The Rosenblatt Perceptron

Christoph Netz & Theo Pannetier

I. Introduction

THE perceptron is a single-layered, binary classifier algorithm written by Frank Rosenblatt in 1957. Initially developed as a model for how the biological brain stores information, it showed how a machine could learn to accurately classify objects after being trained on a set of example data. A major prediction from theoretical studies on the perceptron [1] is that there is a finite upper bound on the number of learning steps needed for the perceptron to converge, implying that the success of the perceptron is guaranteed given enough time. This however assumes linear separability of the data, which is difficult to predict *a priori*. Here, we simulate a number of random training data sets, and measure empirically how often linear separability can be expected.

II. Solution

Input data is presented to the perceptron as a set of feature vectors $\{\boldsymbol{\xi}\}_{\mu=1}^{P}$ in N-dimensions, where P is the number of data points. Each feature vector is associated to a label $\boldsymbol{S}^{\mu} \in \{-1, 1\}$. The problem of the perceptron is then to separate the data linearly, i.e. it tries to find the weights vector $\boldsymbol{w} \in \mathbb{R}^N$ such that $sign(\boldsymbol{w} \cdot \boldsymbol{\xi}^{\mu}) = \boldsymbol{S}^{\mu}$ for all μ.

The pseudocode for the algorithm is presented in Box 1.

Box 1. Pseudocode for the perceptron

```
• Initialize $\boldsymbol{w}(0) = 0$
• repeat
•     for each $\mu$ element from 1 to P
•         if $\boldsymbol{w} \cdot \boldsymbol{\xi}^{\mu}\boldsymbol{S}^{\mu} \leq 0$
•             update $\boldsymbol{w}(t+1) = \boldsymbol{w}(t) + \frac{1}{N}\boldsymbol{\xi}^{\mu}\boldsymbol{S}^{\mu}$
•         else
•             do nothing
•         end if
•     end for
• until $\boldsymbol{w} \cdot \boldsymbol{\xi}^{\mu}\boldsymbol{S}^{\mu} \geq 0$ for all $\mu$ or $n_{max}$ is reached.
```

The input vectors are randomized by drawing N values from a normal distribution with parameters $\{\mu = 0; \sigma = 1\}$. Labels for each vector are drawn from $\{-1, +1\}$ with probability $p = 0.5$. To measure how often a random data set is linearly separable, we set $N = 20$ and $P = \alpha N$, with $\alpha = \{0.75, 1.0, 1.25, 1.5, 1.75, 2.0, 2.25, 2.5, 2.75, 3.0\}$. We run the algorithm for $nD = 50$ independently generated data sets for each value of α and measure the empirical proportion $Q_{l.s.}$ of linearly separable random data sets. We assume that a data set is not linearly separable if the perceptron has failed to converge before $n_{max} = 100$ is reached.

III. Results

The number of input points in a random data set was found to have an important effect on the probability for the data to be linearly separable (Fig. 1).
For $P \leq N$, data was found to always be linearly separable. Beyond this value, the frequency of linear separability decayed down to 0 for $P = 2.5N$.

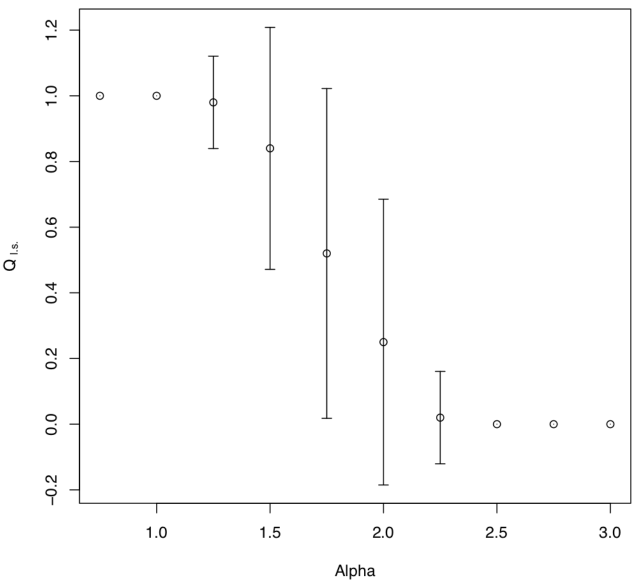

Fig. 1. Average frequency $Q_{l.s.}$ of linear separability for a random data set with $P = \alpha/N$ points. The frequency drops sharply beyond $\alpha = 1.75$.

IV. Discussion

Here we have found that the probability that a random data set with binary labels is linearly separable in N-dimension decreases as data are added. In our experimental setting, this can be interpreted as a lower number of possible dichotomies for more data points, restricting the version space of our data and making it less likely for the data to be linearly separable. The threshold of α beyond which linear separability cannot be found for the limiting behaviour defines the storage capacity of the perceptron. Below this value, the algorithm can converge quickly and bring a perfect classification of the data, and hence has no need to learn - data is resumed (stored) by the weight vectors.

Predicting the probability that a given data set is linearly separable is of interest to verify the validity of the perceptron convergence theorem and hence anticipate whether a linear classification of the data can be found by the perceptron. The theoretical probability $P_{l.s.}$ can be estimated from the count C of possible dichotomies given {P,N}:

$$P_{l.s.} = \frac{C(P,N)}{2^P} \tag{1}$$

Theoretical results [2] have shown that for the limiting behavior ($N \to +\infty$), $P_{l.s.} = 1$ for $\alpha \leq 2$ and $P_{l.s.} = 0$ for $\alpha > 2$. The storage capacity of the perceptron is then $\alpha_c = 2$, that is, a dataset is expected to be linearly separable for $P \leq 2N$. Our empirical curve fits closely the shape of the theoretical one observed in class. The shape of our curve however suggest that the storage capacity of our perceptron is about $\alpha_c = 1.75$, contrary to the $\alpha_c = 2$ value expected from theory. This implies that our perceptron has a lower rate of success than would be expected, and is likely to be due to the threshold n_{max} we have set for the analysis.

V. Conclusion

The storage capacity of the perceptron highlights an important limit of the ability of the perceptron to classify a set of data: the maximum number of data points that can be classified given a set of features; or conversely, the maximum number of features that can be used to classify a set number of data points. Due to this, the perceptron can be expected to be of little use for high-dimension classification problems. Solutions to non-linearly separable problems include: allowing the perceptron to make some error and try to minimize it; performing the classification on a linearly separable transformation of the data; or resort to more complex architectures that include multiple perceptrons. Although limited to linearly separable problems and binomial classifications, the original design of the perceptron and following studies on its theoretical behaviour opened the way to the development of more advanced classifiers able to address the limits of the perceptron.

References

[1] R. Rosenblatt. *Two theorems of statistical separability in the perceptron.* United States Department of Commerce, 1958.

[2] T.M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE transactions on electronic computers*, (3), pp.326-334, 1965.