



University of Stuttgart



ANALYTIC
COMPUTING

Machine Learning (SS 23)

Assignment 10: Neural Networks

Mojtaba Nayyeri

Mojtaba.Nayyeri@ipvs.uni-stuttgart.de

Akram Sadat Hosseini

Akram.Hosseini@ipvs.uni-stuttgart.de

Nadeen Fathallah

Nadeen.Fathallah@ipvs.uni-stuttgart.de

Rodrigo Lopez Portillo Alcocer

rodrigo.lopez-portillo-alcocer@ipvs.uni-stuttgart.de

Tim Schneider

timphillip.schneider@ipvs.uni-stuttgart.de

Osama Mohammed

osama.mohammed@ipvs.uni-stuttgart.de

Daniel Frank

daniel.frank@ipvs.uni-stuttgart.de

Make sure to list the full names of all participants, matriculation number, study program, and B.Sc. or M.Sc on the first page. Optionally, you can *additionally* upload source files (e.g., PPTX files). If you have any questions, feel free to ask them in the exercise forum in ILIAS.

Submission is open until Monday, 10th of July 2023, 12:00 noon.



Formalizing Neural Networks

Create a Feed forward Neural Network with 3 hidden layers and an input dimension of 5. Each hidden layer uses the tanh activation function. Each layer includes a bias term. The FNN is a multi-class classifier for with 3 classes $\{-1, 0, 1\}$. Choose the appropriate output function. In your definition, provide the function for the forward-pass and the dimensions of each weight matrix and bias vector. Afterwards, define an appropriate loss function for your multi-class classification neural network. Assume that the loss function operates on batches of size 32. Then, extend the loss function to punish miss-classifications of class -1 more heavily than the other classes. Use a parameter $\alpha \geq 0$ to control this weighting. The extended loss function should be equivalent to the original loss function if $\alpha = 0$.



Neural Networks Example

Let us define a feedforward neural network for binary classification. The network takes a two-dimensional vector $x = [x_1 \ x_2]^\top$ as input and outputs a scalar $\hat{y} \in \mathbb{R}$.

We formally define this neural network as:

$$h = \text{ReLU}(W^h x + b^h)$$
$$\hat{y} = W^o h + b^o.$$

with the initialized parameters:

$W^h = \begin{bmatrix} w_{11}^h & w_{12}^h \\ w_{21}^h & w_{22}^h \\ w_{31}^h & w_{32}^h \end{bmatrix} = \begin{bmatrix} 0.2 & 0.4 \\ 0.8 & 0.1 \\ 2.0 & 0.5 \end{bmatrix} \in \mathbb{R}^{3 \times 2}$ is the hidden layer matrix, $b^h = [1.0 \ 1.0 \ -4.0]$ is the hidden layer bias, and $W^o = \begin{bmatrix} w_1^o & w_2^o & w_3^o \end{bmatrix} = \begin{bmatrix} 0.5 & -1.0 & 0.0 \end{bmatrix} \in \mathbb{R}^{1 \times 3}$ is the output weight with its corresponding bias $b^o = 3.0$.

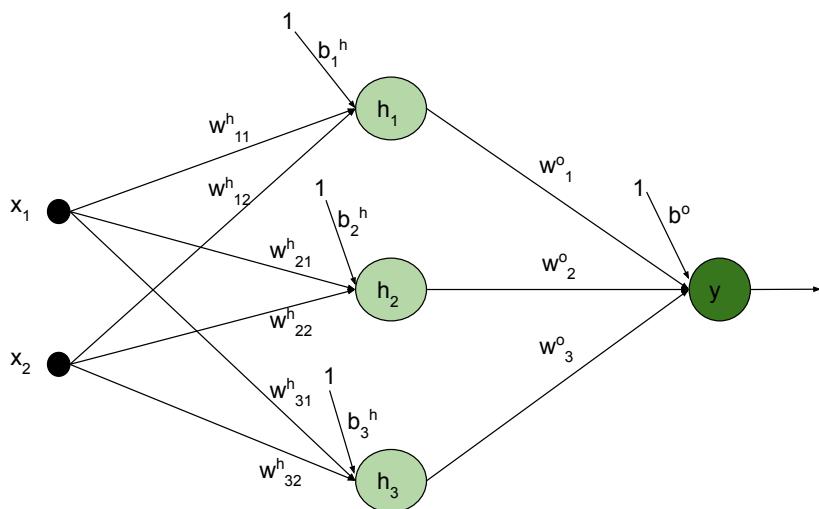


Figure 1 Neural Network

- 1) Compute the forward pass for $[1.0 \ 2.0]^\top$. Show each computation step.



2) What loss function would you use to train the given neural network for binary classification? Define the loss function formally. Compute the loss for the prediction made in the previous task given a true label of $y = -1$. Hint: Check the chapter "Feed-forward networks predicting a Bernoulli distribution" of the lecture slides on neural networks



Implementing a Feedforward Neural Network

Download the provided Jupyter notebook `Assignment10.ipynb` and the dataset `airfoil_self_noise.csv`. Follow the instructions in the Jupyter notebook.