Introduction to Deep Learning in Speech and Text Processing

Written exam - Mock exam WS 2025/2026

Instructions:

- Please do not start the exam until you are asked to do so.
- You have 90 minutes to complete this exam.
- It is not necessary to be too wordy; it is advisable to keep your responses short.
- Please fill out your full name, matriculation number and study program below.
- Only pen and paper are allowed.
- Please write your solutions in the provided boxes; if you need additional paper, then ask an exam supervisor.

Full name: ______________________________

Matriculation number: ______________________________

Study Program: ______________________________

Good Luck!

This part will be filled in by the supervisors.

Question:	1	2	3	4	5	6	Total
Points:	11	19	11	14	10	12	77
Score:							

Exercise 1: Basics in Machine Learning [/11 points]

(a) [2 points] What is the main difference between a regression and a classification task? Name one regression task and one classification task in speech or natural language processing.

Answer

(b) [4 points] Calculate the mean squared error for the following linear regression model and the following test dataset.

Model:

$$w = \begin{bmatrix} -1 \\ 0 \\ 0.5 \\ 2 \end{bmatrix}, b = -1$$

Dataset:

$$(x_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 2 \end{bmatrix}, y_1 = 1), (x_2 = \begin{bmatrix} -1 \\ 1 \\ 0 \\ -2 \end{bmatrix}, y_2 = -3), (x_3 = \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix}, y_3 = 0.5)$$

Answer

(c) [5 points] Mark whether the following statements are True or False.

Statement	True	False
The goal of preventing overfitting is to better generalize to unseen data.		
Regularization typically increases the error on the development set.		
Support vector machine can only be used for binary classification.		
K-means is a partitional clustering method.		
Hyperparameters are tuned using the test set.		

Exercise 2: Neural Networks in General [/19 points]

(a) [4 points] Predict the class for the input "this exam is fair": 1) Compute the bag-of-words vector x for the input using the vocabulary V

$$V = \{v_0 : \text{exam}, v_1 : \text{test}, v_2 : \text{fair}\}$$

and 2) then compute the output of the neural network which predicts

$$y = \begin{bmatrix} \text{positive} \\ \text{negative} \end{bmatrix}.$$

The weights are W^1 and W^2, b^2 for the first and the second layer, respectively. There is no bias in the first layer. The first layer uses ReLU activation, while the output uses softmax activation. 3) Finally, compute the cross-entropy loss assuming "positive" as correct label. It is sufficient to provide the formulas, you do not need to compute the final values.

$$W^1 = \begin{bmatrix} 1 & 1 & 0 \\ -1 & 0 & 2 \end{bmatrix}, \; W^2 = \begin{bmatrix} 1 & 1 \\ -1 & 0 \end{bmatrix}, \; b^2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Answer

(b) [5 points] Perform gradient computation for the second layer of the previous task's neural network and sample. For this, compute and provide δ^2 and the derivative of the loss c with respect to W^2 and b^2, i.e.,

$$\nabla_{W^2} c = \begin{bmatrix} \frac{\partial c}{\partial w_{11}^2} & \frac{\partial c}{\partial w_{12}^2} \\ \frac{\partial c}{\partial w_{21}^2} & \frac{\partial c}{\partial w_{22}^2} \end{bmatrix}, \; \nabla_{b^2} c = \begin{bmatrix} \frac{\partial c}{\partial b_1^2} \\ \frac{\partial c}{\partial b_2^2} \end{bmatrix}$$

Assume predicted output $y = \begin{bmatrix} 0.88 \\ 0.12 \end{bmatrix}$.

Answer

(c) [5 points] Explain what happens in line 12w of the following PyTorch-like code snippet. Furthermore, two important operations regarding the optimizer are missing. Write those two lines of code, where to insert them, and explain what their purpose is.

Note: Assume that all symbols including `dataloader`*,* `model` *and* `loss_function` *have been defined.*

```
 1 dataloader = ...
 2 model = ...
 3 loss_function = ...
 4 optimizer = torch.optim.AdaGrad(model.parameters(), lr
      =0.01)
 5 for batch in dataloader:
 6     # prepare data and labels
 7     inputs, labels = batch["inputs"], batch["labels"]
 8     # compute the outputs of the model
 9     predictions = model(inputs)
10     # calculate the loss w.r.t. the labels and outputs
11     loss = loss_function(predictions, labels)
12     loss.backward()
```

Answer

(d) [5 points] Mark whether the following statements are True or False.

