se le capacità di generalizzazione della rete cablata sono sfruttabili anche utilizzando un training-set ridotto in fase di addestramento con la prospettiva di utilizzo relativo ad una popolazione di dati molto ampia. I risultati sono esposti nella seguente tabella:

ID	η	β	mb su tr	md su tr	errore su tr	misbo su ts	misdef su ts	errore sul ts	id errori
1	0,8	0,2	0	0	0	12,1	25	16,9	38, 63, 64, 65 85, 92, 96, 98 99
2	0,5	0,2	0	0	0	12,1	40	22,6	38, 63, 64, 65 75, 85, 86, 89, 92, 96, 98, 99

Tabella 12. *11 attributi: scambio di training e test set*

La percentuale di misdefault appare inferiore agli esperimenti condotti in precedenza con 8 attributi, e per inciso l'utilizzo di un valore più elevato di η sembra

aver prodotto risultati migliori; tali risultati non sono buoni però se confrontati con gli esperimenti a insiemi non invertiti e mettono in evidenza il problema di come dividere l'insieme dei dati a disposizione tra training-set e test-set.

5.5.2 Esperimenti con la rete standard

Esperimenti con uno strato nascosto Abbiamo poi provato a operare su una rete standard con uno strato nascosto. Nella tabella sono riportati i risultati ottenuti al variare del numero di neuroni nel singolo strato nascosto:

neuroni nascosti	misbo su tr	misdef su tr	errore sul tr	misbo su ts	misdef su ts	errore sul ts	id errori
11	0	0	0	6,6	37,5	17,3	10 83, 84, 100
12	0	0	0	6,6	50	21,7	10 83, 84, 100, 103
13	0	0	0	6,6	50	21,7	10 83, 84, 100, 103
14	0	0	0	6,6	50	21,7	10 83, 84, 100, 103
15	0	0	0	6,6	50	21,7	10 83, 84, 100, 103
16	0	0	0	6,6	50	21,7	10 83, 84, 100, 103
17	0	0	0	6,6	50	21,7	10 83, 84, 100, 103
18	0	0	0	6,6	50	21,7	10 83, 84, 100, 103

I migliori risultati sono ottenuti dalla rete a 11 neuroni nascosti. Tuttavia un esame dei risultati in uscita ci induce a fare una considerazione: il valore restituito per il cliente "103", errato in tutti gli altri esperimenti, è 0.59005, quindi di poco superiore alla soglia che divide lo spazio delle soluzioni in bonis e default (0.5): questo potrebbe portare dei dubbi circa l'attendibilità del valore scelto. Attenzione

neuroni nascosti	misbo su tr	misdef su tr	errore sul tr	misbo su ts	misdef su ts	errore sul ts	id errori
19	0	0	0	6, 6	50	21, 7	10 83, 84, 100, 103
20	0	0	0	6, 6	50	21, 7	10 83, 84, 100, 103
21	0	0	0	6, 6	50	21, 7	10 83, 84, 100, 103
22	0	0	0	6, 6	50	21, 7	10 83, 84, 100, 103
23	0	0	0	6, 6	50	21, 7	10 83, 84, 100, 103
24	0	0	0	6, 6	50	21, 7	10 83, 84, 100, 103
33	0	0	0	6, 6	50	21, 7	10 83, 84, 100, 103

Tabella 13. *11 attributi: rete standard, variazione del numero dei neuroni nascosti*

però: si tratta sempre di un caso isolato, errato in tutte le altre prove, ma ci è utile per farci capire quanto si dimostra delicato il trattamento dei risultati. Abbiamo poi provato ad eseguire l'addestramento su tutti i dati a disposizione, solo per vedere se la rete è in grado semplicemente di apprendere a riconoscere i pattern che riceve in ingresso; anche in questo caso la rete non è riuscita a riconoscere il cliente "10".

Esperimenti con due strati nascosti Aggiungendo un ulteriore strato nascosto[49], riusciamo ad ottenere una classificazione corretta in più, con una precisione abbastanza accurata (per il cliente 103 il valore restituito in uscita dalla rete è 0.92792).

