UNIVERSITY OF OSLO

Candidate no

Faculty of mathematics and natural sciences

Deferred constituent exam in:	TEK5040 — Deep Learning for Autonomous Systems
Day of examination:	20th December 2018
Examination hours:	09:00 – 13:00

This problem set consists of 6 pages.

Appendices:	None.
Permitted aids:	None.

Please make sure that your copy of the problem set is complete before you attempt to answer anything.

(Continued on page 2.)

Contents

Problem 1 Convolutional layers (weight 6%)

Assume you have an input image x and consecutively apply two convolutional layers, both with kernel size $[3, 3]$ and stride 1 in both height and width dimension. We define the *field of view* of a neuron as the pixels in the image that may affect the output of the neuron. The neurons in the first layer thus have a field of view of size 3×3. What is the size of the field of view for a neuron in the second convolutional layer?

Problem 2 Optimization (weight 6%)

Why do we often prefer *stochastic* gradient descent, i.e. estimate the gradient on a mini-batch, rather than calculate the gradient on the whole dataset?

Problem 3 Bidirectonal RNN (weight 6%)

One common way to include future context when making a prediction for RNNs is to *delay* the predictions a certation number of time steps. What is a possible disadvantage of this compared to bidirectional RNNs?

Problem 4 External memory (weight 6%)

4a (weight 3%)

Do one bit memory cells make sense when using *location*-based addressing? Give a brief explanation.

(Continued on page 3.)

4b (weight 3%)

Do one bit memory cells make sense when using *content*-based addressing? Give a brief explanation.

Problem 5 RL reward shifting (weight 6%)

If we shift all intermediate rewards by a constant, e.g. subtract 1, may that change the optimal policy? Does this depend on if you are in a continual or episodic environment?

Problem 6 Data efficiency (weight 6%)

Some machine learning tasks typically needs more data than others, in order to perform well. Order the task of *image classification*, *object detection* and *pixel segmentation* in terms of how many training images you typically need. Why are some of these tasks often more data efficient, even on the same image material?

Problem 7 Tracking in video (weight 10%)

7a (weight 5%)

A siamese network can be used for object tracking in video. Describe a method for tackling changes in depth/distance for a siamese tracking network.

7b (weight 5%)

Why can a siamese network be a simple and efficient way of tracking *multiple* objects in a video?

Problem 8 Graph-convolutions (weight 10%)

Graph convolutions are often used for 3D segmentation. Parametric kernels are one way to do graph-convolutions.

8a (weight 5%)

How do you get the weights for your filter in a parametric kernel convolution? Give two examples of possible kernels.

(Continued on page 4.)

8b (weight 5%)

What is the main benefit of a graph-convolutional operator compared to a standard convolutional operator.

Problem 9 Generative Adverserial Network (weight 10%)

9a (weight 5%)

Sketch and briefly explain the general architecture of a Generative Adverserial Network that generates images of numbers from 1 to 10.

9b (weight 5%)

Explain how you would modify the architecture, so you can control which number are generated at each run.

Problem 10 Bayesian deep learning (weight 11%)

10a (weight 5%)

Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ respectively be a sequence of input vectors and the corresponding outputs of a neural network parameterized by $\boldsymbol{w}$. Assume that the neural network models the likelihood distribution $p\left(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{w}\right)$ and that the prior distribution $p(\boldsymbol{w})$ is known. In Bayesian approach we try to estimate the posterior distribution $p\left(\boldsymbol{w}|\boldsymbol{Y},\boldsymbol{X}\right)$. Write an equation for the posterior distribution in terms of the likelihood and prior distributions.

10b (weight 3%)

Explain why it is difficult to perform exact Bayesian inference in practical deep learning.

10c (weight 3%)

Explain how the Bayesian approach can be utilized in safety critical applications such as medical diagnosis, self-driving cars and military applications

Problem 11 Deep learning for control (weight 12%)

11a (weight 3%)

Briefly explain two differences between control policy learning and perception learning.

(Continued on page 5.)

11b (weight 3%)

Guided Cost Learning (GCL) is a sampling based maximum entropy inverse reinforcement learning algorithm. Why is it more suited for systems with larger state-spaces than dynamic programming based maximum entropy inverse reinforcement learning?

11c (weight 3%)

GCL can be seen as a form of Generative Adversarial Network (GAN) which is typically characterized by a *generator, a discriminator, image data samples* and a *real image data samples*. What are the corresponding quantities of GCL?

11d (weight 3%)

State the main difference between GCL and Generative Adversarial Imitation Learning (GAIL).

Problem 12 Sequence modeling (weight 11%)

12a (weight 2%)

Briefly describe the phenomenon known as *exposure bias* in sequence generation training tasks such as machine translation and abstractive summarization.

12b (weight 3%)

Why is reinforcement learning a solution to the problem of exposure bias?

12c (weight 3%)

A machine translation system based on recurrent neural networks is trained using reinforcement learning with the BLEU evaluation metric. Identify *agent, state, policy, action* and *reward* of the associated reinforcement learning system.

12d (weight 3%)

In reinforcement learning based sequence generation task, we aim at maximizing the expected reward $L(\theta) = \sum_{\boldsymbol{w}} p_\theta(\boldsymbol{w}) r(\boldsymbol{w})$, where $\boldsymbol{w}$ is the generated word sequence, θ is the model parameters and $r(\boldsymbol{w})$ is the total reward for the word sequence $\boldsymbol{w}$. The gradient of $L(\theta)$ can be shown to be

$$\nabla_\theta L(\theta) = \sum_{\boldsymbol{w}} p_\theta(\boldsymbol{w}) r(\boldsymbol{w}) \nabla_\theta \log p_\theta(\boldsymbol{w})$$

(Continued on page 6.)

This quantity can be approximated with the sample mean. Write an expression for this approximated gradient assuming that N samples of word strings $\boldsymbol{w}^s, \;\; s = 1, 2, \ldots, N$ are drawn from the sequence generator.