Statement	**True**	**False**
Each layer in a NN consists of a linear transformation and a non-linear activation.		
Cross-entropy loss is often used for multi-class, multi-label classification.		
For the backward-pass, we always first need to compute δ^l for the first layer.		
Neural networks typically consist of multiple layers.		
$\mathrm{ReLU}(z) = \min(0, z)$		

Exercise 3: Recurrent Neural Networks [/11 points]

(a) [2 points] What is the difference of RNNs to common (feed-forward) neural networks? What type of inputs do RNNs typically handle?

Answer

(b) [2 points] How does an Elman RNN compute the hidden state at timestep t? Provide the formula and define the variables.

Answer

(c) [3 points] Using inputs $x_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $x_2 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$ for timesteps 1 and 2, respectively, compute the output of the second timestep of the following Elman RNN consisting of one hidden layer with ReLU activation and one output layer without activation (there are no biases):

$$W_i = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \; W_h = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \; W_o = \begin{bmatrix} -1 & 1 \end{bmatrix}, \; a_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Answer

(d) [4 points] Mark whether the following statements are True or False.

Statement	True	False
RNNs are trained using backpropagation through time.		
LSTMs suffer from vanishing gradients.		
The last hidden state of an RNN can contain information from the whole input sequence.		
When employing RNNs, we always need to pad the inputs.		

Exercise 4: Convolutional Neural Networks [/14 points]

(a) [2 points] What is the general motivation for CNNs? Name two reasons.

Answer

(b) [2 points] Give one example for a speech task and one for a text task of how the inputs to a CNN can be represented.

Answer

(c) [1 point] Explain the idea of applying CNNs to embeddings in the context of NLP tasks.

Answer

(d) [4 points] Compute the output of a convolutional layer as well as the output of the subsequent max-pooling layer with the input matrix x ($x^l = x$):

$$x = \begin{bmatrix} 0 & -1 & -1 & 0 \\ 1 & 0 & 2 & -1 \\ 2 & 1 & -2 & 2 \end{bmatrix}$$

$$x^{l+1} = ReLU(f^W(x^l)), \text{ where } W = \begin{bmatrix} -1 & 1 \\ 2 & 0 \end{bmatrix} \text{(stride = (1, 2))}$$

The max-pooling layer has a pooling window of size (1, 2) with a stride of (1,2). Strides and sizes are defined as (row, column). There is no padding.

Answer

(e) [1 point] How do we compute the derivative of a max pooling layer?

Answer

(f) [4 points] Mark whether the following statements are True or False.

Statement	**True**	**False**
The number of filters is a hyper-parameter.		
The size of the filters are parameters.		
CNNs are a special case of feed-forward neural networks.		
The filter weights are parameters.		

Exercise 5: Sequence Modeling [/10 points]

(a) [2 points] Name a property of the tasks on which sequence-to-sequence models can be applied. Also name a task for which sequence-to-sequence models can be applied.