[49]Rete con 2 strati nascosti da 11 neuroni ciascuno.

neuroni nascosti	misbo su tr	misdef su tr	errore sul tr	misbo su ts	misdef su ts	errore sul ts	id errori
22	0	0	0	6,6	37,5	17,3	10 83, 84, 100

Tabella 14. *11 attributi: rete con 2 strati nascosti da 11 neuroni ciascuno*

Esperimenti condotti ridefinendo Training e Test set Abbiamo condotto ulteriori esperimenti ridefinendo il training-set ed il test-set (tabella 15). Questo ha portato ad un miglioramento nella classificazione: la rete compie un errore di misdefault sul training-set e due errori di misdefault sul test-set (la rete utilizzata ha un solo strato nascosto).

neuroni nascosti	misbo su tr	misdef su tr	errore sul tr	misbo su ts	misdef su ts	errore sul ts	id errori
11	0	5	1,8	0	25	8,6	75, 85, 99
19	0	5	1,8	0	25	8,6	75, 85, 99
20	0	5	1,8	0	25	8,6	75, 85, 99
21	0	5	1,8	0	25	8,6	75, 85, 99
22	0	5	1,8	0	25	8,6	75, 85, 99
23	0	5	1,8	0	25	8,6	75, 85, 99
24	0	5	1,8	0	25	8,6	75, 85, 99
33	0	5	1,8	0	25	8,6	75, 85, 99

Tabella 15. *11 attributi: ridefinizione di training e test set*

Abbiamo avuto un miglioramento nella classificazione: in tutti i casi la rete compie un errore di misdefault sul training-set e due errori di misdefault sul test-set.

Esperimenti condotti utilizzando Standard Back-Propagation Vista l'uniformità dei risultati ottenuti abbiamo pensato di modificare l'apprendimento eliminando il termine di momentum, utilizzando cioè l'algoritmo Back-Propagation nella

sua forma primitiva (utilizzando $\eta = 0.2$). L'idea era di provocare una modifica dei pesi derivante soltanto dall'operazione sul pattern corrente e non su quello precedente (la rete utilizzata ha un solo strato nascosto), ma questo non ci ha portato a nessun incremento nelle prestazioni (tabella 16).

neuroni nascosti	misbo su tr	misdef su tr	errore sul tr	misbo su ts	misdef su ts	errore sul ts	id errori
25	0	5	1,8	0	25	8,6	75, 85, 99
26	0	5	1,8	0	25	8,6	75, 85, 99
27	0	5	1,8	0	25	8,6	75, 85, 99
28	0	5	1,8	0	25	8,6	75, 85, 99
29	0	5	1,8	0	25	8,6	75, 85, 99
33	0	5	1,8	0	25	8,6	75, 85, 99

Tabella 16. *11 attributi: Standard Back-Propagation*

Esperimenti con errore non propagato Abbiamo allora pensato di aggiungere l'errore non propagato ($\delta_{max} = 0.1$) in modo da non far adattare troppo la rete ai dati del training set[50], in maniera differente da quanto detto in precedenza. Questa scelta ci ha portato ai seguenti risultati:

[50] $\eta = 0.2$

neuroni nascosti	misbo su tr	misdef su tr	errore sul tr	misbo su ts	misdef su ts	errore sul ts	id errori
25	0	0	0	13,3	0	8,6	3,10
26	0	0	0	13,3	0	8,6	3,10
27	0	0	0	13,3	0	8,6	3,10
28	0	0	0	13,3	0	8,6	3,10
29	0	0	0	13,3	0	8,6	3,10
33	0	0	0	13,3	0	8,6	3,10

Tabella 17. *11 attributi: errore non propagato*

Questi risultati si sono ripetuti per quasi tutte le inizializzazioni effettuate. Abbiamo ottenuto una rete in grado di offrire prestazioni superiori alla rete cablata, e gli errori in cui incorre sono misbonis[51]. Comunque anche in questo caso notiamo che per alcuni clienti ("17" e "36") il valore restituito è spesso vicino al valore soglia 0.5 e che, in tutti gli esperimenti condotti, il cliente "10" resta sempre difficile da classificare correttamente.

