Neural Networks
Practical Assignment 2: Learning a Rule

Rick van Veen (s1883933) Laura Baakman (s1869140)

January 7, 2015

I. Introduction

The perceptron introduced by Rosenblatt inspired many variations, one of those is the Minover algorithm by Krauth and Mézard. This algorithm aims to find the largest possible margin between two linearly separable classes. Figure 1 illustrates the optimal separation between two classes according to the Minover algorithm and the area of all suboptimal separations that classify each available pattern correctly, also called the "version space".

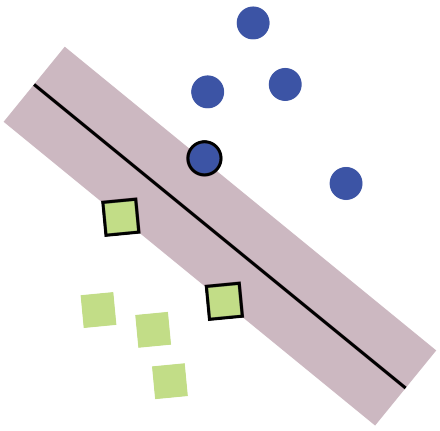

Figure 1: *The separation of two classes, with the optimal boundary shown as the line. All suboptimal separations of the the two classes that separate the points correctly lie in the shaded area. The support vectors are indicated by the symbols with a border.*

Figure 1 illustrates that the optimal boundary is determined by only three points, the support vectors, indicated by symbols with a border in fig. 1. Changing the location of the other patterns does not influence the optimal decision boundary as long as they do not fall within the version space.

II. Method

Given a dichotomy $\mathcal{D} = \{\vec{\xi}^i, S^i\}_i^P$ with P N-dimensional patterns $\vec{\xi} \in \mathcal{R}^N$. The labels S were assigned by a teacher perceptron according to:

$$S^\mu = \text{sign}(\vec{w}^* \cdot \vec{\xi}^\mu) \tag{1}$$

The teacher $\vec{w}^*$ can be chosen randomly as long as the following condition holds:

$$\|\vec{w}^*\|^2 = N. \tag{2}$$

The Minover algorithm is an iterative procedure that runs over a period of time $t = 0, 1, 2, \ldots, t_{max}$ until either the stability of the solution does not change anymore for P steps or the number of maximal time steps $t_{max} = n_{max} \cdot P$ has been reached reached. Where n_{max} is comparable to the number of epochs in the Rosenblatt algorithm. The stability of a pattern $\vec{\xi}$ with label S given a certain weight vector $\vec{w}$ is defined as:

$$k = \frac{\vec{w} \cdot \vec{\xi} \cdot S}{\|\vec{w}\|}. \tag{3}$$

We say that the stability has converged when the last P calculated generalization errors(5) differentiate less than $0 \pm \varepsilon$.

The stability defined in (3) can be thought of as the distance between all the patterns and the current solution $\vec{w}(t)$. To eventually find the weight vector with maximal stability the algorithm takes the following two steps every iteration: it selects the pattern $\mu(t)$ with the minimal distance/stability (minimal overlap) to the current solution. The current weight

vector $\vec{w}(t)$ is then updated with Hebb's rule, see (4).

$$\vec{w}(t+1) = \vec{w}(t) + \frac{1}{N}\xi^{\mu(t)} S^{\mu(t)} \tag{4}$$

See algorithm 1 for the pseudo code of the procedure described above. The method `notConverged` compares the generalization error (5) of the last P iterations, as explained earlier.

Algorithm 1: *minover*$(\mathcal{D}, n_{max}, \vec{w})$

input : $\mathcal{D} = \{\xi^i, S^i\}_i^P, \forall i\ \xi^i \in \mathcal{R}^N$
n_{max} maximum number of epochs
$\vec{w}^*$ the teacher weights
output: $\vec{w}$ the student weights

```
t_max := n_max · P
t := 0
w := [0, ..., 0]^T                    /* w ∈ R^N */
while t < t_max ∧ notConverged() do
    find μ(t) such that k^μ(t) = min_ν {k^ν(t)}
    w(t+1) = w(t) + (1/N) · ξ^μ(t) · S^μ(t)
    t := t + 1
```

The earlier mentioned generalization error ϵ gives the probability that the teacher vector $\vec{w}^*$ and the student vector $\vec{w}$ disagree:

$$\epsilon = \frac{2\phi}{2\pi} = \frac{1}{\pi}\arccos\left(\frac{\vec{w}\cdot\vec{w}^*}{\|\vec{w}\|\|\vec{w}^*\|}\right). \tag{5}$$

The parameter ϕ mentioned iin (5) is the angle between the teacher and student vector, shown in fig. 2. The probability of a different classification by the two vectors is indicated by the shaded area of the unit circle in fig. 2.

