



Neural networks

... in practice

AForge.NET

- Andrew Kirillov, LGPL v3
- AI libraries , C# language
- published with source code and examples
- aforgenet.com

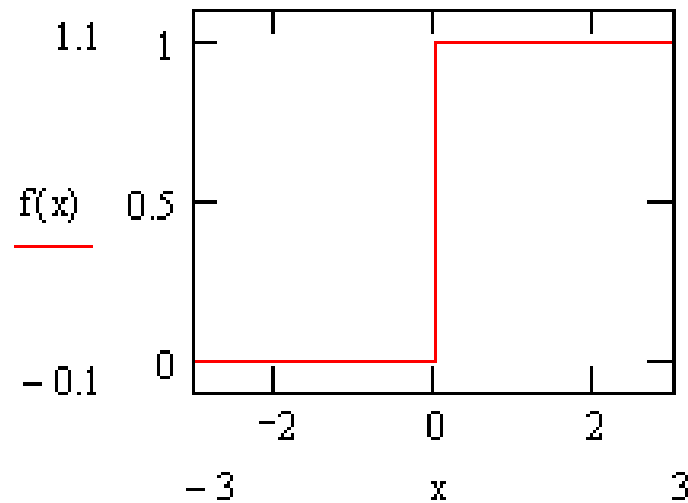
- In this lab: two simplified examples.

Perceptron example

- Single-layer network for classification of 2D points.
- Classes:
 - ActivationNeuron
 - ActivationLayer
 - ActivationNetwork
 - PerceptronLearning

