P05-06 Perceptron

Luis Sa-Couto[1] and Andreas Wichert[2]

INESC-ID, Instituto Superior Tecnico, Universidade de Lisboa
{luis.sa.couto,andreas.wichert}@tecnico.ulisboa.pt

1 Perceptron

1) Consider the following linearly separable training set:

$$\left\{\mathbf{x}^1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \mathbf{x}^2 = \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \mathbf{x}^3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \mathbf{x}^4 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}\right\}$$

$$\{t_1 = -1, t_2 = +1, t_3 = +1, t_4 = -1\}$$

a) Initialize all weights to one (including the bias). Use a learning rate of one for simplicity. Apply the perceptron learning algorithm until convergence.
b) Draw the separation hyperplane.
c) Does the perceptron converge on the first epoch if we change the weight initialization to zeros?

2) Consider the following linearly separable training set:

$$\left\{\mathbf{x}^1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \mathbf{x}^2 = \begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix}, \mathbf{x}^3 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \mathbf{x}^4 = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}\right\}$$

$$\{t_1 = -1, t_2 = +1, t_3 = +1, t_4 = -1\}$$

a) Initialize all weights to one (including the bias). Use a learning rate of one for simplicity. Apply the perception learning algorithm for one epoch.
b) For an additional epoch, do the weights change?
c) What is the perceptron output for the query point $\begin{pmatrix} 0 \; 0 \; 1 \end{pmatrix}^T$?

3) What happens if we replace the sign function by the step function?

$$\Theta(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$$

Specifically, how would you change the learning rate to get the same results?

4) The perceptron can learn a relatively large number of functions. In this exercise, we focus on simple logical functions.

a) Show graphically that a perceptron can learn the logical NOT function. Give an example with specific weights.

b) Show graphically that a perceptron can learn the logical AND function for two inputs. Give an example with specific weights.

c) Show graphically that a perceptron can learn the logical OR function for two inputs. Give an example with specific weights.

d) Show graphically that a perceptron can not learn the logical XOR function for two inputs.

2 Gradient descent learning

1) Consider the following training data:

$$\left\{\mathbf{x}^{(1)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \mathbf{x}^{(2)} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \mathbf{x}^{(3)} = \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \mathbf{x}^{(4)} = \begin{pmatrix} 3 \\ 3 \end{pmatrix}\right\}$$

$$\left\{t^{(1)} = 1, t^{(2)} = 1, t^{(3)} = 0, t^{(4)} = 0\right\}$$

In this exercise, we will work with a unit that computes the following function:

$$output\left(\mathbf{x};\mathbf{w}\right) = \frac{1}{1 + \exp\left(-2\mathbf{w}\cdot\mathbf{x}\right)}$$

And we will use the half sum of squared errors as our error (loss) function:

$$E\left(\mathbf{w}\right) = \frac{1}{2}\sum_{k=1}^{N}\left(t^{(k)} - output\left(\mathbf{x}^{(k)};\mathbf{w}\right)\right)^2$$

a) Determine the gradient descent learning rule for this unit.

b) Compute the first gradient descent update assuming an initialization of all ones .

c) Compute the first stochastic gradient descent update assuming an initialization of all ones.

2) Consider the following training data:

$$\left\{\mathbf{x}^{(1)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \mathbf{x}^{(2)} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \mathbf{x}^{(3)} = \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \mathbf{x}^{(4)} = \begin{pmatrix} 3 \\ 3 \end{pmatrix}\right\}$$

$$\left\{t^{(1)} = 1, t^{(2)} = 1, t^{(3)} = 0, t^{(4)} = 0\right\}$$

In this exercise, we will work with a unit that computes the following function:

$$output\left(\mathbf{x};\mathbf{w}\right) = \frac{1}{1 + \exp\left(-\mathbf{w}\cdot\mathbf{x}\right)}$$

And we will use the cross-entropy loss function:

$$E(\mathbf{w}) = -\log\left(p\left(\mathbf{t} \mid \mathbf{w}\right)\right) = -\sum_{k=1}^{N}\left(t^{(k)} \log output^{(k)}\left(\mathbf{x}^{(k)}; \mathbf{w}\right) + \left(1 - t^{(k)}\right) \log\left(1 - output^{(k)}\left(\mathbf{x}^{(k)}; \mathbf{w}\right)\right)\right)$$

a) Determine the gradient descent learning rule for this unit.

b) Compute the first gradient descent update assuming an initialization of all ones .

c) Compute the first stochastic gradient descent update assuming an initialization of all ones.

3) Consider the following training data:

$$\left\{\mathbf{x}^{(1)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \mathbf{x}^{(2)} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \mathbf{x}^{(3)} = \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \mathbf{x}^{(4)} = \begin{pmatrix} 3 \\ 3 \end{pmatrix}\right\}$$

$$\left\{t^{(1)} = 1, t^{(2)} = 1, t^{(3)} = 0, t^{(4)} = 0\right\}$$

In this exercise, we will work with a unit that computes the following function:

$$output\left(\mathbf{x}; \mathbf{w}\right) = \exp\left(\left(\mathbf{w} \cdot \mathbf{x}\right)^2\right)$$

And we will use the half sum of squared errors as our error (loss) function:

$$E(\mathbf{w}) = \frac{1}{2}\sum_{k=1}^{N}\left(t^{(k)} - output\left(\mathbf{x}^{(k)}; \mathbf{w}\right)\right)$$

a) Determine the gradient descent learning rule for this unit.

b) Compute the stochastic gradient descent update for input $\mathbf{x} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $t = 0$ initialized with $\mathbf{w} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$ and learning rate $\eta = 2$.

3 Thinking Questions

a) Think about the error functions we have seen. Do you think that one is clearly better than the other? What changes when one changes the error function?

b) Could you implement a statistical machine learning application that classifies a number into two classes, primes and non-primes? What is the main difference between this problem and the salmon and sea bass example from the lecture?