

TLEARN AND SIMPLE RECURRENT NETWORKS

Learning AND operator

- (1) Start TLearn
- (2) Select "Network" > "New project"
- (3) Create a new folder "and" and choose a project name ("and")
- (4) These files are generated:
 1. **and.cf** (network configuration)
 2. **and.data** (inputs)
 3. **and.teach** (output according to the inputs in and.data)
- (5)

and.cf	and.data	and.teach
NODES:	distributed	distributed
nodes = 1	4	4
inputs = 2	1 1	1
outputs = 1	1 0	0
output node is 1	0 1	0
	0 0	0
CONNECTIONS:		
groups = 0		
1 from i1-i2		
1 from 0		
SPECIAL:		
selected = 1		
weight_limit = 1.00		
- (6) Select "Displays" > "Error display" and "Network Architecture"; then "Window" > "Tile"
- (7) Select "Network" > "Training options"; press "more..." to set other parameters
- (8) Select "Network" > "Train the Network" (shortcut = CTRL+T)
- (9) Change training parameters and options and re-"Train the Network"
- (10) Select "Displays" > "Node activations" and "Connection Weights"; then "Window" > "Tile"
- (11) Substitute <0, 0, 0, 1> in the file "and.teach" with: <0, 1, 1, 0>
- (12) Change training parameters and options and re-"Train the Network"
- (13) Save the project and close it

XOR network

(14)	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	xor.data distributed 4 1 1 1 0 0 1 1 1	xor.teach distributed 4 0 1 1 0
------	--	---	--

SimpleRecurrent Network

(15)	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. & 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	srn.data localist 17 2 3 5 5 1 2 3 4 5 1 5 3 2 1 2 1 2 3 4 5 5	srn.teach localist 17 3 5 1 2 3 4 5 1 5 3 2 1 2 3 4 5 1
------	---	--	---

Sentence file

John
 kisses
 Mary
 .
 John
 often
 kisses
 Mary
 .
 Mary
 kisses
 John
 .
 Mary
 often
 kisses
 John
 .

"translation" file

MAPPINGS:
 1-5 from frasi

 frasi:
 . 1
 John 2
 kisses 3
 often 4
 Mary 5

- (16) test for cluster analysis a new data-set (2,3,4,5) and teach-set (3,4,5,1) (use a "names file" with the words "John" "kisses" "often" "Mary" one per line (for creating a "vector file" go on "network" > "probe selected nodes" and save only the numeric part of the output in a file, named vector_file or similar);

Network for learning $a^n b^n$ (counting) recursion

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

- (18) modify file.data and file.teach for testing XX e XX^R recursion

References

TLearn can be freely downloaded from here:
<http://crl.ucsd.edu/innate/tlearn.html>

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