

Identificazione dei Parametri Caratteristici di un Plasma Circolare Tramite Rete Neuronale

Il presente lavoro, facente seguito a quanto scritto precedentemente ha il fine di descrivere il codice di calcolo attuale, le metodologie utilizzate ed i risultati ultimi raggiunti.

Descrizione del codice

La schermata iniziale (vedi Allegato 1) si presenta suddivisa in 7 riquadri: 3 di essi sono di utilità mentre negli altri 4 sono riportati tutti i dati necessari alla identificazione del caso che si sta trattando e dei relativi parametri utilizzati.

Il primo riquadro è relativo alla definizione della banca dati, i cui valori saranno utilizzati nelle successive elaborazioni, alla definizione dell'architettura adottata per il sistema di rete neuronale ed alla generazione dei pesi iniziali che utilizza l'architettura precedentemente fissata.

Il secondo riquadro è relativo alla fase di training ed è stato suddiviso in 3 step. Per la descrizione del significato degli step e dei parametri associati si rimanda al punto 2) relativo alle Metodologie.

Il terzo ed il quarto riquadro sono relativi rispettivamente alla fase di testing della rete ed all'analisi dei risultati ottenuti nella fase di testing.

Metodologie

a) scelta di una nuova Banca Dati: scegliendo la casella "Altra Banca Dati" il sistema richiede il nome del file relativo, il numero di patterns di cui la banca dati è composta, il numero di dati che costituiscono il singolo input ed il singolo output per la rete ed, infine, le etichette che il sistema attribuirà ai singoli outputs. Finito l'inserimento dei dati è necessario passare alla normalizzazione scegliendo la casella "Normalizzazione Banca Dati". Il sistema inizia la procedura creando un altro file, con estensione ".NOR" che costituirà il file di input per tutte le elaborazioni precedenti. La normalizzazione consiste nel prendere singolarmente ogni vettore di output (di dimensione evidentemente pari al numero di patterns totali), trovarne i valori minimo e massimo, farne la somma e la differenza ed effettuare una trasformazione del tipo

$$\text{Val. normaliz.} = \frac{2 \cdot \text{Valore} - \text{somma}}{\text{differenza}}$$

La stessa cosa viene fatta per la matrice composta dai dati di input. Il risultato è che ogni valore di output e tutti i valori di input sono compresi tra -1 ed 1. I fattori di conversione sono memorizzati in un altro file con estensione ".FAT" ed è possibile vederli scegliendo la casella "?".

b) Scelta dell'architettura della rete: l'unico parametro che manca per definire interamente la rete è il numero di neuroni Hidden. Scegliendo la casella "HID" è possibile fissare questo parametro. Si ricorda che il numero di Layers è fisso e pari a 3.

c) Generazione dei pesi iniziali: conoscendo l'architettura della rete sono note il numero e la disposizione delle connessioni; scegliendo la casella "RUN" si attiva la procedura di determinazione dei pesi iniziali. Il parametro β definisce il tipo di funzione sigmoidale scelta: come detto precedentemente la funzione di attivazione è del tipo:

$$g(x) = \frac{2}{1 + e^{-2\beta x}} - 1$$

che risulta compresa tra -1 ed 1 ed è derivata dalla funzione Glauber compresa, invece, tra 0 ed 1.

La derivata prima di $g(x)$ è:

$$g'(x) = \beta \cdot [1 - g(x)^2]$$

(Per $\beta=1$ si ha $g(x)=\tanh(x)$, per $\beta=0.5$ si ha $g(x)=\tanh(x/2)$). Il parametro β sarà poi utilizzato sia nelle fasi di Training che di Testing della rete.

Per il livello 0 (connessioni layers input-hidden), la procedura consiste nel trovare i pesi in modo che, per tutti gli N patterns considerati, gli ingressi a tutti i neuroni Hidden siano compresi in un range che non porti in saturazione la funzione di attivazione sigmoidale: il parametro R_{sig} è un valore vicino al ginocchio della curva;

evidentemente variando β il valore di R_{sig} va cambiato in modo opportuno.

