Exercise 3-1  A simple Neural Network

The illustration below depicts a two-layered neural network with inputs $x \in \mathbb{R}$ and for each input one bias $x_0 = z_0 = 1$ (i.e. $x_i = (1, x_{i,1})^T$) in the input as well as the hidden layer.

As function of the hidden neurons we employ a sigmoid, i.e.

$$z_h = \frac{1}{1 + \exp\left(-\sum_{j=0}^{M} v_{h,j} x_{i,j}\right)}$$

the output neuron $\hat{y}$ is, as usual, a linear combination.

(a) Prove that the following holds:

$$\frac{\partial z_h}{\partial v_{h,j}} = x_{i,j} \cdot z_h \cdot (1 - z_h)$$

Possible Solution:

$$\frac{\partial z_h}{\partial v_{h,j}} = \frac{\partial \phi(x, v_h)}{\partial v_{h,j}} = (-x_{i,j}) e\left(-\sum_{j=0}^{M} v_{h,j} x_{i,j}\right) \left(1 + \exp\left(-\sum_{j=0}^{M} v_{h,j} x_{i,j}\right)\right)^2 =$$

$$= x_{i,j} e\left(-\sum_{j=0}^{M} v_{h,j} x_{i,j}\right) \left(1 + \exp\left(-\sum_{j=0}^{M} v_{h,j} x_{i,j}\right)\right)^2 = x_{i,j} \left(1 + e\left(-\sum_{j=0}^{M} v_{h,j} x_{i,j}\right)\right)^2 =$$

$$= x_{i,j} \cdot \frac{1}{1 + e\left(-\sum_{j=0}^{M} v_{h,j} x_{i,j}\right)} \cdot \left(1 - \frac{1}{1 + e\left(-\sum_{j=0}^{M} v_{h,j} x_{i,j}\right)}\right) = x_{i,j} \cdot z_h \cdot (1 - z_h).$$
(b) Express the maximal value of \( \hat{y} \) subject to \( w \), if all original weights \( w_h \ (h \in \{0, \ldots, M_\phi\}) \) are positive. What’s the minimal value?

Possible Solution:

\[
\text{max } z_h = \max \frac{1}{1 + e^{-\sum_{j=0}^{M} v_{h,j}x_{i,j}}} = \max \frac{1}{1 + e^{-\sum_{j=0}^{M} v_{h,j}x_{i,j}}} = \min \left(1 + e^{-\sum_{j=0}^{M} v_{h,j}x_{i,j}} \right) = 1 + e^{-\max \sum_{j=0}^{M} v_{h,j}x_{i,j}}
\]

if \( \sum_{j=0}^{M} v_{h,j}x_{i,j} \to +\infty \) \( \approx \) maximal value of \( \hat{y} \)

if \( \sum_{j=0}^{M} v_{h,j}x_{i,j} \to -\infty \) \( \approx \) minimal value of \( \hat{y} \):

\[ \hat{y}_{\text{max}}: \text{ if } \sum_{j=0}^{M} v_{h,j}x_{i,j} \to +\infty \Rightarrow \text{max } z_h \to \frac{1}{1+\infty} = 1 \]
\[ \Rightarrow \text{max } \hat{y} = \max x \sum_{h=0}^{M_\phi-1} w_h z_h = \sum_{h=0}^{M_\phi-1} w_h. \]

\[ \hat{y}_{\text{min}}: \text{ for each } j \in \{1, \ldots, M\} \text{ it has to hold that: } v_{h,j}x_{i,j} \to -\infty \]
\[ \Rightarrow \text{min } z_h \to \frac{1}{1+\infty} = 0 \Rightarrow \text{the weighted sum over all } w_h(h \in \{1, \ldots, M_\phi\}) \text{ is also 0} \]
\[ \Rightarrow \text{min } \hat{y} = \min w_0 z_0 + \sum_{h=1}^{M_\phi-1} w_h z_h = w_0. \Rightarrow \text{Bias-Functionality} \]

(c) If \( v_{h,j} = 0 \) for all \( j \in \{0, \ldots, M\}, h \in \{1, \ldots, M_\phi\} \), then what is \( \hat{y} \)? Which functions describe \( \hat{y} \) if all \( v_{h,j} = c, c \neq 0 \)?

Possible Solution:

\[ \text{c) For } v_{h,j} = 0 \text{ the influence of } x \text{ is lost, } z_h = \text{sig}(0) = 0.5, \text{ und } \]
\[ \hat{y} = w_0 + 0.5 \cdot \sum_{h=1}^{M_\phi-1} w_h. \]
\[ \text{The result is therefore a constant value for every } x. \]

If \( v_{h,j} = c \Rightarrow z_h = \frac{1}{1+e^{(-c \sum_{j=0}^{M} x_{i,j})}} = \frac{1}{1+e^{-c(1+x_{i,1})}} \), for all \( h \).

Hence: \[ \hat{y} = w_0 + \sum_{h=1}^{M_\phi} w_h \frac{1}{1+e^{-c(1+x_{i,1})}} = w_0 + \left( \sum_{h=1}^{M_\phi} w_h \right) \cdot \frac{1}{1+e^{-c(1+x_{i,1})}} \]

Thus we get a sigmoidal. The weights \( w_h \) sum up to a scaling factor, \( w_0 \) corresponds to the offset (minimum) of the curve and \( c \) defines the gradient of the sigmoidal.
Exercise 3-2 Convolutional Neural Networks

In this exercise we address a convolutional neural network (CNN) with one-dimensional input. While two-dimensional CNNs can be used for example for grayscale images, one-dimensional CNNs could be used for time-series such as temperature or humidity readings. Concepts for the 1D-case are equivalent to 2D networks. We interpret data in our network as three-dimensional arrays where a row denotes a feature map, a column denotes a single dimension of the observation, and the depth of the array represents different observations. As we will only work with a single input vector, the depth will always be one.

Let the following CNN be given:

- Input $I$: Matrix of size $1 \times 12 \times 1$. We therefore have an input with twelve dimensions consisting of a single feature map.
- First convolutional layer with filters $F^1_0 = (-1, 0, 1)$ and $F^1_1 = (1, 0, -1)$ that generates two output feature maps from a single input feature map. Use valid mode for convolutions.
- Max-pooling layer with stride 2 and filter size 2. Note that max-pooling pools each feature map separately.
- Convolutional layer with convolutional kernel $F^2_0 = ((-1, 0, 1), (1, 0, -1))$ of size $2 \times 3 \times 1$.
- Fully connected layer that maps all inputs to two outputs. The first output is calculated as the negative sum of all its inputs, and the second layer is calculated as the positive sum of all its inputs.
- Sigmoidal activation function

Calculate the response of the CNN for the two inputs $(0, 0, 0, 0, 1, 1, 1, 0, 0, 0)$ and $(0, 0, 0, 0, 1, 1, 1, 0, 0, 0)$.
Exercise 3-3  ‘The Simpsons’ Characters Classification

The provided notebook on the website leads you through a classification task of the main characters of the TV-series ‘The Simpsons’. Therefore, we will use the ‘The Simpsons Characters Data’ from kaggle.com, provided by the user alexattia. The original source of the dataset can be found here: https://www.kaggle.com/alexattia/the-simpsons-characters-dataset

The dataset contains images of the characters. As the images are of different size, we will provide a slightly pre-processed data, where the images are already equally sized (64x64). The links are in the notebook. For the classification, we will implement a neural network with the high-level API of keras. Keras is a model-level library, providing high-level building blocks for developing deep learning models.