UNIVERSITY OF OSLO

Candidate no

Faculty of mathematics and natural sciences

Deferred constituent exam in: TEK5040 — Deep Learning for Autonomous Systems
Day of examination: MOCK EXAM
This problem set consists of 5 pages.
Appendices: None.
Permitted aids: None.

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

Contents

Problem 1 Tensorflow (weight 5%)

What will be the printed output of the following code?

```
import tensorflow as tf
a = tf.constant(5)
b = tf.constant(3)
sess = tf.Session()
```

(Continued on page 2.)

```
print(sess.run(a*b + a, feed_dict={a: 3}))
print(sess.run(a*b + a)
```

Problem 2 Data efficiency (weight 5%)

Typical examples of convolutional neural networks use millions of images to get good results. What common methods can we use, to train neural networks with less data?

Problem 3 Self-supervised learning (weight 8%)

Recently self-supervised or semi-supervised learning approaches have been getting more attention. Instead of annotated data with ground truth labels, you can use assumptions to train your network. The vid2depth network used self-supervised learning to learn depth and camera motion directly from video.

How can assumptions help you train a deep neural network with little or no annotations? *Give a few examples of possible assumptions that can be used to train a network for estimation of depth and camera motion.*

Problem 4 Tracking (weight 8%)

You have a trained object detection network and want to train a general object tracking network. You get a bounding box a long representing an object, along with an image, and should be able to track any object.

4a

Describe how can you design such a tracking network. How do you train your network and how do you run it?

4b

What is a typical problem with general object tracking networks and how can you alleviate it?

Problem 5 3D segmentation (weight 5%)

On many 3D image datasets, multi-view segmentation solutions give the best results. Describe some typical scenarios, where *other* 3D deep learning techniques should work better.

(Continued on page 3.)

Problem 6 Optimization (weight 5%)

When we optimize neural networks we usually use a version of (stochastic) gradient descent. Why we would we not always calculate second order derivatives to get curvature information which we could use to make better local steps?

Problem 7 Batch normalization (weight 5%)

During training the incoming data is normalized in a batch-normalization layer. How do we deal with batch-normalization during inference?

Problem 8 Bidirectional RNNs (weight 4%)

What is the motivation behind bidirectional RNNs?

Problem 9 External memory (weight 8%)

Assume we use content-based addressing over a memory M, with three d-dimensional memory cells M_1, M_2 and M_3. Assume that we match a query vector q against key vectors k_1, k_2 and k_3 for the three memory cells. Let α_1, α_2 and α_3 denote the matching scores for the three memory cells. How would you proceed to return a d-dimensional vector with memory contents, both in the case of (1) *hard* addressing and (2) *soft* addressing?

Problem 10 RL value functions (weight 5%)

Assume an agent has a policy π, let s be a state and assume there are two possible actions, a_1 and a_2. Let v_π be the state-value function and q_π the action-value function. Assume $\pi(a_1|s) = 0.2$ and $\pi(a_2|s) = 0.8$. Assume $q_\pi(s, a_1) = -10$ and $q_\pi(s, a_2) = 10$. What is $v_\pi(s)$?

Problem 11 RL policy gradient (weight 7%)

Let π_θ be a differentiable parametrized family of policies, with parameters θ. A version of the policy-gradient update rule, with an update at end of each episode, is given by

$$\theta \leftarrow \theta + \alpha \sum_{t=1}^{\tau} \nabla_\theta \log \pi_\theta(a_t|s_t) G_t \tag{1}$$

where s_t and a_t is the state and action at time t and G_t is the *return*, i.e. the discounted future reward from time t. τ is the episode length and α is a positive real number. Give a brief interpretation of this update rule.

(Continued on page 4.)

Problem 12 Word embedding (weight 5%)

Skip-gram model of word embedding consists of two linear operations and a softmax operation. Write the equations for the forward pass of the system, assuming that the first and the second linear operations are defined by $\mathbf{W} \in \mathbb{R}^{V \times d}$ and $\mathbf{U} \in \mathbb{R}^{V \times d}$ where V is the vocabulary size and d is the embedding dimension. Indicate the dimensions of the vectors and matrices of your equations. How do you interpret the rows of the matrix $\mathbf{W}$?

