



# Machine Learning (SS 23)

## Assignment 11: Neural Networks (Continue) and PyTorch

Mojtaba Nayyeri

[Mojtaba.Nayyeri@ipvs.uni-stuttgart.de](mailto:Mojtaba.Nayyeri@ipvs.uni-stuttgart.de)

Akram Sadat Hosseini

[Akram.Hosseini@ipvs.uni-stuttgart.de](mailto:Akram.Hosseini@ipvs.uni-stuttgart.de)

Nadeen Fathallah

[Nadeen.Fathallah@ipvs.uni-stuttgart.de](mailto:Nadeen.Fathallah@ipvs.uni-stuttgart.de)

Rodrigo Lopez Portillo Alcocer

[rodrigo.lopez-portillo-alcocer@ipvs.uni-stuttgart.de](mailto:rodrigo.lopez-portillo-alcocer@ipvs.uni-stuttgart.de)

Tim Schneider

[timphillip.schneider@ipvs.uni-stuttgart.de](mailto:timphillip.schneider@ipvs.uni-stuttgart.de)

Osama Mohammed

[osama.mohammed@ipvs.uni-stuttgart.de](mailto:osama.mohammed@ipvs.uni-stuttgart.de)

Daniel Frank

[daniel.frank@ipvs.uni-stuttgart.de](mailto: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, 17th of July 2023, 12:00 noon.**



## Note:

This assignment builds upon the previous Neural Networks assignment from last week, encompassing additional components such as backpropagation and the implementation of the training process for a neural network.



## Neural Networks Example

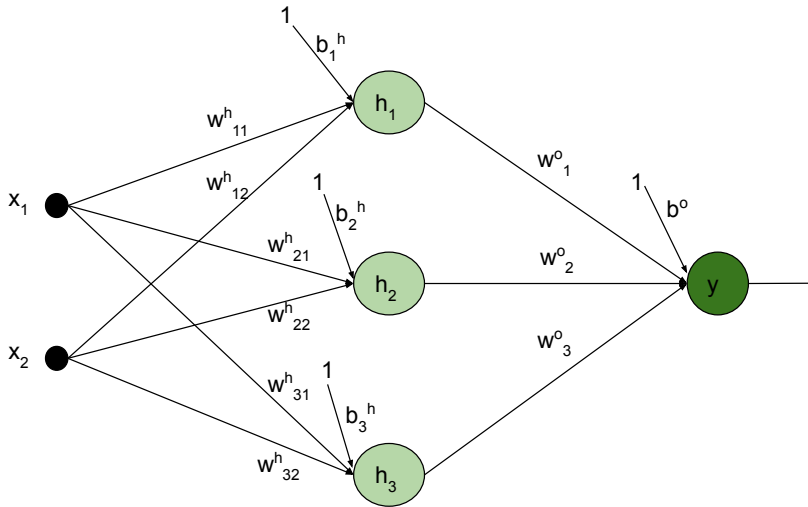
Let us define a feedforward neural network for binary classification. The network takes a two-dimensional vector  $x = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^\top$  as input and outputs a scalar  $\hat{y} \in \mathbb{R}$ .

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,  $b^h = \begin{bmatrix} 1.0 & 1.0 & -4.0 \end{bmatrix}$  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  $\begin{bmatrix} 1.0 & 2.0 \end{bmatrix}^\top$ . Show each computation step (From assignment 10).



- 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  $y = -1$  (From assignment 10). Hint: Check the chapter "Feed-forward networks predicting a Bernoulli distribution" of the lecture slides on neural networks.
- 3) Given the computed loss and a learning rate  $\eta = 0.1$ , adjust the weight  $W_{12}^h$  using backpropagation with gradient descent. Hint: Find an example step-by-step computation of backpropagation in footnote <sup>1</sup>.

---

<sup>1</sup><https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example>



## Implementing a Feedforward Neural Network

Download the provided Jupyter Notebook `Assignment11_PyTorch_P1.ipynb`, `Assignment11_PyTorch_P2.ipynb`, and the dataset `airfoil_self_noise.csv`. Follow the instructions in the Jupyter Notebook. (Forward pass was given in assignment 10, implementation of backward and training process are required).