Per il livello1 (connessioni layers hidden-output), la procedura consiste nel trovare i pesi in modo che, per tutti gli N patterns considerati, le uscite di tutti i neuroni di output siano comprese tra -1 ed 1: il parametro R_{lin} determina, con un margine di conservatività, tale condizione. La funzione di attivazione in uscita è lineare del tipo: $g(x) = x$.

La procedura genera inizialmente un set di pesi casuali, in seguito verifica che tali pesi non generino con tutti gli N patterns delle situazioni anomale di ingresso agli Hidden e di uscita dei neuroni di output (valori tutti o in massima parte negativi, nulli o positivi). Se ciò dovesse accadere rigenera casualmente i pesi che hanno prodotto quelle anomalie fino ad ottenere condizioni accettabili. Il passo successivo è quello di variare i pesi, tramite un coefficiente di amplificazione, in modo che vengano soddisfatte le condizioni fissate dai parametri R_{sig} ed R_{lin} .

d) Training della rete: questa fase è stata suddivisa in tre step per differenti finalità:

Step1: è possibile modificare il Learning Rate LR, il termine di momento α e due fattori chiamati β_1 e β_2 . Tali fattori comportano un apprendimento basato sulla variazione del LR iniziale e variabile per ogni connessione e per ogni pattern con un modello del tipo:

$$LR_{ij(p)}^{(t+1)} = LR_{ij(p)}^{(t)} * \beta_2 \text{ con } \Delta W_{ij}^{(t+1)} / \Delta W_{ij}^{(t)} < 0$$

ed

$$LR_{ij(p)}^{(t+1)} = LR_{ij(p)}^{(t)} + \beta_1 \text{ con } \Delta W_{ij}^{(t+1)} / \Delta W_{ij}^{(t)} > 0$$

se si pone $\beta_1 = 0$ e $\beta_2 = 1$ gli $LR_{ij(p)}$ rimangono costanti ed uguali al LR iniziale.

Il termine di momento comporta un apprendimento basato su un modello del tipo:

$$W_{pq}(t+1) = -\mu \frac{\delta E}{\delta W_{pq}} + \alpha \cdot W_{pq}(t)$$

in cui si è indicato con μ il Learning Rate ed il termine $\alpha \cdot W_{pq}(t)$ ha un compito di tipo "inerziale" riducendo le oscillazioni dei $W(t+1)$ ed agevolando il passo in situazioni di costanza di andamento.

E' indicato e modificabile il numero di Learning Patterns: ora scegliendo la casella "RUN Step1" si attiva la procedura descritta con i parametri impostati; scegliendo ancora la casella "DER" si abilita o meno il grafico dell'andamento della derivata dell'errore quadratico medio.

Step2: consiste in una ripetizione della procedura di apprendimento effettuata tante volte quanto è scritto nella casella "N° iterazioni". Ciò è dovuto alla evidenza che se si risottopongono alla procedura gli stessi patterns già fissati, utilizzando i pesi ottenuti dalla precedente iterazione, il sistema seguita ad "apprendere" diminuendo via via l'errore della risposta. In questa fase LR varia per ogni Pattern: inizialmente è associato a tutti il valore indicato nell'apposita casella ma, al procedere delle iterazioni, il sistema ricerca per ognuno il valore che minimizza l'errore quadratico medio complessivo (è inutile

inserire un termine di momento differente da zero).

In definitiva nello Step1 si possono avere LR variabili per ogni connessione e per ogni pattern (con $\beta_1 \neq 0$ e $\beta_2 \neq 1$) mentre nello Step2 si hanno LR uguali per tutte le connessioni ma variabili per ciascun pattern cercando, in modo iterativo, il valore di LR migliore per quel pattern.

