Project work part II: Questions on Deep Learning

Prof. Dr.-Ing. Stache

Fill out the following:

Name: Dominik Bücher,

Matrikel-Nr. / Student-ID: 216825

I confirm that the work I handed in is my own work. All sources are cited.
I know that my work will be checked for plagiarism, and I accept that any occurrence of plagiarism and/or non-cited sources will lead to a grade of 5.0 for this part. The due date for this work and the upload-link are present in ILIAS, uploads after the due are not accepted.

Questions:

Please answer the questions briefly but very precisely. You will not receive the point if the answer is not precise and clear. Nor do you get the point(s) if your answer itself is correct but contains additional statements that are wrong.

1. Topic "Architecture & Training"
   1.1. Why could it be reasonable to use only a linear activation function in the output layer instead of a sigmoid? (1 P)
   1.2. For what kind of data do we typically use "cross entropy loss" as loss function? (1 P)
   1.3. What is meant by the principle of weight sharing in the context of convolutional neural networks? (1 P)

2. Topic: "Language processing"
   2.1. How is an embedding determined? (1 P)
   2.2. What does the parameter "size" of an embedding mean and to which parameter does it correspond in a neural network? (1 P)
   2.3. What is the main difference between CBOW and Skip-gram? (1 P)

3. Topic: "Reinforcement Learning"
   3.1. What's the issue with training an agent to maximize the expected immediate reward? (1 P)
   3.2. Is the value loss function for Deep Q-Learning usually monotonically decreasing? Explain why / why not. (1 P)
   3.3. What are the differences and similarities between (tabular) Q-Learning and Dynamic Programming? (2 P)

4. Topic: “Transformers”
   4.1. How does the transformer get information of the order of words and how is this implemented? (2 P)
   4.2. How is an attention filter created in the transformer and how is cosine similarity taken into account? (2 P)
   4.3. Why does the decoder structure of the transformer use a masking in the attention Mechanism? (1 P)

Answers:

1. Topic “Architecture & Training”

1.1 Why could it be reasonable to use only a linear activation function in the output layer instead of a sigmoid?

Using only a linear activation function in the output layer instead of a sigmoid can be a reasonable choice for several reasons. Firstly, in the sigmoid activation function the neurons never truly reach the output values of 1 to 0 that’s because of saturation. When using a linear activation function instead, it helps to reduce the saturation.

The next reason is, that its very efficient to implement, that’s because of the threshold matrix is 0.

The third reason is to reduce vanishing gradient issues which enables deep network structures.

The last reason is that the sigmoid function is learning slow when there are large weights, and the gradients are small. This is not the case for linear activation functions.

(Stache P. D.-I., 5. Deep Neural Networks, pp. 15-16)

1.2 For what kind of data do we typically use “cross entropy loss” as loss function?

The cross entropy los function is used when there is a classification problem with multiple classes. It is important that the total probability added together is 1. That is, when the classes are mutually exclusive.

(Stache P. D.-I., 5. Deep Neural Networks, pp. 23-29)

1.3 What is meant by the principle of weight sharing in the context of convolutional neural networks?

Weight sharing is used to find local features across an image. For this, the same weights are used in all input samples. Weight sharing is used when, for example, the same object appears in different images but is positioned in different places.

(Stache P. D.-I., 6. Convolutional Neural Networks, S. 7)

2. Topic: “Language processing”

2.1 How is an embedding determined?

An embedding consists of words which are defined as vectors and assigned a certain length. The vectors represent the properties of the words. The aim is to obtain a relationship between the different words through the defined vectors.

(Stache P. D.-I., 9. Embeddings, Word2Vec, S. 2-7)

2.2 What does the parameter “size” of an embedding mean and to which parameter does it correspond in a neural network?

The parameter "size" determines the number of dimensions used to represent the individual word vectors in the embedding. Where the "size" in relation to a neural network indicates the number of neurons that are used.

(Stache P. D.-I., 9. Embeddings, Word2Vec, S. 8-14)

2.3 What is the main difference between CBOW and Skip-gram?

The main difference between CBOW and Skip-gram is that CBOW has as its main objective the prediction of the central word based on the surrounding context words, while Skip-gram focuses on predicting the context words based on the central word.

3. Topic: “Reinforcement Learning”

3.1 What’s the issue with training an agent to maximize the expected immediate reward?

The noisy TV problem illustrates the limitations of training agents based entirely on maximising immediate rewards. In this scenario, the agent learns to exploit a loophole by changing the channel to silence the TV completely instead of finding more appropriate solutions to reduce the noise. This highlights unintended side effects and reward hacking, where the agent favours short-term rewards over the intended goal.

(Stache P. D.-I., S. 35-37)

3.2 Is the value loss function for Deep Q-Learning usually monotonically decreasing? Explain why / why not.

3.3 What are the differences and similarities between (tabular) Q-Learning and Dynamic Programming?

4. Topic: “Transformers”

4.1 How does the transformer get information of the order of words and how is this implemented?

4.2 How is an attention filter created in the transformer and how is cosine similarity taken into account?

4.3 Why does the decoder structure of the transformer use a masking in the attention Mechanism?

Literaturverzeichnis

Stache, P. D.-I. (n.d.). 1. Overview: From AI to Deep Learning. *Autonomous Systems: Deep Learning*. Heilbronn, Germany.

Stache, P. D.-I. (n.d.). 2. Familiarization with Python + Toolin. *Autonomous Systems: Deep Learning*. Heilbronn, Germany.

Stache, P. D.-I. (n.d.). 3. Your first neural network. *Autonomous Systems: Deep Learning*. Heilbronn, Germany.

Stache, P. D.-I. (n.d.). 4. Network Validation and Training. *Autonomous Systems: Deep Learning*. Heilbronn, Germany.

Stache, P. D.-I. (n.d.). 5. Deep Neural Networks. *Autonomous Systems: Deep Learning*. Heilbronn, Germany.

Stache, P. D.-I. (kein Datum). 6. Convolutional Neural Networks. *Autonomous Systems: Deep Learning*. Heilbronn, Germany.

Stache, P. D.-I. (kein Datum). 7. Advanced Exploration. *Deep Reinforcement Learning Introduction*. Heilbronn, Germany.

Stache, P. D.-I. (kein Datum). 9. Embeddings, Word2Vec. *Autonomous Systems: Deep Learning*. Heilbronn, Germany.