Sviluppi futuri

Ulteriori risultati in questo campo possono essere raggiunti utilizzando, per l'addestramento della rete, delle serie storiche più consistenti sia come numero di aziende che come numero di anni. L'introduzione della "Centrale dei Rischi" da parte della Banca d'Italia è una agevolazione in tal senso, anche perché ci fornisce dei dati che non saremmo stati in grado di trovare in altri modi.

Le reti che abbiamo considerato nei nostri esperimenti sono però vincolate dal numero di anni per i quali abbiamo dei dati a disposizione. Questo aspetto, se non risulta mostrare problemi utilizzando una base di rilevazione di tre anni, potrebbe avere dei limiti in caso di orizzonti temporali più lunghi: innanzitutto questo potrebbe portare, utilizzando l'architettura da noi definita, ad un aumento spropositato dei neuroni di input alla rete; in secondo luogo ciò farebbe emergere il problema di come trattare i dati relativi ad aziende per cui non si dispone dei dati per tutto l'orizzonte temporale (ad esempio aziende di costituzione recente). Una ipotetica soluzione potrebbe essere quella di utilizzare una rete che presenta un neurone di input per ogni attributo utilizzato alla quale, per ogni azienda, viene presentata la sequenza dei pattern per tutti gli anni a disposizione. Finiti gli anni un "segnale" (ad esempio l'attivazione di un neurone appositamente definito) dovrebbe far capire alla

[51] Ricordiamo che per le banche è preferibile incorrere in un misbonis piuttosto che in un misdefault.

rete che i dati successivi sono relativi ad un altra azienda e così via[52]. Probabilmente ciò porterebbe all'utilizzo della rete neurale nell'ambito di un contesto più ampio, unitamente ad altri strumenti.

Pensando però a eventuali sviluppi partendo dal lavoro svolto fino ad ora, ci sembra interessante l'idea di incorporare nell'analisi alcuni indicatori della congiuntura economica per contestualizzare i dati relativi alle aziende. In questo modo saremmo in grado di ricavare informazioni maggiori ai fini della stima del default o meno. Un particolare studio dovrebbe essere rivolto a determinare quali indicatori possono essere considerati indicativi della congiuntura economica (ad esempio tassi di interesse, inflazione, dati relativi alla Cassa Integrazione Guadagni etc.).

Altri dati da prendere in considerazione possono essere quelli relativi al settore economico di appartenenza. Gli attributi "RAE" e "SAE" esaminati in precedenza potrebbero fornire un interessante punto di avvio, ma riteniamo possa essere utile un partizionamento in modo da diminuire il numero dei valori assumibili dall'attributo così determinato. Riteniamo infatti che un possibile modo di incorporare questi valori nell'analisi sia inserire un neurone di ingresso per ogni possibile valore assumibile in modo che soltanto uno di questi si attivi per ogni azienda. Si può facilmente intuire che un tale approccio potrebbe funzionare in caso si utilizzi un attributo i cui valori assumibili sono quattro (settore primario, secondario, terziario, terziario avanzato), mentre risulterebbe meno proponibile utilizzando la codifica RAE.

Riteniamo utile, riferendoci specificamente all'aspetto tecnico delle reti neurali, uno studio relativo all'inizializzazione dei pesi: ci sembra che questo aspetto sia stato preso in considerazione meno di altri, quando invece ci è sembrato che abbia una notevole importanza nell'ambito degli esperimenti. Inoltre, relativamente al possibile trattamento dei dati mancanti, disponendo di un data-base più ampio sarebbe auspicabile un lavoro che confronti i diversi modi di procedere per scoprire quale delle diverse soluzioni determina il comportamento migliore.

