The perceptron learning model

Big Data Analytics 2021-2022 - UMONS
Souhaib Ben Taieb

1 Exercise 1

Exercise 1.4

Let us create our own target function f and data set $\mathcal{D}$ and see how the perceptron learning algorithm works. Take $d = 2$ so you can visualize the problem, and choose a random line in the plane as your target function, where one side of the line maps to $+1$ and the other maps to -1. Choose the inputs $\mathbf{x}_n$ of the data set as random points in the plane, and evaluate the target function on each $\mathbf{x}_n$ to get the corresponding output y_n.

Now, generate a data set of size 20. Try the perceptron learning algorithm on your data set and see how long it takes to converge and how well the final hypothesis g matches your target f. You can find other ways to play with this experiment in Problem 1.4.

Figure 1: Source: Abu-Mostafa et al. Learning from data. AMLbook.

2 Exercise 2

Problem 1.3 Prove that the PLA eventually converges to a linear separator for separable data. The following steps will guide you through the proof. Let $\mathbf{w}^*$ be an optimal set of weights (one which separates the data). The essential idea in this proof is to show that the PLA weights $\mathbf{w}(t)$ get "more aligned" with $\mathbf{w}^*$ with every iteration. For simplicity, assume that $\mathbf{w}(0) = \mathbf{0}$.

(a) Let $\rho = \min_{1 \le n \le N} y_n(\mathbf{w}^{*\mathrm{T}}\mathbf{x}_n)$. Show that $\rho > 0$.

(b) Show that $\mathbf{w}^{\mathrm{T}}(t)\mathbf{w}^* \ge \mathbf{w}^{\mathrm{T}}(t-1)\mathbf{w}^* + \rho$, and conclude that $\mathbf{w}^{\mathrm{T}}(t)\mathbf{w}^* \ge t\rho$. *[Hint: Use induction.]*

(c) Show that $\|\mathbf{w}(t)\|^2 \le \|\mathbf{w}(t-1)\|^2 + \|\mathbf{x}(t-1)\|^2$.

[Hint: $y(t-1) \cdot (\mathbf{w}^{\mathrm{T}}(t-1)\mathbf{x}(t-1)) \le 0$ *because* $\mathbf{x}(t-1)$ *was misclassified by* $\mathbf{w}(t-1)$.*]*

(d) Show by induction that $\|\mathbf{w}(t)\|^2 \le tR^2$, where $R = \max_{1 \le n \le N} \|\mathbf{x}_n\|$.

Figure 2: Source: Abu-Mostafa et al. Learning from data. AMLbook.

(e) Using (b) and (d), show that

$$\frac{\mathbf{w}^{\mathrm{T}}(t)}{\|\mathbf{w}(t)\|}\mathbf{w}^* \ge \sqrt{t} \cdot \frac{\rho}{R},$$

and hence prove that

$$t \le \frac{R^2\|\mathbf{w}^*\|^2}{\rho^2}.$$

[*Hint:* $\frac{\mathbf{w}^{\mathrm{T}}(t)\mathbf{w}^*}{\|\mathbf{w}(t)\|\,\|\mathbf{w}^*\|} \le 1$. *Why?*]

In practice, PLA converges more quickly than the bound $\frac{R^2\|\mathbf{w}^*\|^2}{\rho^2}$ suggests. Nevertheless, because we do not know ρ in advance, we can't determine the number of iterations to convergence, which does pose a problem if the data is non-separable.

Figure 3: Source: Abu-Mostafa et al. Learning from data. AMLbook.

3 Exercise 3

Problem 1.4 In Exercise 1.4, we use an artificial data set to study the perceptron learning algorithm. This problem leads you to explore the algorithm further with data sets of different sizes and dimensions.

(a) Generate a linearly separable data set of size 20 as indicated in Exercise 1.4. Plot the examples $\{(\mathbf{x}_n, y_n)\}$ as well as the target function f on a plane. Be sure to mark the examples from different classes differently, and add labels to the axes of the plot.

(b) Run the perceptron learning algorithm on the data set above. Report the number of updates that the algorithm takes before converging. Plot the examples $\{(\mathbf{x}_n, y_n)\}$, the target function f, and the final hypothesis g in the same figure. Comment on whether f is close to g.

(c) Repeat everything in (b) with another randomly generated data set of size 20. Compare your results with (b).

(d) Repeat everything in (b) with another randomly generated data set of size 100. Compare your results with (b).

(e) Repeat everything in (b) with another randomly generated data set of size $1,000$. Compare your results with (b).

(f) Modify the algorithm such that it takes $\mathbf{x}_n \in \mathbb{R}^{10}$ instead of $\mathbb{R}^2$. Randomly generate a linearly separable data set of size $1,000$ with $\mathbf{x}_n \in \mathbb{R}^{10}$ and feed the data set to the algorithm. How many updates does the algorithm take to converge?

(g) Repeat the algorithm on the same data set as (f) for 100 experiments. In the iterations of each experiment, pick $\mathbf{x}(t)$ randomly instead of deterministically. Plot a histogram for the number of updates that the algorithm takes to converge.

(h) Summarize your conclusions with respect to accuracy and running time as a function of N and d.

Figure 4: Source: Abu-Mostafa et al. Learning from data. AMLbook.