Exercise: 6

Meeting on 9th/ 11th February

Aufgabe 1: Properties of the softmax function

The softmax function

$$\text{softmax}(\vec{x})_i = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{1}$$

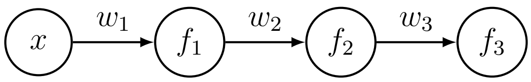
is often used as the final actication function in a multi-class classification problem. Show/explain for the following properties of the softmax function:

(a) $\text{softmax}(\vec{x})_i$ corresponds to a probability distribution. Hence, show that $\sum_i \text{softmax}(\vec{x})_i = 1$.

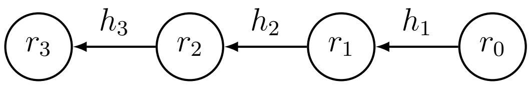
(b) For i it is for all $j \neq i$: $x_i \gg x_j$, then $\text{softmax}(\vec{x})_i \to 1$ and $\text{softmax}(\vec{x})_j \to 0$. Hence, why is the function called *soft*max?

Aufgabe 2: Backpropagation in a Deep Network - analytically

Consider a one-dimensional neural network with $\tanh(x)$ as activation function. Forward-Pass and Backpropagation can be represented by:



Forward-Pass



Backpropagation

The loss function as well as forward and backward variables are computed as

$$\begin{aligned} L(f_3) &= \frac{1}{2}(y - f_3)^2 & r_0 &= h_0 \\ f_3(f_2) &= b_3 + w_3 f_2 & r_1 &= h_1 r_0 \\ f_2(f_1) &= \tanh(b_2 + w_2 f_1) & r_2 &= h_2 r_1 \\ f_1(x) &= \tanh(b_1 + w_1 x) & r_3 &= h_3 r_2 \end{aligned}$$

with

$$\begin{aligned} h_0 &= \frac{\partial L(f_3)}{\partial f_3} \\ h_1 &= \frac{\partial f_3(f_2)}{\partial f_2} \\ h_2 &= \frac{\partial f_2(f_1)}{\partial f_1} \\ h_3 &= \frac{\partial f_1(x)}{\partial x} \end{aligned}$$

Compute the derivatives in the variables h_i explicitly. You will need the following derivatives for backpropagation:

$$\begin{aligned}
\frac{\partial L}{\partial w_3} &= \frac{\partial L}{\partial f_3}\frac{\partial f_3}{\partial w_3} \\
\frac{\partial L}{\partial b_3} &= \frac{\partial L}{\partial f_3}\frac{\partial f_3}{\partial b_3} \\
\frac{\partial L}{\partial w_2} &= \frac{\partial L}{\partial f_3}\frac{\partial f_3}{\partial f_2}\frac{\partial f_2}{\partial w_2} \\
\frac{\partial L}{\partial b_2} &= \frac{\partial L}{\partial f_3}\frac{\partial f_3}{\partial f_2}\frac{\partial f_2}{\partial b_2} \\
\frac{\partial L}{\partial w_1} &= \frac{\partial L}{\partial f_3}\frac{\partial f_3}{\partial f_2}\frac{\partial f_2}{\partial f_1}\frac{\partial f_1}{\partial w_1} \\
\frac{\partial L}{\partial b_1} &= \frac{\partial L}{\partial f_3}\frac{\partial f_3}{\partial f_2}\frac{\partial f_2}{\partial f_1}\frac{\partial f_1}{\partial b_1}
\end{aligned}$$

Rewrite these equations using the forward variables f_i and the backward variables r_i.

Aufgabe 3: Convolution and Pooling operations in a CNN

Consider the following three matrices:

- $A = \begin{pmatrix} 4 & -4 & 1 & 3 \\ -5 & 1 & -1 & 0 \\ 6 & 4 & 7 & 4 \\ 18 & 3 & -5 & 11 \end{pmatrix}$
- $B = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 8 & 7 & 6 & 5 \\ 4 & 3 & 2 & 1 \end{pmatrix}$

As well as the filter matrix W:

$$W = \begin{pmatrix} -3 & 3 \\ -1 & 2 \end{pmatrix}$$

Using W perform the convolution operations on A and B, once without padding (= "valid") and once with padding while keeping the original resolution (= "same").

Aufgabe 4: Programming a simple Neural Network

The lecture showed how you can program and train a simple neural network without any major frameworks. In this exercise you should try this yourself.

(a) Write a class "neuralNetwork" which initializes all necessary parameters, like input and output nodes as well as weights, and possesses the methods "train" and "query". Train and evalute the network on the MNIST[1] dataset.

(b) Try using your own or downloaded images as inputs.

Aufgabe 5: Q-Learning

Consider the *Markov Decision Problem* with three states $Z \in \{1, 2, 3\}$ and their rewards B with $B_1 = -1$, $B_2 = -2$ and $B_3 = 0$. State $Z = 3$ is a terminal state. In the states $Z = 1$ and $Z = 2$ there are two possible actions a and b. The indeterministic transition model is shown in Fig. 1.

[1] https://pjreddie.com/projects/mnist-in-csv/

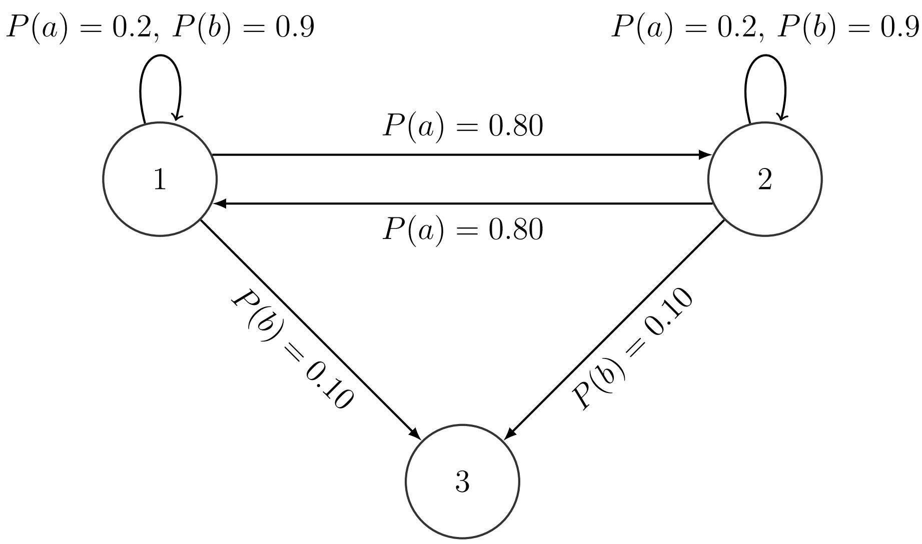


Figure 1: The indeterministic transition model

In this scenario the agent does not know the transition model nor the rewards. Instead he know only about the number of states and the available actions.

(a) Discuss, which action would be the most sensible one (as an external observer with access to all information) to end with the highest possible reward (or the lowest penalty) in the terminal state.

(b) Explain the components of the formula $Q(s, a)$ for the Q-learning algorithm.

(c) What is the meaning if the discount factor γ?

(d) Provide an outline of the sequence of the Q-learning algorithm.

(e) Program the Q-learning algorithm for $\alpha = 0.001$ and show the evolution of the (normalized) Q-matrix across 1000 iterations.