Conclusioni

Le reti neurali sono strumenti di calcolo data-driven che si ispirano al funzionamento del cervello umano ed hanno trovato applicazione nella risoluzione di problemi di classificazione, predizione, ottimizzazione etc., fornendo risultati spesso superiori ai tradizionali metodi di risoluzione. Nel nostro lavoro abbiamo applicato questi strumenti di calcolo al problema della classificazione dell'insolvenza: la rete deve essere in grado di dirci se una azienda è in grado di restituire o meno un eventuale credito concessole da una banca. Per fare ciò la rete riceve in ingresso dei dati ricavati da un data-base contenente diversi indicatori relativi al bilancio, al modello Centrale dei Rischi ed al modello Andamentale di piccole e medie imprese italiane relativi

[52]Siamo dell'idea che uno studio sulle reti ricorrenti possa risultare utile per questo tipo di lavoro.

ad un triennio. Innanzitutto abbiamo proceduto ad operazioni di pre-processing dei dati, operando una analisi di correlazione tra i diversi indicatori presenti nel data-base per eliminare quelli più intensamente correlati, poi abbiamo proceduto al trattamento dei dati mancanti, all'eliminazione dei dati ritenuti errati e degli attributi ritenuti irrilevanti e alla normalizzazione degli attributi utilizzati affinchè i valori dei dati utilizzati dalla rete rientrino in un intervallo di variazione pre-determinato. Poi abbiamo proceduto alla fase sperimentale, utilizzando due modelli di rete neurale: il primo consiste in una architettura feed-forward tradizionale; il secondo consiste invece in una variante di una semplice feed-forward a due strati nascosti in cui i neuroni di di input sono connessi a tre a tre con un neurone del primo strato nascosto, mentre il primo strato nascosto è completamente connesso con il secondo ed il secondo strato nascosto è completamente connesso con lo strato di output.

Alla fine del nostro lavoro possiamo trarre delle interessanti conclusioni:

- La determinazione del rischio di credito deve avvenire attraverso una analisi congiunta dei dati di bilancio delle aziende e dei dati relativi ai loro rapporti con gli istituti di credito. I primi sono in grado di fornirci informazioni utili circa lo stato di salute delle aziende, ma sono anche i più suscettibili di errate interpretazioni e di possibili falsificazioni; i secondi contengono invece informazioni preziose per quanto riguarda la *storia* delle posizioni debitorie verso le banche e delle condizioni da esse offerte;

- Siamo riusciti a creare delle reti neurali in grado di fornire delle prestazioni spendibili per quanto riguarda la classificazione dell'insolvenza. Le architetture che hanno presentato risultati migliori sono la semplice feed-forward ad uno strato nascosto e la rete cablata da noi definita. In particolare non si è dimostrato rilevante, con questi due modelli, il numero di neuroni nello strato nascosto. Entrambe le architetture hanno portato a 2 errate classificazioni su 76, ma queste sono risultate diverse nella loro composizione. Con la prima architettura abbiamo ottenuto un errore sul test-set dell'$8,6\%$ (2 errori), e si tratta di errori relativi ad errate classificazioni di clienti in bonis, per cui il risultato può essere considerato buono. Ricordiamo però che questi risultati sono stati conseguiti ridefinendo il training-set ed il test-set rispetto a quelli utilizzati in precedenza per cui possiamo dire che la scelta di questi insiemi influenza pesantemente le prestazioni della rete. La seconda architettura ci ha portato ad un errore sul training-set ($1,8\%$) ed uno sul test set ($4,3\%$), ma questi sono dovuti a errate classificazioni di clienti in default, per cui tra i due modelli preferiamo il primo;

- Le potenzialità delle reti neurali come strumenti di calcolo sono note, ma devono essere accompagnate da una attenta analisi dei dati a disposizione: bisogna

capire con che tipo di dati si ha a che fare e cercare di incorporare tutte le informazioni ritenute utili, in particolare quando si ha a che fare con dati rilevati da terzi e che non sono stati raccolti al fine di essere utilizzati in tali esperimenti. Infatti la bontà dei risultati ottenuti è da ricondurre anche al fatto che è stata operata una attenta analisi dei dati a disposizione: le due operazioni di interpretare i dati mancanti derivati da presunti errori di calcolo (divisione di un numero per zero) e sostituire i valori mancanti in quanto non rilevati con la media (facendo affidamento sulla capacità della rete di operare su dati rumorosi) ci hanno aiutato a preservare delle informazioni che poi sono state utili per sviluppare la capacità di generalizzazione;