Answer

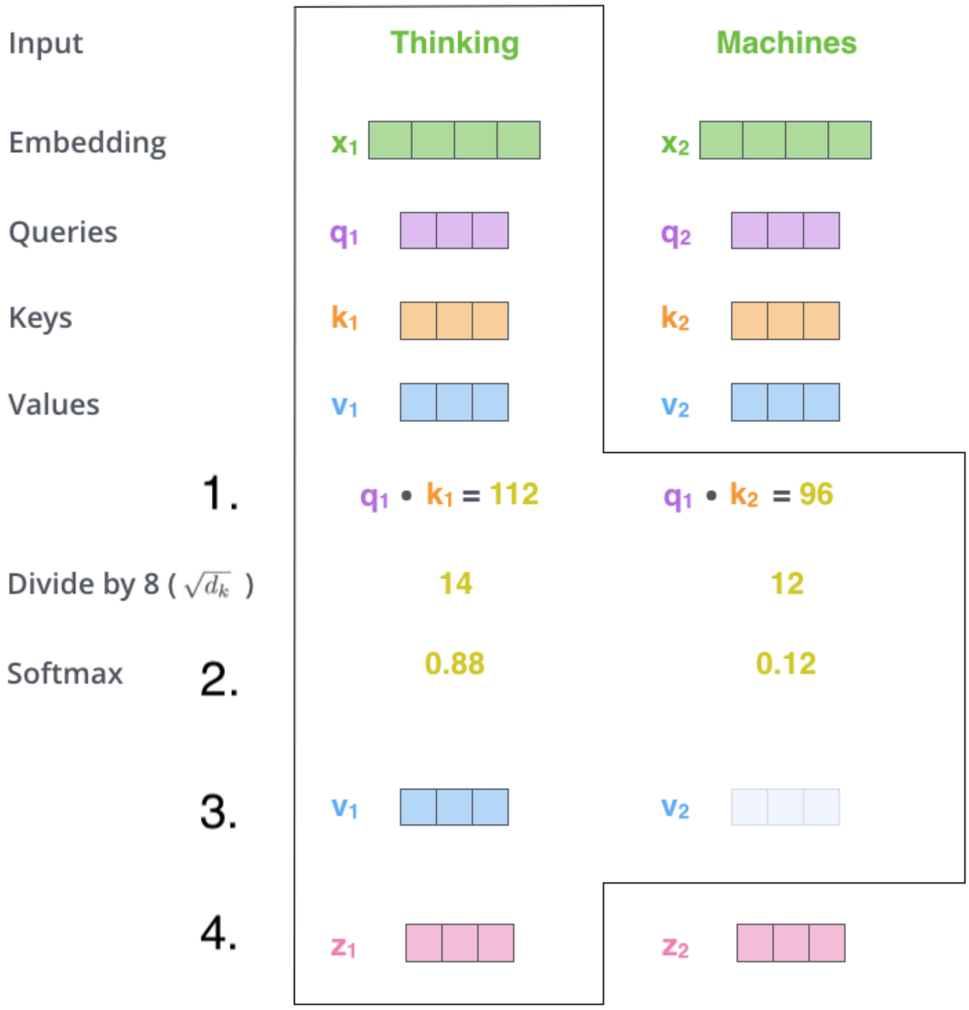
(b) [2 points] For two attention scoring methods, explain how the hidden states of an encoder model are used by the decoder model. Both, the encoder and decoder are recurrent neural networks.

Answer

(c) [2 points] How are the queries, keys and values computed in self-attention? What are the learnable weights in this operation?

Answer

(d) [4 points] In the following figure on self-attention operations[1], shortly describe steps 1, 3 and 4. What is the purpose of step 2?



Answer

[1] Source: `https://jalammar.github.io/illustrated-transformer/`

Exercise 6: Tricks [/12 points]

(a) [1 point] Explain the main difference between Stochastic Gradient Descent and Mini-Batch Gradient Descent.

Answer

(b) [3 points] Why does parameter initialization matter? Name and describe two methods.

Answer

(c) [2 points] What is weight decay and what is its purpose?

Answer

(d) [2 points] How do the outputs of neurons using dropout have to be scaled during testing if we do not want to scale during training?

Answer

(e) [4 points] Consider the following simplified PyTorch-like code snippet for training a model with Early Stopping. There are **four conceptual mistakes** regarding how early stopping is applied. Find and correct them. NOTE: Conceptual errors are logical errors of the algorithm, they are NOT syntax mistakes, typos, or runtime bugs that cause the program to crash.

```
model = ... # initialize model
optimizer = ... # define optimizer
train_loader, val_loader, test_loader = ...

best_loss = float('inf')
patience = 5
no_improvement = 0
best_model = None

for epoch in range(epochs):
    train_loss = train_epoch(model, optimizer, train_loader)
    val_loss = eval_epoch(model, test_loader)

    if train_loss < best_loss:
        best_loss = train_loss
        best_model = model.state_dict()
    else:
        no_improvement += 1

    if no_improvement >= patience:
        break

save_model(model, 'final.pt')
```

Answer