Problem 13 Loss functions (weight 5%)

When is the Negative Sample Loss (NSL) equivalent to Noise Contrastive Estimation (NCE)?
You can use the expressions of loss functions for noise contrastive estimation (E_{NCE}) and negative sample loss (E_{NSL})are given by

$$E_{NCE} = \log\left[\frac{\exp\left(z\left(y^d\right)\right)}{\exp\left(z\left(y^d\right)\right) + kP_n(y^d)}\right] + \sum_{j=1}^{k} \log\left[\frac{kP_n(y_j^n)}{\exp\left(z\left(y_j^n\right)\right) + kP_n(y_j^n)}\right]$$

$$E_{NSL} = \log\left[\sigma\left(z\left(y^d\right)\right)\right] + \sum_{j=1}^{k} \log\left[\sigma\left(-z\left(y_j\right)^n\right)\right]$$

where y^d is a target class for a data sample, y_j^n is a target class drawn from a noise distribution P_n, $z(y)$ is the y^{th} input to the softmax, σ is a sigmoid function and k is the number of noise samples per data sample.

Problem 14 Bayesian deep learning (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 $\boldsymbol{w}$. Assuming the neural network models the distribution $p\left(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{w}\right)$, write the equations for modeling the neural network parameters $\boldsymbol{w}$ using approaches of Maximum Likelihood (ML), Maximum a-Posteriori (MAP) and Bayesian.

Problem 15 Evidence lower bound (weight 5%)

Evidence Lower Bound (ELBO) in variational inference is given by

$$\mathbb{E}_{q(\boldsymbol{w})} p(\mathcal{D}|\boldsymbol{w}) - \mathrm{KL}\big(q(\boldsymbol{w})||p(\boldsymbol{w})\big)$$

where $q(\boldsymbol{w})$ is an auxiliary distribution with which we try to approximate the posterior $p(\boldsymbol{w}|\mathcal{D})$ whereas $\boldsymbol{w}$ and $\mathcal{D}$ are network parameters and training data respectively. Interpret the two terms in the above expression when we maximize ELBO

(Continued on page 5.)

Problem 16 Guided cost learning (weight 5%)

Guided Cost Learning (GCL) is a sampling based maximum entropy inverse reinforcement learning algorithm. The gradient of the loss function $\mathcal{L}$ is given by

$$\frac{d\mathcal{L}}{d\theta} = \frac{1}{N}\sum_{\tau_i \in \mathcal{D}_{\text{demo}}} \frac{dc_\theta}{d\theta}(\tau_i) - \frac{1}{Z}\sum_{\tau_j \in \mathcal{D}_{\text{samp}}} w_j \frac{dc_\theta}{d\theta}(\tau_j)$$

with $w_j = \frac{\exp(-c_\theta(\tau_j))}{q(\tau_j)}$ and $Z = \sum_j w_j$. Here, $c_\theta(\cdot)$ is the cost function implemented as a neural network with parameters θ which are to be estimated. $\mathcal{D}_{\text{demo}}$ and $\mathcal{D}_{\text{samp}}$ are respectively the sets of expert demonstrations and system generated data samples consisting of trajectories τ. The probability of a sample trajectory τ_j is given by $q(\tau_j)$.

What quantities do you need to back-propagate through the neural network implementing the cost function in training? Outline the basic steps for calculating these quantities.

Problem 17 Dialog systems (weight 5%)

Briefly describe two issues of dialog system training that can be addressed with reinforcement learning. Identify the *agent, state, policy, action* and *reward* of reinforcement learning system applied to dialog system training. In reinforcement learning based dialog 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 utterance, θ is the model parameters and $r(\boldsymbol{w})$ is the total reward for the utterance $\boldsymbol{w}$. Gradient of $L(\theta)$ can be estimated using a single sample of utterance $\boldsymbol{w}^s$ with

$$\nabla_\theta L(\theta) \approx r(\boldsymbol{w}^s) \nabla_\theta \log p_\theta(\boldsymbol{w}^s)$$

One problem with the above estimate is that it has a high variance. Give an outline of a method for reducing the variance of the estimate and write a modified version of the above equation to support your answer.

Problem 18 Sequence modeling (weight 5%)

Path length in sequence modeling is defined as the number of neural network block operations required to relate two vectors in the input vector sequence. Write down the maximum path length in big O notation for a recurrent network, stacked convolution network (contiguous kernels) and a self-attention based network. Assume a sequence length of n, kernel size of k with $k < n$.