- Particolare attenzione nel costruire una rete neurale dev'essere riservata all'architettura della rete. Questa attività deve avvenire sempre tenendo in considerazione i dati che si hanno a disposizione: nel nostro caso buoni risultati sono stati forniti dalla rete cablata, che è stata progettata tenendo a mente la natura dei dati e la loro dinamica temporale. Riteniamo che tuttavia non sia particolarmente utile creare delle architetture troppo ad hoc influenzate dai dati a disposizione: i migliori risultati sono stati ottenuti utilizzando la semplice architettura feed-forward. In tal modo inoltre si rende proponibile un confronto con altri lavori. Alla luce di questa considerazione sembrano non particolarmente interessanti le opzioni messe a disposizione da diversi software che permettono una definizione "automatica" della topologia della rete (abbiamo parlato in precedenza dell'opzione *Cascade* che non offre buone prestazioni);

- quando si sceglie di lavorare con le reti neurali è buona norma ripetere gli esperimenti diverse volte. Abbiamo riscontrato che l'inizializzazione dei pesi ha una influenza considerevole sui risultati, quindi è utile fare diversi tentativi prima di decidere quali estremi utilizzare negli esperimenti e successivamente, una volta decisi questi, operare diverse inizializzazioni per ogni esperimento ed utilizzare i risultati migliori. Questo vale soltanto quando non si dispongono di informazioni adeguate, ed infatti negli esperimenti ad 11 attributi diverse inizializzazioni hanno portato ai medesimi risultati. Tuttavia è impossibile determinare a priori se le informazioni detenute siano o meno sufficenti per gli scopi che ci si prefigge, per cui bisogna procedere con diversi tentativi:

- Non è di fondamentale importanza la definizione del valore dei parametri utilizzati dall'algoritmo di apprendimento: abbiamo riscontrato che il variare di questi ultimi ha una influenza sul tempo di addestramento, ma la potenza di elaborazione dei computer attuali mette in secondo piano l'importanza di una eventuale ricerca relativa a determinarne la combinazione ottimale;

- Le reti neurali da noi definite rappresentano un buona soluzione per la classifi-

cazione dell'insolvenza. Questo risultato andrebbe però verificato con data-base più ampi;

- Alla luce di tutto quanto detto e realizzato fino ad ora, l'approccio classico incrementalista che caratterizza la ricerca sulle reti neurali (*Try-and-error*) resta l'unico in grado di ottenere dei risultati. Non esiste una soluzione universale per quanto riguarda l'architettura della rete, l'inizializzazione, i parametri dell'algoritmo di apprendimento (etc.); l'analisi dei dati e l'esperienza del ricercatore giocano il ruolo fondamentale.

Elenco delle figure

1 Il neurone biologico 7
2 Il neurone binario a soglia 10
3 Funzione di HEAVISIDE 11
4 Rappresentazione di un neurone artificiale 12
5 Rete completamente connessa 16
6 Rete stratificata 17
7 Rete feed-forward con uno strato nascosto 18
8 Un esempio di rete ricorrente: rete di Elman 19
9 Il percettrone 20
10 Problemi linearmente separabili 21
11 Delta rule 22
12 Il problema dello XOR 24
13 Evoluzione dell´apprendimento 28
14 Back-Propagation, inizializzazione $[-0.3, 0.3]$ 79
15 Back-Propagation, inizializzazione $[-1, 1]$ 80
16 Back-Propagation Momentum, inizializzazione $[-1, 1]$ (1) 81
17 Back-Propagation Momentum, inizializzazione $[-0.3, 0.3]$ (1) 82
18 Back-Propagation-Momentum, inizializzazione $[-0.3, 0.3]$ (2) 83
19 Back-Propagation-Momentum, inizializzazione $[-0.3, 0.3]$ (3) 83
20 Back-Propagation-Momentum, inizializzazione $[-0.3, 0.3]$ (4) 84
21 Back-Propagation-Momentum, inizializzazione $[-0.3, 0.3]$ (5) 84
22 Back-Propagation-Momentum, inizializzazione $[-0.3, 0.3]$ (6) 85
23 Back-Propagation-Momentum, inizializzazione $[-0.3, 0.3]$ (7) 85
24 Back-Propagation-Momentum, inizializzazione $[-0.3, 0.3]$ (8) 86
25 Back-Propagation-Momentum, inizializzazione $[-1, 1]$ (2) 86
26 Back-Propagation-Momentum, inizializzazione $[-1, 1]$ (3) 88
27 Back-Propagation-Momentum, inizializzazione $[-1, 1]$ (4) 89
28 Back-Propagation-Momentum, inizializzazione $[-0.5, 0.5]$ (1) 90
29 Back-Propagation-Momentum, inizializzazione $[-0.5, 0.5]$ (2) 91