Perceptron example

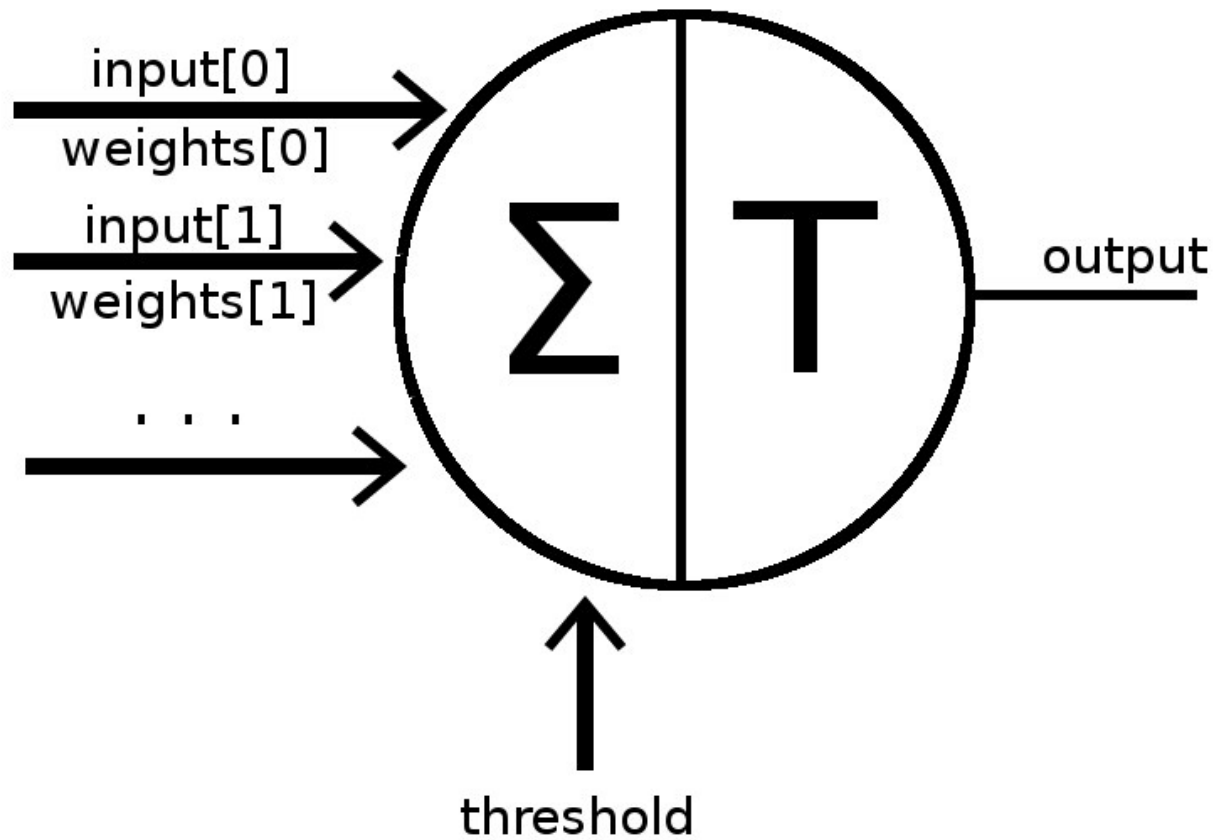
- Transfer function: step function



Perceptron example

- Complete the application with the following algorithms:
 - computing neuron output
 - supervised training: weights correction

Variables



Solution Part 1

```
double sum = 0.0;
for (int i = 0; i < inputsCount; i++)
{
    sum += weights[i] * input[i];
}
sum += threshold;
output = sum > 0 ? 1 : 0;
return output;
```

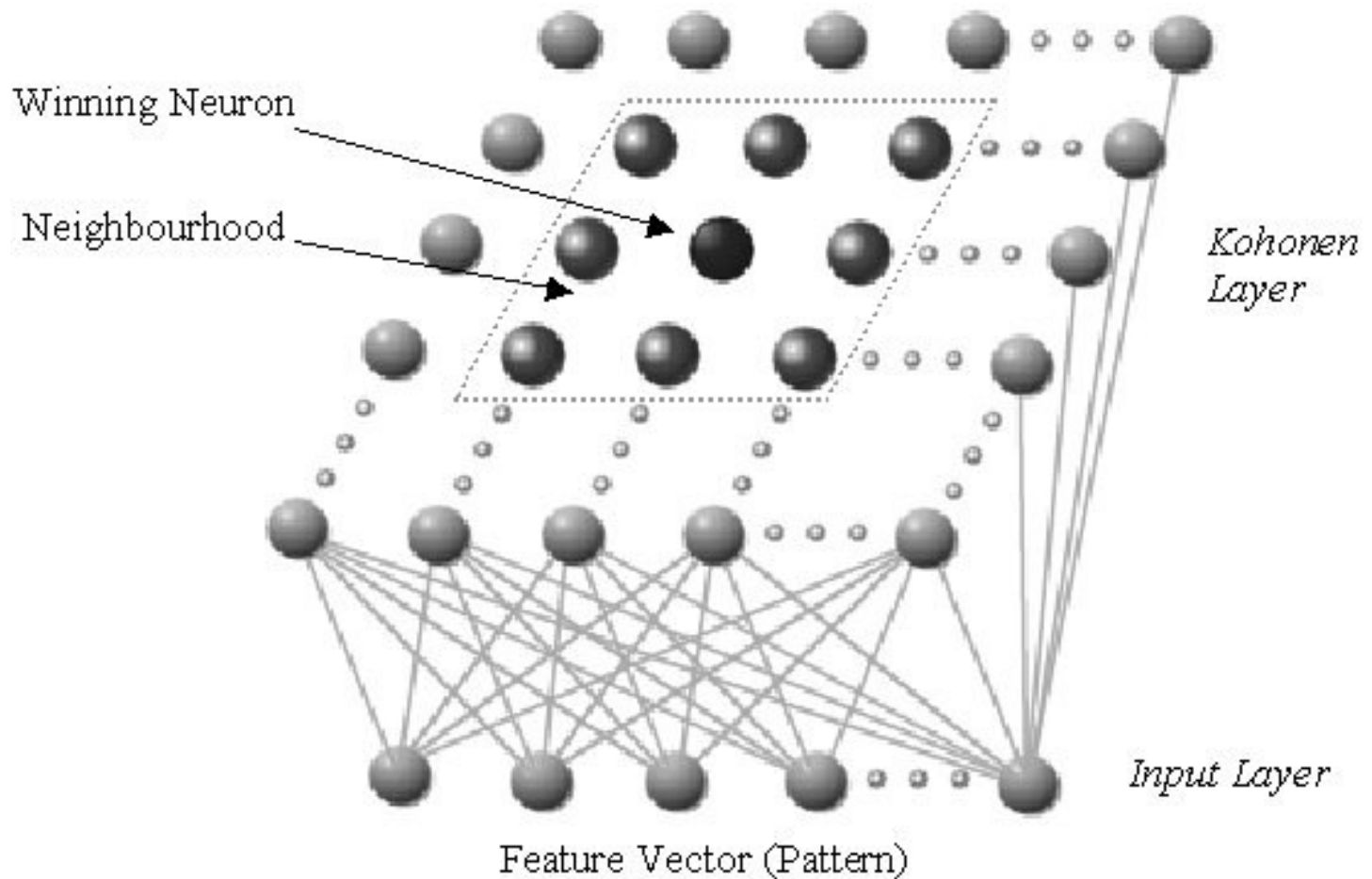
Solution Part 2

```
for (int i = 0; i < layer.NeuronsCount; i++)
{
    double e = output[i] - networkOutput[i];
    if (e != 0) {
        ActivationNeuron neuron = layer[i];
        for (int j = 0; j < neuron.InputsCount; j++)
        {
            neuron[j] += learningRate*e*input[j];
        }
        neuron.Threshold += learningRate * e;
        error += Math.Abs(e);
    }
}
```


