

## TLEARN E LE SIMPLE RECURRENT NETWORKS

### Rete per l'AND

- (1) Avviare TLearn
- (2) Selezionare dai menù testuali la voce "Network" > "New project"
- (3) Creare una nuova cartella "and" e dare lo stesso nome al nuovo progetto ("and")
- (4) Si apriranno automaticamente tre file:
  1. **and.cf** (file di configurazione della rete)
  2. **and.data** (file dove andranno messi gli input che vogliamo fornire alla rete)
  3. **and.teach** (file dove metteremo l'output atteso dato l'input definito in and.data)
- (5)
 

<b>and.cf</b>	<b>and.data</b>	<b>and.teach</b>
NODES:	distributed	distributed
nodes = 1	4	4
inputs = 2	0 0	0
outputs = 1	1 0	0
output node is 1	0 1	0
CONNECTIONS:	1 1	1
groups = 0		
1 from i1-i2		
1 from 0		
SPECIAL:		
selected = 1		
weight_limit = 1.00		
- (6) Selezionare "Displays" > "Error display" e "Network Architecture"; quindi "Window" > "Tile"
- (7) Selezionare "Network" > "Training options"; premere "more..." per vedere tutti i parametri settabili
- (8) Selezionare "Network" > "Train the Network" (shortcut = CTRL+T)
- (9) Variare a piacimento i parametri di apprendimento e riprovare ancora "Train the Network"
- (10) Selezionare "Displays" > "Node activations" e "Connection Weights"; quindi "Window" > "Tile"
- (11) Variare a piacimento i parametri di apprendimento e riprovare ancora "Train the Network"
- (12) Provare a sostituire la sequenza <0, 0, 0, 1> nel file "and.teach" con: <0, 1, 1, 0>
- (13) Variare a piacimento i parametri di apprendimento e riprovare ancora "Train the Network"
- (14) Salvare tutto e chiudere il progetto

### Rete per l'XOR

<b>xor.cf</b>	<b>xor.data</b>	<b>xor.teach</b>
NODES:	distributed	distributed
nodes = 4	4	4
inputs = 2	0 0	0
outputs = 1	1 0	1
output node is 4	0 1	1
CONNECTIONS:	1 1	0
groups = 0		
1-3 from i1-i2		
4 from 1-3		
SPECIAL:		
selected = 1-3		
weight_limit = 1.00		

### Simple Recurrent Network

<b>srn.cf</b>	<b>srn.data</b>	<b>srn.teach</b>
NODES:	localist	localist
nodes = 25	17	17
inputs = 5	2	3
outputs = 5	3	5
output nodes are 21-25	5	1
CONNECTIONS:	1	2
groups = 0	2	3
1-10 from i1-i5	3	4
11-20 from 1-10 = 1. & 1.	4	5
fixed one-to-one	5	1
1-10 from 11-20	1	5
21-25 from 1-10	5	3
SPECIAL:	2	1
linear = 11-20	1	2
weight_limit = 0.1	2	3
selected = 1-10	3	4
	4	5
	5	1

#### file di frasi

gianni  
ama  
maria  
.  
gianni  
ama  
molto  
maria  
.  
maria  
ama  
gianni  
.  
gianni  
ama  
molto  
maria  
.

#### file da usare per la "translation"

MAPPINGS:  
1-5 from frasi

frasi:  
.  
gianni 1  
ama 3  
molto 4  
maria 5

- (15) fare la cluster analysis testando la rete su un nuovo data-set (2,3,4,5) e teach-set (3,4,5,1) (usare un "names file" con le sole parole "gianni" "ama" "molto" "maria" incolonnate (per creare il "vector file" andare su "network" > "probe selected nodes" e salvare l'output numerico su un nuovo file da usare come vector file);

se si vuole fare la cluster analysis/PCA su "file di frasi" fare una media con Excel delle attivazioni per le singole parole in input (altrimenti vengono proiettate più volte sul grafico).

#### Rete per linguaggi ricorsivi di tipo $a^n b^n$

file.cf	file.data	file.teach
NODES:	localist	localist
nodes = 12	14	14
inputs = 2	1	1
outputs = 2	1	2
output nodes are 11-12	2	2
	2	0
CONNECTIONS:	0	1
groups = 0	1	2
1-5 from i1-i2	2	0
6-10 from 1-5 = 1. & 1.	0	1
fixed one-to-one	1	1
1-5 from 6-10	1	2
11-12 from 1-5	1	2
	2	2
	2	0
SPECIAL:	2	
linear = 6-10		
weight limit = 0.1		
selected = 1-5		

- (16) modificare file.data e file.teach per testare anche linguaggi di tipo  $XX$  e  $XX^R$

#### Riferimenti

TLearn si può scaricare gratuitamente al seguente indirizzo web:  
<http://crl.ucsd.edu/innate/tlearn.html>

Plunkett & Elman (1996) *Exercises in Rethinking Innateness: A Handbook for Connectionist Simulations*. MIT Press.