30 Back-Propagation-Momentum, inizializzazione [−0.5, 0.5] (3) 91
32 Back-Propagation-Momentum, 16 neuroni nascosti nel secondo strato (2) 93
33 Cascade 101
34 Esperimenti con 11 attributi 102

Elenco delle tabelle

1 Back-Propagation Momentum, inizializzazione [−1, 1] (1) 81
2 Back-Propagation-Momentum, inizializzazione [−0.3, 0.3] 87
3 Back-Propagation-Momentum, inizializzazione [−1, 1] (2) 88
4 Back-Propagation-Momentum, inizializzazione [−0.5, 0.5] 92
5 Back-Propagation-Momentum, 16 neuroni nascosti nel secondo strato (1) 93
6 Back-Propagation-Momentum, 16 neuroni nascosti nel secondo strato (2) 94
7 Back-Propagation-Momentum, errore non propagato (1) 95
8 Back-Propagation-Momentum, errore non propagato (2) 96
9 Rete standard, variazione del numero di neuroni nascosti 99
10 Rete standard, 2 strati nascosti 100
11 11 attributi: rete cablata 102
12 11 attributi: scambio di training e test set 103
13 11 attributi: rete standard, variazione del numero dei neuroni nascosti 105
14 11 attributi: rete con 2 strati nascosti da 11 neuroni ciascuno 106
15 11 attributi: ridefinizione di training e test set 106
16 11 attributi: Standard Back-Propagation 107
17 11 attributi: errore non propagato 108

Riferimenti bibliografici

[1] D.H. Ackley, G.E. Hinton, and T.J. Sejnowski. A learning algorithm for boltzmann machines. *Connectionist models and their implications: readings from cognitive science*, pages 285–307, 1988.

[2] E. Altman. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *J. Finance*, 13, 1968.

[3] M. Anolli and P. Gualtieri. *La misurazione del rischio di credito nella gestione delle banche.* Il Mulino, 1999.

[4] A.F. Atiya. Bankruptcy prediction for credit risk using neural networks: A survey and new results. *IEEE Transactions on Neural Networks*, 12(4), 2001.

[5] V.S. Desai and R. Bharati. A comparison of linear regression and neural network methods for predictiong excess return on large stocks. *Annals of Operational Research*, 78, 1998.

[6] J.L. Elman. Finding structure in time. *Cognitive Science*, 14:179–211, 1990.

[7] D. Floreano and S. Nolfi. Reti neurali: Algoritmi di apprendimento, ambiente di apprendimento, architettura. *Giornale italiano di psicologia*, XX.

[8] G. Gabbi. L'utilizzo delle reti neurali per la misurazione del rischio di credito. In *La misurazione e la gestione del rischio di credito*. Bancaria, 1998.

[9] G. Gabbi. *La previsione nei mercati finanziari: trading system, modelli econometrici e reti neurali*. Bancaria, 1999.

[10] S.A. Hamid and Z.S. Iqbal. Using neural networks for forecasting volatility of s & p 500 index futures prices. *Journal of Business Research*, 57, 2004.

[11] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers, 2000.

[12] S.J. Hanson and L.Y. Pratt. A comparison of different biases for minimal network construction with back-propagation. *Advance in Neural Information Processing Systems*, 1, 1988.

[13] D. Hebb. *The organization of behaviour*. John Wiley & Sons, NY, 1949.

[14] S. Heravi, D.R. Osborn, and C.R. Birchenall. Linear versus neural network forecast for european industrial production series. *International Journal of Forecasting*, 20, 2004.

