

Pseudocode Algorithm – Adaptive Model

Neuron n = null

Subclass ScHighError = 0; //subclass with the highest error

Subclass actualSc = 0 // actual subclass specified by the pattern

Initialize input & target values

Create Subclasses (–objects) with one neuron per subclass with the given target values and map it with the neurons

while *NET_ACCURACY()* better than before (2 loops before first check)

CREATE_NET()

Initialize PatternSet

TRAINER.train()

for each pattern

n = getWinningNeuron()

actualSc = pattern.subclass()

actualSc.numberOfUses() + 1

if *n.subclass() = actualSc* //winning neuron is in correct subclass

if *n ≠ expected Outputneuron (n.subclass() ≠ actualSc.outputneuron())*

set *n* as outputneuron (value 1) in target values and old outputneuron 0

else

actualSc.numberOfWrongOutput() + 1

if *actualSc.subclass().error > ScHighError.error*

ScHighError = actualSc

actualSc.safeWrongPattern(pattern)

ScHighError.addNeuron() //in wrong patterns: set old outneuron 0, added neuron 1

end while

Subclass().error = numberOfWrongOutput() / numberOfUses() = % indication of error frequency

NET_ACCURACY(): = ScHighError = Subclass with highest errorrate is the accuracy of net