OTH Amberg-Weiden
Fakultät für Elektrotechnik, Medien und Informatik
Prof. Dr.-Ing. Christian Bergler, Prof. Dr. Fabian Brunner

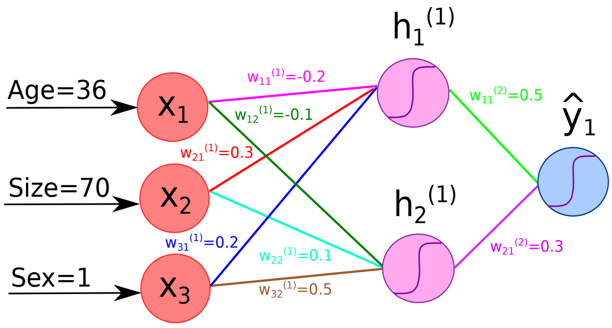
Deep Learning – Summer Semester 2024
Exercise for Lecture – Multi-Layer Perceptron

Task 1 (Multi-Layer Perceptron/Network)
Parameters, Shapes, and layers in a multi-layer Perceptron

a) Draw a multi-layer neural network consisting of three layers. The input layer consists of 4 nodes, the hidden layer of 5 nodes and the output layer of 3 nodes.

b) How many unknown parameters does the network have and what are the shapes of the individual vectors/matrices?

Task 2 (Multi-Layer Perceptron/Network)
The fully connected neural network (MLP) outlined below is used as a modelling approach for a regression task. The activation function is $\sigma(x) = x$ and it is assumed for simplicity that all bias values are equal to zero.



a) Outline the relationship described by the network between the output $\hat{y}$ and the inputs $\boldsymbol{x} = (x_1, x_2, x_3)^T$ in vector form by adding the entries of the weight matrices $W^{(1)}$ and $W^{(2)}$ to the following representation:

$$\hat{y} = f(\boldsymbol{x}) = W^{(2)}W^{(1)}\boldsymbol{x} = \begin{pmatrix} \qquad\qquad\qquad \end{pmatrix}\begin{pmatrix} \qquad\qquad\qquad\qquad\qquad \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}.$$

b) Describe the dimensionality (parametric complexity) of functions that can be represented by this model. Draw an equivalent MLP with as few nodes as possible that can represent the same functions.

Now let us also have an annotated training data set of length m consisting of samples $(\boldsymbol{x}^{(i)}, y^{(i)})$, $i \in \{1, \ldots, m\}$, $\boldsymbol{x}^{(i)} \in \mathbb{R}^3, y^{(i)} \in \mathbb{R}$. To determine the weights, the loss function

$$L = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y}^{(i)} - y^{(i)})^2$$

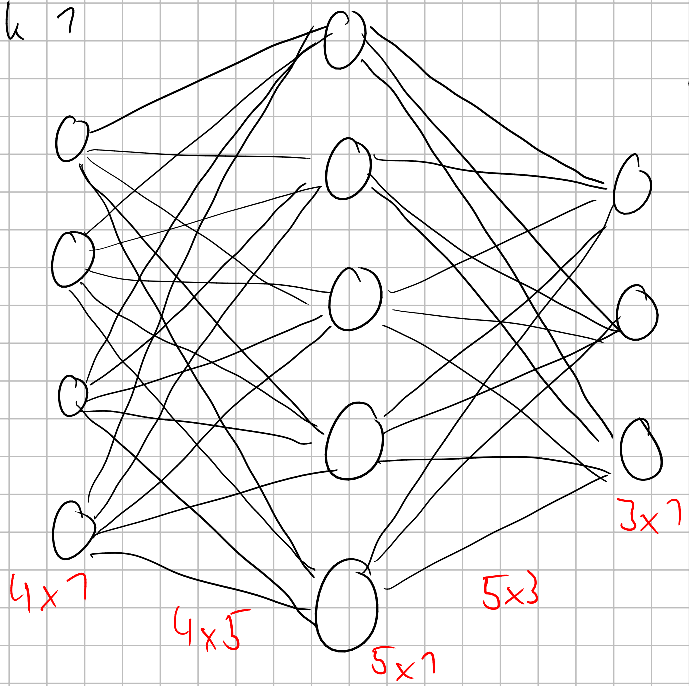
where $\hat{y}^{(i)} = f(\boldsymbol{x}^{(i)})$ denotes the model output for the input $\boldsymbol{x}^{(i)} = (x_1^{(i)}, x_2^{(i)}, x_3^{(i)})^T$.

c) Explain the task and the principle of the backpropagation algorithm in model training.

d) Calculate the partial derivatives of L according to the weights $w_{11}^{[1]}$ and $w_{21}^{[2]}$.

Task 1

a)



b) Anzahl Parameter

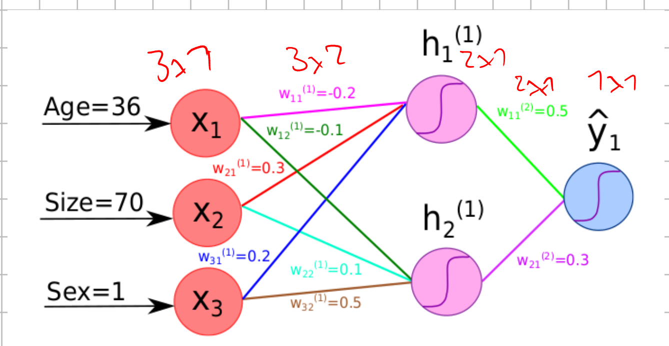
$$w^{(1)} \in \mathbb{R}^{4\times5} + b^{(1)} \in \mathbb{R}^{5} + w^{(2)} \in \mathbb{R}^{5\times3}$$

$$+ b^{(2)} \in \mathbb{R}^{3}$$

$$\Rightarrow 4\times5 + 5 \quad 5\times3 + 3$$

$$= 43$$

Task 2



b)

a)
$$\hat{y} = \begin{pmatrix} h_1^{(1)} \cdot w_{11}^{(2)} + b \\ h_2^{(1)} \cdot w_{21}^{(2)} + b \end{pmatrix} \cdot \begin{pmatrix} x_1 \cdot w_{11}^{(1)} + x_2 \cdot w_{21}^{(1)} + x_3 \cdot w_{31}^{(1)} + b \\ x_1 \cdot w_{12}^{(1)} + x_2 \cdot w_{22}^{(1)} + x_3 \cdot w_{32}^{(1)} + b \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

c) predict target values y given input features x
the model minimizes the difference between predicted values and target values

1. Forward pass: pass input data through the network, layer by layer
2. Calculate loss using L
3. Backward Pass: calculate gradient of loss w.r.t model parameter
4. Gradient Descent: update model parameters using gradients
5. repeat

d)
$$\frac{\partial L}{\partial w_{11}} = \frac{1}{m} \sum_{i=1}^{m} \left(f_\theta(\vec{x}^{(i)}) - y^{(i)} \right) \cdot \frac{\partial f_\theta(x^{(i)})}{\partial w_{11}}$$

$$\frac{\partial L}{\partial w_{21}} = \frac{1}{m} \sum_{i=1}^{m} \left(f_\theta(\vec{x}^{(i)}) - y^{(i)} \right) \cdot \frac{\partial f_\theta(x^{(i)})}{\partial w_{21}}$$