Step3: non è propriamente una fase di apprendimento ma una fase di ottimizzazione dei pesi al fine di minimizzare gli errori quadratici medi delle singole uscite. Si tratta di variare i pesi ottenuti dalle fasi di apprendimento precedenti in modo da ottenere il minimo errore totale minimizzando gli errori delle singole uscite.

Risultati ottenuti

Si è verificato che il minimo errore raggiunto è stato ottenuto:

a) attraverso una prima fase di ricerca manuale, per tentativi, di un adeguato LR, costante per tutte le connessioni e per tutti i patterns, con lo Step1;

b) attraverso una seconda fase in cui, partendo dal valore di LR precedentemente trovato, il sistema identifica un LR ottimale e variabile per ogni pattern, con lo Step2. Il numero di iterazioni è stato di 20;

c) attraverso una ultima fase di variazione dei singoli pesi, ottimizzando le singole uscite.

Analisi dei risultati

Come si può vedere dalla Fig.1 le distribuzioni delle uscite ottenute sono sostanzialmente identiche a quelle delle uscite attese sia come valore medio che come deviazione standard

Le distribuzioni degli errori sono molto più ristrette di quelle ottenute nel caso precedentemente analizzato e l'errore rms assoluto è quasi la metà dell'altro e pari a rms totale = 0. 077.

Sulle singole uscite abbiamo:

Valori:	attuali	precedenti	iniziali
R:	rms = 0.047	0.072	0.140
Z:	rms = 0.020	0.044	0.169
Λ :	rms = 0.077	0.101	0.179

Gli errori percentuali sulla media, riferiti al valore massimo, sono in tutti i casi nulli.

Dalla Fig. 1 si possono vedere i confronti tra le uscite attese e le ottenute con evidente miglioramento rispetto alle precedenti.

In Allegato 1 sono riportati i parametri significativi dell'esempio in questione (la casella "FCO" nella schermata iniziale abilita

o disabilita la antitrasformazione dei dati utilizzando o meno i Fattori di Conversione sia nell'analisi dei risultati che nella loro successiva presentazione grafica).

