

## 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

(15)	<pre> xor.cf NODES: nodes = 4 inputs = 2 outputs = 1 output node is 4  CONNECTIONS: groups = 0 1-3 from i1-i2 4 from 1-3  SPECIAL: selected = 1-3 weight_limit = 1.00                 </pre>	<pre> xor.data distributed 4 00 10 01 11                 </pre>	<pre> xor.teach distributed 4 0 1 1 0                 </pre>
------	--	---	--

Simple Recurrent Network

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

**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:
.      1
gianni 2
ama    3
molto  4
maria  5
                
```

- (17) 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
(18)	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 11-12	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		
	wheight_limit = 0.1		
	selected = 1-5		

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

**Reti per la traduzione automatica**

- (20) Costruire le macchine a stati finiti (FSA) che codificano le transizioni necessarie per descrivere le frasi seguenti:

**Training**

- a.
- |      |  |
|------|--|
| IT:  | Domani sera il tuo amico vedrà un caro compagno                  |
| ENG: | Tomorrow evening your friend will see a dear friend              |
| FR:  | Demain soir ton ami verra un cher ami                            |
| POR: | Amanhã nivelando seu amigo prestará atenção <b>um caro amigo</b> |
| DE:  | Morgen abend sieht Ihr Freund einen lieben Freund                |
- b.
- |       |                                 |
|-------|---------------------------------|
| IT:   | Il vostro amico partirà domani  |
| ENG : | Your friend will leave tomorrow |
| FR:   | Votre ami partira demain        |
| POR:  | Seu amigo sairá amanhã          |
| DE:   | Uw vriend zal morgen weggaan    |

**Testing**

- c.
- |       |   |
|-------|---|
| IT:   | Un caro compagno partirà domani sera      |
| ENG:  | A dear friend will leave tomorrow evening |
| FR:   | Un cher ami partira demain soir           |
| POR : | Um caro amigo deixará amanhã nivelar      |
| DE:   | Een beste vriend zal morgenavond weggaan  |

- (21) Espandere la rete in (15) in modo da avere un numero di neuroni di input sufficienti a codificare tutti gli archi tra le parole nelle frasi di training in italiano e tanti neuroni di output quanti ne sono necessari per codificare tutte le transizioni delle FSA delle traduzioni (state così costruendo un MLP simile a quello descritto in Prat, Casacuberta & Castro (2001))

- (22) Espandere la rete in (16) in modo da avere un numero di neuroni di input sufficienti a codificare un intorno di tre parole e tanti neuroni di output quanti ne sono necessari per codificare una parola tradotta (così si costruisce una **SRN** simile a quella descritta in Castano & Casacuberta (1997))

#### 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.

Per le reti impiegate nella traduzione automatica vedere:

Castano & Casacuberta (1997) A connectionist approach to machine translation. In *Procs. of the EuroSpeech'97*, volume 1, pag. 91-94

Prat, Casacuberta & Castro (2001) *Machine Translation with Grammar Association: Combining Neural Networks and Finite-State Models*. II Workshop on Natural Language Processing and Neural Networks