Assignment 1:
Basics and Word Embeddings

- This assignment is due on **May 9th, 2022 (23:59, CET)**
- You should form groups of two for the assignments and hand in one solution per group.
- You can discuss the problems with other groups or browse the Internet to get help. However, copy and paste is cheating.
- There will be no direct grading of assignments, but to be elegible for the exam, you need to sucessfully solve the assignments, which will be checked by us.
- The points associated with the tasks are there only for reference. This way you get an idea of how detailed the answers should be.
- There are 3 assignments in total.
- Submit at https://elearn.informatik.uni-kiel.de/course/view.php?id=51
  - only pdf files and only one file per group per assignment (lastName-lastName-assignment1.pdf)

Task 1: Linear Algebra

Given the following matrix $\boldsymbol{X} = \begin{bmatrix} 2 & 4 \\ 1 & 3 \end{bmatrix}$ and the vectors $\boldsymbol{y} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$ and $\boldsymbol{z} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$

a) What is the dot product of $\boldsymbol{y}$ and $\boldsymbol{z}$, often also written as $\boldsymbol{y}^T\boldsymbol{z}$? **1 P**

b) What is the product $\boldsymbol{X}\boldsymbol{y}$? **1 P**

c) What is $\boldsymbol{X}^2$? **1 P**

d) Is $\boldsymbol{X}$ invertible? If yes, what is the inverse $\boldsymbol{X}^{-1}$; if no, why not? **2 P**

e) What is the rank of $\boldsymbol{X}$? **1 P**

Task 2: Derivatives

What is the derivative of y with respect to x, if

a) $y = x^3 + x - 5$ **1 P**

b) $y = y = (5x^3 - 2x)(2x)$ **1 P**

c) $y = \frac{2x^2+3}{8x+1}$ **2 P**

d) $y = (3x - 2)^8$ **2 P**

e) $y = log(x^2 + x)$ **2 P**

Task 3: Cross Entropy Gradient

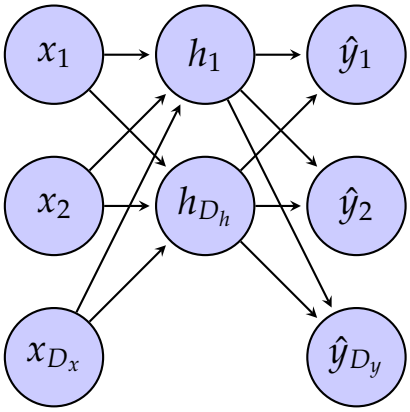
Derive the gradient of the cross entropy function with respect to the input of the softmax function. **7 P**

- In practice, the softmax output function is usually combined with cross entropy loss.
- The cross entropy (CE) is calculated for the general, D-dimensional case as follows: $CE(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\sum_D y_i log(\hat{y}_i)$
- Output: $\hat{\boldsymbol{y}} = softmax(\boldsymbol{\theta})$
- We are looking for:

$$\frac{\partial CE(\boldsymbol{y}, \hat{\boldsymbol{y}})}{\partial \boldsymbol{\theta}} \tag{1}$$

Task 4: Neural Network Gradients

a) Derive the gradient of cross entropy (CE) with respect to the input of the following neural network. **12 P**



- That is, we are looking for $\frac{\partial J}{\partial \boldsymbol{x}}$ with $J = CE(\boldsymbol{y}, \hat{\boldsymbol{y}})$
- $\boldsymbol{h} = sigmoid(\boldsymbol{x}\boldsymbol{W_1} + \boldsymbol{b_1})$
- $\hat{\boldsymbol{y}} = softmax(\boldsymbol{h}\boldsymbol{W_2} + \boldsymbol{b_2})$
- $\boldsymbol{y}$ is one-hot encoded; $\boldsymbol{W_i}$ are the respective weights and $\boldsymbol{b_i}$ are the biasterms

b) How many parameters does this neural network have? **3 P**

- Assumptions: dimension of $\boldsymbol{x} = D_x$; dimension of $\hat{\boldsymbol{y}} = D_y$; and D_h hidden units.

Task 5: Implementing Backpropagation

Implement a neural network and the backparopagation algorithm for the iris dataset. You can load it in python, e.g., through *from sklearn.datasets import load_iris*. Explore the data to understand the features and the classes, e.g., by looking at the data distributions.

Don't use any libraries for neural networks (scikit.learn, keras, etc.) but only pyhon, numpy, pandas, etc. You can hard-code the network architecture:

- 4 input nodes
- 1 hidden layer with 8 nodes and relu activation function
- 3 output nodes with softmax activation function
- cross entropy as loss function

Split the data into 80% training and 20% test with balanced class distributions Initialize your weights randomly and then train your network for 100 epochs.

Try this without help from the Web first. If you get stuck, you can consult the Web: there are many tutorials on using the iris dataset, e.g., https://janakiev.com/blog/keras-iris/ and on implementing backprop, e.g., https://machinelearningmastery.com/implement-backpropagation-algorithm-scratch-python/. **30 P**

a) Output the accuracy of your trained model on the training data.

b) Output the accuracy of your trained model on the test data.

Task 6: Implementing word2vec

Implement your own word2vec method to create word embeddings. For training, use the american standard version (asv) bible. You can find it, e.g., here: https://www.kaggle.com/datasets/oswinrh/bible. You can follow the example at https://www.tensorflow.org/tutorials/text/word2vec. They implement the skip-gram version of word2vec. Optionally, you could also implement the CBOW version. In any case, store your learned embeddings in a format to be able to load it into http://projector.tensorflow.org/ for visualization.
30 P

a) Output the 10 nearest neighbors for the term
   1) holy
   2) woman
   3) light

b) Output the top-3 nearest neighbosr for the algebraic expression
   1) jesus - man + woman = ?
   2) money - evil + good = ?
   3) find an interesting expression on your own