

Manuale sintetico programma Neural 1.0

da <http://escher07.altervista.org>

La form principale

La maschera iniziale del programma è questa:

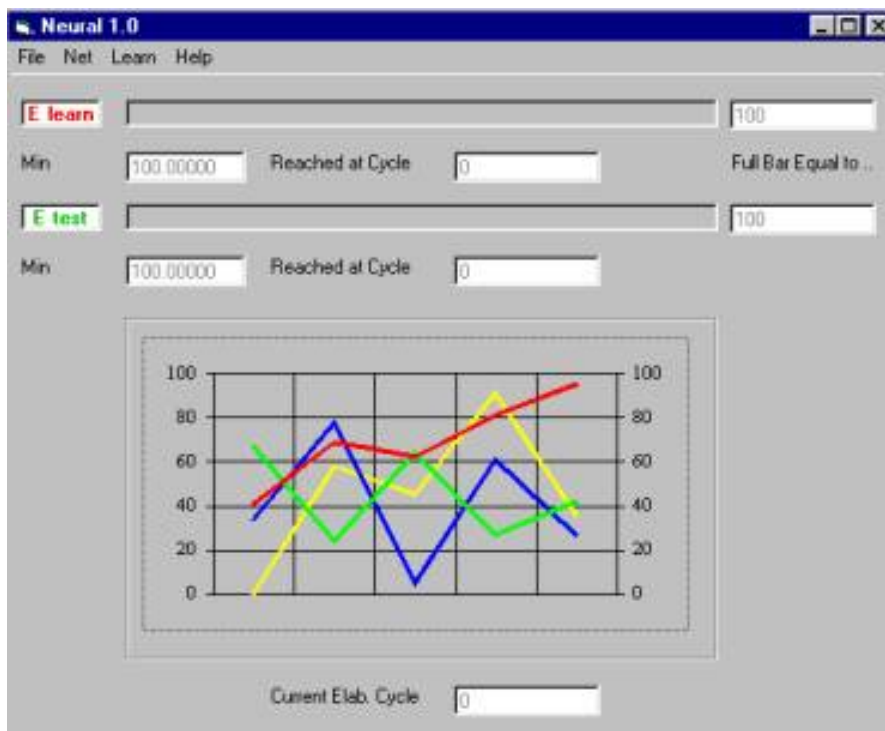


Figura 1

Funzioni Disponibili

File

- *Load Net File*
- *Save Net File*
- *No Matrix Log File (Default: Checked)*
- *No Error Log File (Default: Checked)*

Net

- *Parameters Definition*
- *I/O Definition*
- *On Line Output*

Learn

- *Start*
- *Auto Stop Config*
- *Manual Stop*

Help

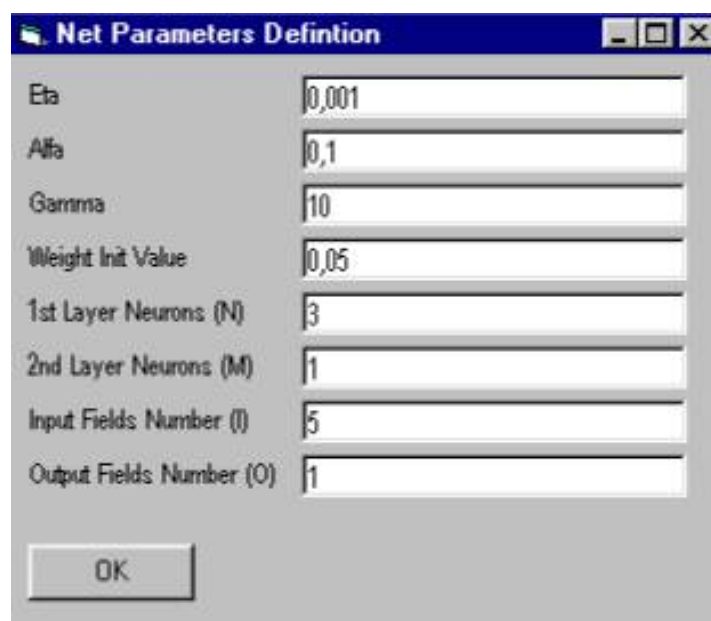
- *About*

Utilizzo di base

Per iniziare un nuovo processo di apprendimento i passi da seguire sono questi:

1. *Definire la struttura della rete*
2. *Definire i file degli ingressi e delle uscite dell'insieme di apprendimento (ed eventualmente di quello di test).*
3. *Far partire il processo di apprendimento*
4. *Fermare il processo di apprendimento*
5. *Salvare la rete ed il relativo apprendimento*

*Il punto 1) si attua attraverso la funzione **Net/ Parameters Definition** a seguito della quale compare la seguente maschera:*



Parameter	Value
Eta	0,001
Alfa	0,1
Gamma	10
Weight Init Value	0,05
1st Layer Neurons (N)	3
2nd Layer Neurons (M)	1
Input Fields Number (I)	5
Output Fields Number (O)	1

Figura 2

in cui si specificano i valori delle costanti del processo di apprendimento (Eta, Alfa, Gamma, Weight Initial Value) e i parametri topologici della rete (N, M, I,O). La rete qui implementata è una Feedforward a due strati intermedi con algoritmo di apprendimento Back Propagation (rimando a testi specifici per la loro descrizione).

Nel punto 3 si deve innanzitutto dire al programma dove sono e quali sono i file dell'insieme di apprendimento e dell'insieme di test da utilizzare. Lo si fa tramite la seguente maschera:

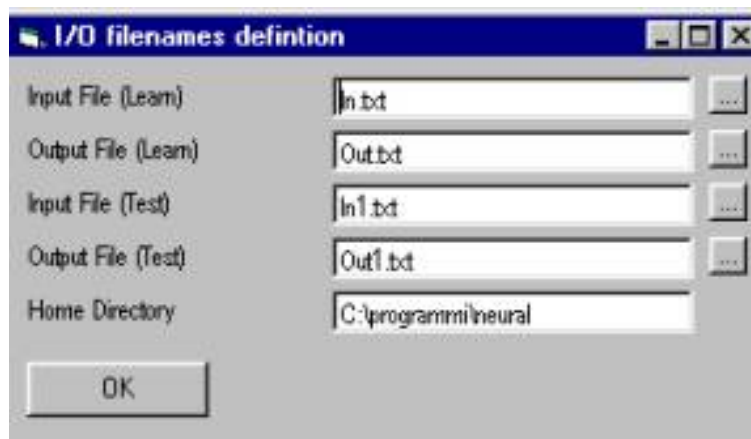


Figura 3

A seguito di questo passo bisogna definire "come sono fatti" tali files. Questi sono di tipo testo "line sequential", dunque si tratta praticamente di dare a Neural le informazioni per attuare la seguente "decodifica": da

File
000133000400000355
000244000555001234
etc...

alla seguente forma:

Rappresentazione Interna		
<i>Ingresso 1</i>	<i>Ingresso 2</i>	<i>Ingresso 3</i>

000133	000400	000355
000244	000555	001234
etc..		

Tutto questo si fa tramite la seguente maschera che viene presentata automaticamente dal programma $V=I+O$ volte (dove I ed O sono stati definiti in Figura 3) ovvero una per ogni ingresso ed una per ogni uscita.

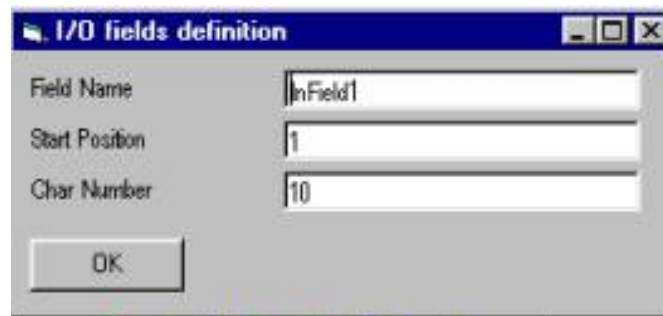


Figura 4

Il punto 3 lo si attua tramite l'opzione **Learn/Start**. A seguito di essa parte l'apprendimento e come si può vedere dal grafico man mano che le iterazioni (**Cycles**) proseguono l'errore di apprendimento (**E learn**) tende a scendere. La maschera relativa a tutto questo è la principale, che assume questa forma:

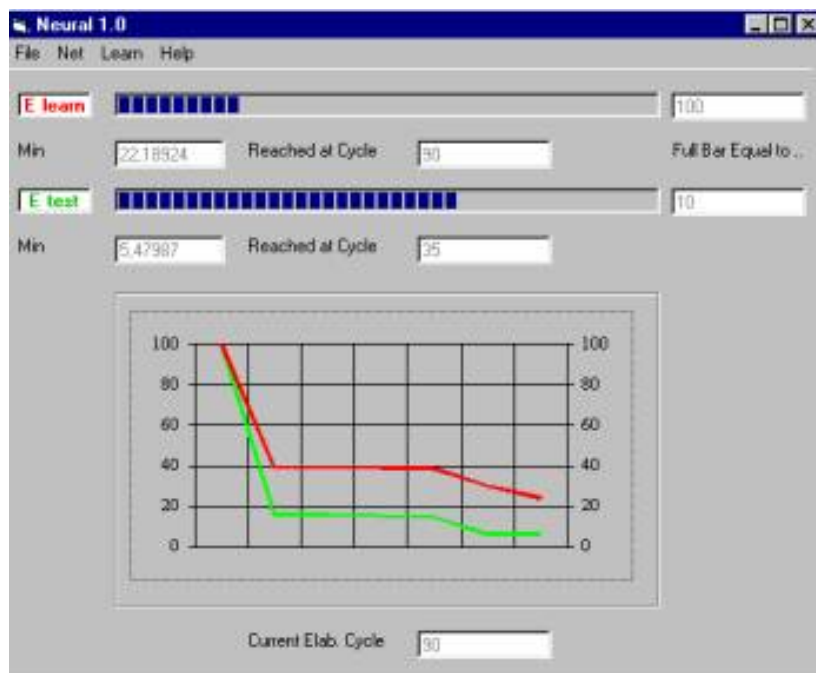


Figura 5

Chiaramente questa maschera si aggiorna di continuo automaticamente man mano che l'elaborazione procede.

Senza stare a dilungarmi su questo aspetto, per il quale rimando ai “sacri testi” l'E learn indica quanto la rete ha imparato a “capire” l'insieme di apprendimento. E' il parametro principale che descrive la sua capacità di associare e, se questo è sopra il 10-20% difficilmente si può avere realizzato una rete di qualche utilità.

L'E test invece riassume la capacità della rete di generalizzare. Questo errore diminuisce inizialmente con E learn, mentre ad un certo punto inizia a crescere. Questa condizione è detta di overlearning ed indica che la rete ha cominciato a “fissare in mente” così bene gli esempi di apprendimento che, in presenza di un ingresso non compreso in questi gli associa comunque una uscita presente nel training set, commettendo un grosso errore.

Morale: l'apprendimento va fermato prima che si manifesti la condizione di overlearning ad un punto in cui però l'E learn è sufficientemente basso.

Questo arresto si ottiene con la funzione **Learn/Manual Stop** (punto 3). Se si è soddisfatti della rete e del suo processo di apprendimento si possono salvare i suoi dati con la funzione **File/Save Net File** (punto 3).

Cenni alle altre funzioni

“No Matrix Log File” e “No Error Log File” corrispondono a due flag da settare prima di iniziare il processo di apprendimento. Essi sono attivi (cioè non si produce alcun log file) di default perchè i log possono essere molto pesanti e rallentare sensibilmente l'elaborazione. Il file di log delle matrici rappresenta l'evoluzione delle matrici di pesi che la rete ricalcola via via ad ogni ciclo, ovvero l'evoluzione del modello black box della funzione di trasferimento I/O che la rete sintetizza. Il secondo descrive invece l'evolversi dell' errore con dettaglio su ciascun campione del training set.

“On Line Output” permette di ricavare l'uscita corrispondente ad un particolare ingresso che l'utente può inserire on line durante l'apprendimento. Utile per testare in corsa l'assenza di overlearning.

“Auto Stop Config” permette di impostare dei criteri di arresto per l'apprendimento, ovvero un'istruzione tipo “fermati quando l'E learn è sceso sotto il 10%”. Utile per far

eseguire alla rete il suo processo di apprendimento in modo automatico senza che esso derivi nell'overlearning.

Le solite note finali

Il prodotto è rilasciato "com'è" senza alcuna garanzia sul suo utilizzo e sugli eventuali danni derivanti da uso improprio. Se volete segnalarmi dei bugs o desiderate maggiori informazioni su qualche aspetto qui citato contattatemi pure, mi farà piacere rispondervi.