Assingnment 1

Luca Brena

October 18, 2019

1 The Perceptron

1. The size of the output is:
$$[d \times 1]$$
   I used the formula shown in answer 2. Explanation of this answer is the same as for answer 2.

2. $$\bar{y} = \underline{xW} + 1_d b$$
   with:
$$[d \times 1] = [d \times k][k \times 1] + [d \times 1][1 \times 1]$$
   where:
   - $[d \times k] = [number\ of\ samples\ per\ batch \times features\ per\ sample]$
   - $[k \times 1] = [number\ of\ weights \times number\ of\ neurons\ of\ the\ layer]$
   - $[d \times 1] = [number\ of\ samples\ per\ batch \times number\ of\ neurons\ of\ the\ layer]$
   - $[1 \times 1]$ = Scalar rapresenting the bias. In this particular case it's a scalar value because the layer has only one neuron, otherwise it's should be a vector (one bias per neuron).

3. $$E = \frac{1}{2d}(\bar{y} - \hat{\bar{y}})^T(\bar{y} - \hat{\bar{y}})$$
   Where:
   - d : Number of samples of the batch
   - $\bar{y}$: net value
   - $\hat{\bar{y}}$: target value

4. According to the computation graph in figure 1 in order to compute the derivative of the error function w.r.t weights, the path is the following:
$$\frac{\partial E}{\partial \underline{W}} = E \rightarrow \bar{y} \rightarrow m_1 \rightarrow \underline{W}$$

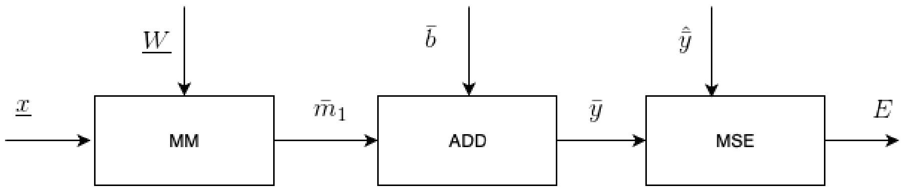

Figure 1: Computation graph of the net

- For the MSE block:

$$\frac{\partial E}{\partial \bar{y}} = \frac{\partial}{\partial \bar{y}} \frac{1}{2d} (\bar{y} - \hat{\bar{y}})^T (\bar{y} - \hat{\bar{y}}) = \frac{1}{d} (\bar{y} - \hat{\bar{y}})$$

- For the Add block:

$$\bar{y} = \bar{m}_1 + \bar{b}$$

$$\frac{\partial \bar{y}}{\partial \bar{m}_1} = 1 \qquad \frac{\partial \bar{y}}{\partial \bar{b}} = 1$$

$$\frac{\partial E}{\partial \bar{b}} = \frac{\partial E}{\partial \bar{y}} \cdot \frac{\partial \bar{y}}{\partial \bar{b}} = \frac{\partial E}{\partial \bar{y}} = \frac{1}{d} (\bar{y} - \hat{\bar{y}})$$

$$\frac{\partial E}{\partial \bar{m}_1} = \frac{\partial E}{\partial \bar{y}} \cdot \frac{\partial \bar{y}}{\partial \bar{m}_1} = \frac{\partial E}{\partial \bar{y}} = \frac{1}{d} (\bar{y} - \hat{\bar{y}})$$

- For the MM block:

$$\bar{m}_1 = \underline{x} \times \underline{W}$$

$$\frac{\partial \bar{m}_1}{\partial \underline{W}} = \underline{x}^T$$

$$\frac{\partial E}{\partial \underline{W}} = \underline{x}^T \times \frac{\partial E}{\partial \bar{m}_1} = \underline{x}^T \times \frac{1}{d} (\bar{y} - \hat{\bar{y}})$$

Considering the bias as a weight with costant 1 input, the derivative of the the error function w.r.t weights are:

$$\frac{\partial E}{\partial \bar{b}} = \frac{1}{d} (\bar{y} - \hat{\bar{y}}) \qquad \frac{\partial E}{\partial \underline{W}} = \underline{x}^T \times \frac{1}{d} (\bar{y} - \hat{\bar{y}})$$

5.

$$\underline{W}^{new} = \underline{W}^{old} - \eta \frac{\partial E}{\partial \underline{W}^{old}} \qquad \bar{b}^{new} = \bar{b}^{old} - \eta \frac{\partial E}{\partial \bar{b}^{old}}$$

where:

- $\underline{W}^{new}$ Updated weights matrix of the layer
- $\underline{W}^{old}$: weights matrix of the layer
- η: learning rate

6. I applied the answer 4 in order to get the derivative of the Error w.r.t the weights. Then i used the formula in aswer 5 to update the weights matrix.

$$\underline{W}^{new} = [0.1716, -0.026, 0.0288] \qquad \bar{b}^{new} = 1.9556$$

7. Basically the backpropagation is a learning algorithm based on gradient descend method. Our aim is to minimize the error function. We locally minimize the Error following the opposite direction of the gradient of the error function w.r.t the weights, this is the gradient descent. We apply gradient descend on every neuoron of every layer backpropagating the error throw the net with backporpagation alg.