Neural Networks
Practical Assignment I: Perceptron training

Rick van Veen (s1883933)   Laura Baakman (s1869140)

November 25, 2014

I. Introduction

Artificial neural networks are non-linear mapping systems inspired by biological nervous systems. The most import parts of a biological neuron include a dendrite that receives signals from other neurons, a soma that integrates the signals and generates a response that is distributed via a branching axon.

An artificial neural network consists of a large numbers of simple processors linked by weighted connections, analogously the neurons. Each processor receives inputs from many other nodes and generates a single scalar output that depends on locally available information. This scalar is distributed as input to other nodes.

The most simple case of a neural network is the perceptron, see Figure 1, This network consists of one layer of input nodes, connected to a processing unit through a single layer of weights, which determine the result of the output node. Mathematically a perceptron is any feed-forward network of noes with responses like $f(\vec{w}^T\vec{x})$ where $\vec{w}$ is the vector of weights, $\vec{x}$ is the pattern and f is a sigmoid-like squashing function[1]. This squashing function ensures the binary output of the perceptron.

The perceptron is a linear binary classifier, these classifiers separate classes via a hyperplane. Homogeneously separable data sets can be separated by a hyperplane through the origin, inhomogeneously separable data sets cannot be separated by a hyperplane through the origin but they can be separated by a hyperplane if it is offset with a certain bias with respect to the origin.

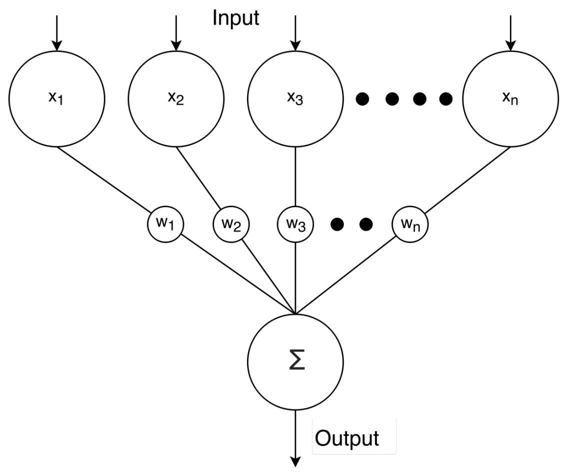

Figure 1: *A perceptron.*

It can be shown that the perceptron converges, i.e. it separates positive examples for the negatives, if the input data set is linearly separable.

II. Method

Given a dichotomy $\mathcal{D}$ with N d-dimensional patterns $\xi \in \mathcal{R}^d$ each with a label $S \in \{-1, +1\}$: $\mathcal{D} = \left\{\xi^i, S^i\right\}_i^N$, a perceptron can be trained using the Rosenblatt algorithm which updates the weights each time step $t = 1, 2, \ldots$:

$$\vec{w}(t+1) = \begin{cases} \vec{w}(t) + \frac{1}{d}\xi^{\mu(t)}S^{\mu(t)} & \text{if } E^{\mu(t)} \leq 0 \\ \vec{w}(t) & \text{otherwise} \end{cases} \tag{1}$$

Where $\mu(t) = 1, 2, \ldots, N, 1, 2, \ldots$ denotes the present pattern. $E(\cdot)$ the energy function is defined as:

$$E^{\mu(t)} = \vec{w}(t) \cdot \xi^{\mu(t)}S^{\mu(t)}. \tag{2}$$

The energy function indicates if the perceptron gives the correct output for the given input pattern $\xi^{\mu(t)}$. Since we know that independent of the initial value of the weights the perceptron will converge, if the $\mathcal{D}$ is linearly separable, we can start with $\vec{w} = 0$.

The update defined in Equation 1 is executed until $E^{\xi^i} > 0$ for $i \in [1, N]$ in this cases the algorithm has converged. If the dataset is not linearly separable the perceptron never converges, thus it is generally a good idea to set a maximum number of time steps, d_{max}. The total number of steps taken by the algorithm has thus the upper bound $d_{max} \cdot N$.

To classify a inhomogenously separable dataset with a perceptron one needs to add a one extra input to all patterns, namely minus one and one extra weight, associated with the extra input. Since we can thus classify both homogenously and inhomogenously data sets with a perceptron we will ignore this distinction from here on on.

The chance that a randomly chose dichotomy is linearly separable can be proven to be [1]:

$$f(N,d) = \begin{cases} 1 & N \leq d+1 \\ \frac{2}{2^N} \sum_{k=0}^{d} \binom{N-1}{k} & \text{otherwise} \end{cases} \tag{3}$$

To verify this theoretical result we have performed two experiments.

I. Experiment I

II. Experiment II

Experimenten uitleggen

References

[1] Russell D Reed and Robert J Marks. *Neural smithing: supervised learning in feedforward artificial neural networks*. Mit Press, 1998.