[15] J.H. Holland and J.S. Reitman. Cognitive systems based on adaptive algorithms. *SIGART Bull.*, (63):49, 1977.

[16] J. Hopfield. Neural networks and physical systems with emergent collective computational properties. 79:2554–2558, 1982.

[17] K. Hornik, K. Stinchombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2, 1989.

[18] M.I. Jordan. Attractor dynamics and parallelism in a connectionist sequential machine. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society, Hillsdale, NJ*, pages 531–546. Erlbaum, 1986.

[19] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science, 13 May 1983*, 220(4598):671–680, 1983.

[20] T. Kohonen. *Self-organization and associative memory: 3rd edition.* Springer-Verlag New York, Inc., New York, NY, USA, 1989.

[21] T. Kohonen. Speech recognition based on topology-preserving neural maps. pages 26–40, 1989.

[22] B. Kosko and S. Isaka. Fuzzy logic. *Scientific American*, 1993.

[23] M. Lam. Neural network techniques for financial performance prediction: integrating fundamental and technical analysis. *Decision Support System*, 37, 2004.

[24] B.B. Mandelbrot. A multifractal walk down wall street. *Scientific American, February, 1999.*

[25] S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bullettin of Mathematical Biophysics*, 7, 1943.

[26] M. Minsky and S. Papert. *Perceptrons.* M.I.T. Press, Cambridge, Mass., 1969.

[27] M. Mitchell. *An introduction to genetic algorithms.* MIT press, Cambridge, MA, 1998.

[28] L. Molteni, E. Coffetti, and G. De Laurentis. Uno scoring evoluto per le banche italiane. In *La misurazione e la gestione del rischio di credito.* Bancaria, 1998.

[29] Basel Committee on Banking Supervision. *International Convergence of Capital Measurement and Capital Standards, a revised framework.* Bank for International Settlements, CH-4002 Basel, Switzerland, june 2004.

[30] S.L. Pang, Y.M. Wang, and Y.H. Bai. Credit scoring model based on neural network. In *Proc. of the First International Conference on Machine Learning and Cybernetics*, Beijing, 4-5 Novembre 2002.

[31] S. Piramuthu. Financial credit-risk evaluation with neural and neurofuzzy systems. *European Journal of Operational Research*, 112, 1999.

[32] Li Rong-Zhou, Pang Su-Lin, and Xu Jian-Min. Neural network credit-risk evaluation model based on back-propagation algorithm. In *Proceedings of the First International Conference on Machine Learning and Cybernetics*, Beijing, 4-5 Novembre 2002.

[33] E. Rosenblatt. *Principles of Neurodynamics.* Spartan, New York, 1962.

[34] G. Rotundo. Neural networks for large financial crashes forecast. *Physica*, 344, 2004.

[35] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning internal representations by error propagation. pages 318–362, 1986.

[36] T.J. Sejnowski and G.E. Hinton. Separating figure from ground with a boltzmann machine. pages 703–724, 1987.

[37] D. Specht. A general regression neural network. *Neural Network, vol. 2, n. 6*, pages 568–576, 1991.

[38] R. Sun. Connectionist models of commonsense reasoning incorporating rules and similarities. *Knowledge Acquisition*, 4, 1992.

[39] K. Tam and M. Kiang. Managerial applications of the neural networks: The case of bank failure predictions. *Management Science*, 38, 1992.

[40] P. Werbos. Backpropagation, past and future. In *Proceedings of the IEEE International Conference on Neural Networks*. IEEE Press, 1988.

[41] B. Widrow and M. E. Hoff. Adaptive switching circuits. pages 96–104. IRE WESCON Conv. Record, 1960.

[42] J. Wooldridge. *Introductory Econometrics: A Modern Approach (second edition)*. South-Western College Pub., 2002.

[43] C. Wu and X.M. Wang. A neural network approach for analyzing small business lending decisions. *Review of Quantitative Finance and Accounting*, 15, 2000.

[44] G. Zhang, B.E. Patuwo, and M.Y. Hu. Forecasting with artificial neural networks: the state of the art. *International Journal of Forecasting*, 14, 1998.