As stated before Minover contrary to Rosenblatt does not stop when a solution is found but continues until the solution with optimal stability is found.

III. Experiment

The performance of the perceptron trained with the Minover algorithm can be measured using the generalization error defined in Equation 5.

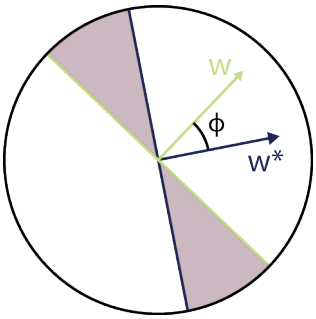

Figure 2: *The difference between the student vector $\vec{w}$ and the weight vector $\vec{w}^*$.*

To explore the behaviour of the Minover algorithm discussed in section II we have tested a perceptron trained with this algorithm on several N-dimensional datasets with $P = \alpha N$, for $N = 10$ and $\alpha = 0.25, 0.5, \ldots, 5$. To ensure that the dataset was linearly separable we have determined its labels via (1), using the weight vector $\vec{w}^* = [1, \ldots, 1]^T$.

Figure 3 shows the final generalization error for different values of α as an average of $n_d = 20$ iterations with each α. We consider a generalization error final when it has converged or when t_{max} has been reached.

Based on fig. 3 we can state that in general the generalization error decreases as α increases. A small α results in a smaller number of patterns, this means that it is more probable that the teacher is not the optimal solution, because of this the generalization error is likely to be larger than with a large α which results in a larger

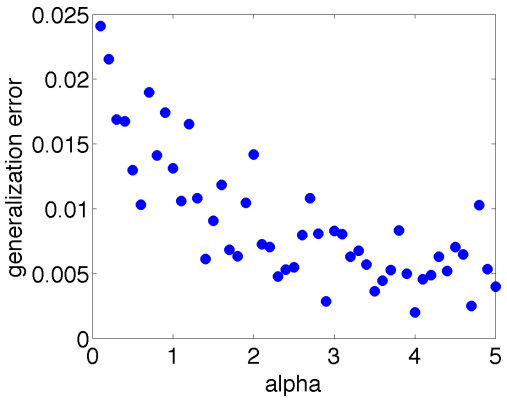

Figure 3: *Generalization error at the final step of training as an average of twenty iterations for $\alpha = 0.25, 0.5, \ldots, 5$, $\varepsilon = 0.0005$, $n_{max} = 500$ and $N = 10$.*

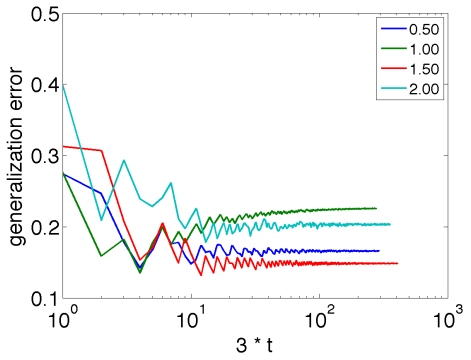

Figure 4: *Learning curve for different* α *for* $N = 5$*, and* $\varepsilon = 0.0005$ *and* $n_{max} = 500$*.*

number of patterns and less possible solutions, thus meaning a higher probability the teacher lies near (or is) the solution with maximum stability.

The fluctuations in the ϵ_g in fig. 3 may be due to the random data sets.

Figure 4 shows the learning curves for different values of α for one dataset per α. This figure clearly shows that for smaller values of α the algorithm needed less steps to converge. Furthermore for nearly all values of α the generalization decreased strongly before decreasingly oscillating around its final generalization error. The curves also show that the error can increase. This is because the found solution at that point in time maybe close to the teacher, but still can increase in stability, thus the final solution may lay further away from the teacher (resulting in a higher error).

One possible explanation for these oscillations is that the optimal weights cannot be formulated as a linear combination of the support vectors. In this case the algorithm continues to add and subtract vectors from the weights, resulting in a constantly changing generalization error. Since our convergence criterion is the stabilization of the generalization error for P steps, these oscillations prevent convergence.

Smaller datasets converge faster due to the fact that they have less support vectors and thus have fewer factors to consider when computing the optimal weights.

References

[1] Werner Krauth and Marc Mézard. "Learning algorithms with optimal stability in neural networks". In: *Journal of Physics A: Mathematical and General* 20.11 (1987), p. L745.

[2] Frank Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain." In: *Psychological review* 65.6 (1958), p. 386.