Contents

List of Figures

1 Project Description

1.1 Basics

We use the following notation:

- Matrices are written in uppercase bold e.g. **X**.
- Vectors are written in lowercase bold e..g **x**.
- scalars are written in lowercase or uppercase. Lowercase indicates that it is a counting variable and uppercase that it is one of the limits in an finite set.
- For all functions , e.g. $f(x)$, where $x =$ **X** , we apply the function elementwise.
- Dot product is indicated by simple concatination of two matrices/vectors e.g. **Xy**.
- Element wise multiplication is indicated as $\mathbf{X} \otimes \mathbf{Z}$.

Firstly we initialize the weights in a matrix.

$$\mathbf{W}_{j\times k} = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1k} \\ w_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ w_{j1} & \dots & \dots & w_{jk} \end{bmatrix}, \mathbf{x}_{1\times k} = \begin{bmatrix} a_1 \\ \vdots \\ \vdots \\ a_k \end{bmatrix} \tag{1.1}$$

Forward propagation is one of the two passes the network needs to do (hence it's name). In forward propagation the network calculates the predicted output of the network.
Here we assume having

The output of the hidden layer j of the network can be calculated as the weighted sum of the inputs

$$\mathbf{nnet}_j = \sum_{k=1}^{n} w_{kj} x_k = \mathbf{Wx} \tag{1.2}$$

Even though this method should already work, we add a bias to every node to increase the learning speed and void that the network stops learning.

$$\mathbf{nnet}_j = \sum_{k=1}^{n} w_{kj} x_k + b_k = \mathbf{Wx} + \mathbf{b} \tag{1.3}$$

Later we see that back-propagation needs some variables, which can be already pre-computed during the forward step. These two variables are the output $\mathbf{o}_j$ of layer j and the derivative of the output w.r.t to the weighted sum $\mathbf{nnet}_j$. We represent the derivatives in a matrix $\mathbf{D}$ and the output of the current layer

$$\mathbf{D} = \frac{\mathbf{o}_j}{\mathbf{nnet}_j} \tag{1.4}$$

In case of sigmoid activation function, we obtain:

$$\mathbf{D} = \varphi\left(\mathbf{nnet}_j\right)\left(1 - \varphi\left(\mathbf{nnet}_j\right)\right) \tag{1.5}$$

$$\mathbf{o}_j = \varphi\left(\mathbf{Wx} + \mathbf{b}\right) \tag{1.6}$$