Assignment 1:
Basic of Neural Network and Text Classification

- This assignment is due on **May 10th, 2024 (23:59, CET)**
- You can discuss the problems with your classmates or browse the Internet to get help. However, copy and paste is cheating.
- You need at least 50% of the points of every assignment to take part in the exam.
- There are 3 assignments in total.
- Submit at https://elearn.informatik.uni-kiel.de/course/view.php?id=274
  - only pdf and jupyter files and only one zip file per assignment.
  - put your names and matriculation numbers on *each* page in the pdf file

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

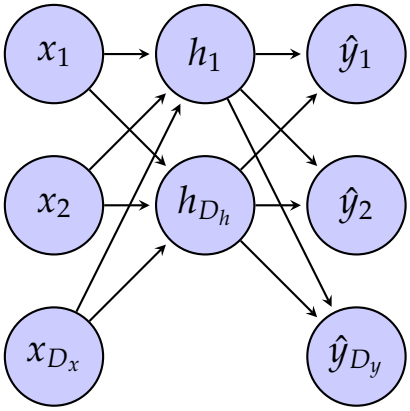
Derive the gradient of the cross entropy function with respect to the input of the softmax function. **7 P**

- 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. **10 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: Text classification

Implement a feedforward neural network for the 20 Newsgroups dataset. You can import the dataset in Python using fetch_20newsgroups from sklearn.datasets, or alternatively, download it from this link: 20 Newsgroups Dataset.

Feel free to utilize any libraries such as Scikit-learn, Keras, PyTorch, etc.

a) Exploratory Data Analysis **4 P**

- Calculate and visualize the class distribution
- Calculate and visualize the distribution of unigrams and bigrams (top 20 most frequently occurring terms after removing stop words).
- Calculate and visualize the number of words in each sample.
- Plot wordclouds for two classes: one with the highest number of samples in the dataset and another with the lowest.

b) Data Pre-processing **3 P**

- Remove stop words.
- Apply either lemmatization or stemming based on your analysis (provide reasoning for your choice).
- Remove punctuation, non-English characters, special characters, and URLs (if present).

c) Data Split **1 P**

- Divide the dataset into training and testing sets, with 20% allocated for training and 80% for testing

d) Word Embedding **5 P**

- Divide the dataset into training and testing sets, with 20% allocated for training and 80% for testing
- Utilize gloVe embedding for text encoding. Explore different dimensions such as 50, 100, or 300, depending on what you believe will yield the best results.
- Optional, fine-tune the embedding on this dataset if you think it will increased model performance.

e) Model Designing and Tainning **6 P**

- Implement a feedforward neural network. You have the freedom to design the architecture, including the number of neurons per layer, the number of hidden layers, choice of activation function, loss function, optimizer, etc.
- Experiment with different architectures, activation functions, and optimization algorithms to improve the model's performance.

f) Model Evaluation **2 P**

- Evaluate the model by computing F1 scores (Micro and Macro) and accuracy scores.

g) TensorBoard **3 P**

- Implement TensorBoard to visualize the plots of losses, F1 scores, and accuracy scores on both training and test data.

Task 6: Regression

Implement a feedforward neural network for Vehicle dataset. You can download the data from this link: Vehicle dataset.

Feel free to utilize any libraries such as Scikit-learn, Keras, PyTorch, etc.

a) Data Pre-processing **3 P**

- Clean the dataset by handling missing values, outliers, and any inconsistencies in the data (if its present).
- Normalize or standardize the numerical features to bring them to a similar scale.
- Encode categorical variables (if any) using techniques like one-hot encoding.

b) Data Split **1 P**

- Divide the dataset into training and testing sets, with 20% allocated for training and 80% for testing

c) Model Designing and Training **6 P**

- Implement a feedforward neural network. You have the freedom to design the architecture, including the number of neurons per layer, the number of hidden layers, choice of activation function, loss function, optimizer, etc.
- Tune the hyperparameters such as learning rate, batch size, and number of epochs to achieve optimal performance.

d) Model Evaluation **2 P**

- Calculate metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) to assess the model's performance.

e) Fine-tuning and Optimization **4 P**

- Experiment with different architectures, activation functions, and optimization algorithms to improve the model's performance.

- Perform techniques like regularization (e.g., L2 regularization) and dropout to prevent overfitting.