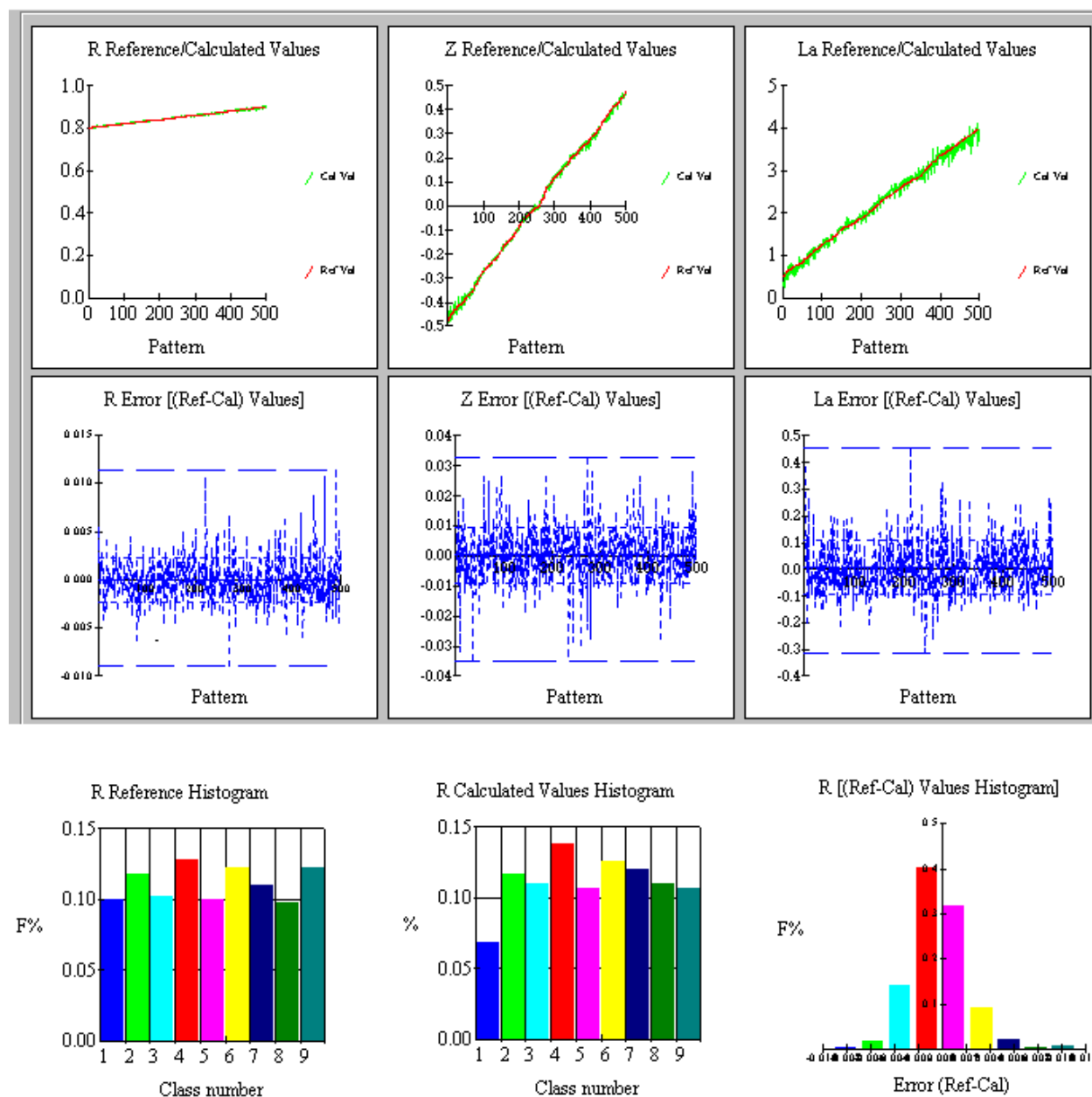
Conclusioni

Con l'architettura fissata sembra che quello ottenuto sia il miglior risultato raggiungibile.

Ulteriori tentativi potrebbero essere fatti variando il coefficiente Rsig ed il numero di iterazioni dello Step2.

Ultimo passo è quello di aumentare a 12 i neuroni di input ed a 10 quelli Hidden.

Fig.1



Neural Network

Architecture and Initial Steps

Not Normalized Data Banc: D:\VB4\PROGRAM\NEUNET\PLASMA1\PROVA\PROVA.BAD
Total Patterns = 1000
Open Config File

Setting:
Input: 50
Hidden: 10
Output: 1
(Layers = 3)

Data Banc Normalization
Input File:
Conv. Fac.

Initial Weights RUN
(N = 500)
B = 0.5
Rsig = 3.5
Rlin = 0.9
Sigmoid Trend

Initial Weights View
Graphic View

Resultant Data:
n° Bias = 11
n° Weights = 510
Total Elements = 521

CONF Generation
CONF Save
Quit

Training

Step 1 Inputs
From 1 to 500
Learning Rate 0.5
Momentum 0
B1 0
B2 1

Step 2 Inputs
From 1 to 500
Learning Rate 0.5
Momentum 0
Files from Step Init.
Max Iter. n 1

Step 3 Inputs
From 1 to 200
Files from Step step3

Mean Square Error Derivative Trend
Training RUN

Testing

Testing Inputs
Weights from Files:
D:\VB....\T\PLASMA1\PROVA\PROW1.ST3
D:\VB....\T\PLASMA1\PROVA\PROW2.ST3

Patterns from: 501
to: 1000
MSE Derivative Trend
Testing RUN

Analysis

Analysis Inputs
Patterns from: 501
to: 1000
Conversion Factors use
Analysis RUN