Kohonen networks

- Self-organizing Map (SOM)
- Unsuperwised learning
- Representing multi-dimensional data in 2 dimensions (finding relationships)

Kohonen networks



Source: http://homepage.ntlworld.com/richard.clark/rs_kohonen.html

Kohonen networks

- Neuron output: difference between input and weight
- Winner neuron: closest to input
- Training: only winner and neurons in its neighborhood learn. Simplest method:

$$w_i(t+1) = w_i(t) + \alpha * (I_i - w_i(t))$$

Purpose: get weights even closer to input.

Kohonen networks

- Learning coefficient and neighborhood radius change during training!

Simplest method:

$$\alpha(t+1) = \alpha(0) * (1 - t / n)$$

$$r(t+1) = r(0) * (1 - t / n)$$

t: current iteration

n: total number of iterations

SOM example

- Demonstrating SOM by mapping colored pixels into 2D
- Classes:
 - DistanceNeuron
 - DistanceLayer
 - DistanceNetwork
 - SOMLearning

SOM example

- Complete the application with the following algorithms:
 - computing neuron output
 - unsupervised training: weights correction
 - adjusting learning rate and neighborhood radius

Solution Part 1

```
double diff = 0.0;
for (int i = 0; i < inputsCount; i++)
{
    diff +=
        Math.Abs(weights[i] - input[i]);
}
output = diff;
```

Solution Part 2

```
if (Math.Sqrt((dx*dx+dy*dy))< learningRadius)
{
    for (int i=0; i<neuron.Weights.Length; i++)
    {
        //difference:
        double e = input[i] - neuron.Weights[i];
        neuron.Weights[i] += learningRate * e;
        error += Math.Abs(e);
    }
}
```


Solution Part 3

```
trainer.LearningRate =  
learningRate*(1.0-(double)i/iterations);
```

```
trainer.LearningRadius =  
radius*(1.0-(double)i/iterations);
```

Resources

- AForge.Net

<http://www.aforgenet.com/aforge/framework/>

- Kohonen's Self Organizing Feature Maps

<http://www.ai-junkie.com/ann/som/som1.html>