UNIVERSITY OF OSLO

Faculty of mathematics and natural sciences

Exam in:	IN 54000/IN 9400 — Machine Learning for Image Analysis
Day of examination:	31th May 2019
Examination hours:	14:30 – 18:30

This exercise set consists of 11 pages.

Appendices:	None
Permitted aids:	Certified calculator

Read the entire exercise text before you start solving the exercises. Please check that the exam paper is complete. If you lack information in the exam text or think that some information is missing, you may make your own assumptions, as long as they are not contradictory to the "spirit" of the exercise. In such a case, you should make it clear what assumptions you have made.

You should spend your time in such a manner that you get to answer all exercises shortly. If you get stuck on one question, move on to the next question.

Your answers should be short, typically a few sentences and / or a sketch should be sufficient.

Every subtask has equal weight in the evaluation.

(Continued on page 2.)

Exercise 1 Computational graph for a simple RNN

You are given a simple RNN network as illustrated in the computational graph. Assume that we use identity functions as activation functions.

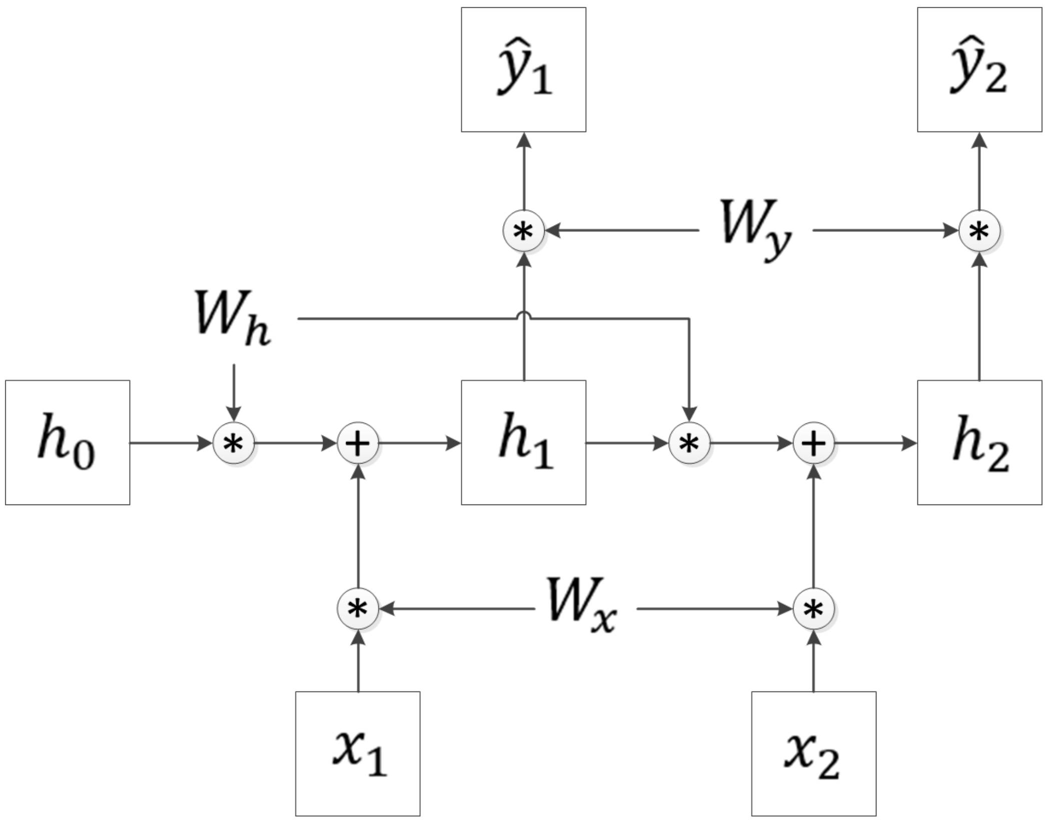


Figure 1: The structure of a very simple RNN network

Assume that the input at a given time, the hidden state, and the output at a given time are scalar.
Let $h_0 = 1, x_1 = 10, x_2 = 10, y_1 = 5, y_2 = 5$.
We assume the initial values of the weights are: $W_h = 1, W_x = 0.1, W_y = 2$

1a

Compute the predicted value $\hat{y}_2$.

$$h_1 = W_x x_1 + W_h h_0 = (0.1 * 10 + 1 * 1) = 2$$
$$\hat{y}_2 = W_y(W_h h_1 + W_x x_2) = 2(1 * 2 + 0.1 * 10) = 6$$

(Continued on page 3.)

1b

If we use quadratic loss, the loss at a given time t is $L_t = (\hat{y}_t - y_t)^2$. Compute the total loss given the values for the weights and inputs given above.

$$\hat{y}_1 = W_y(W_h h_0 + W_x x_1) = 2(1 * 1 + 0.1 * 10) = 4$$
$$L_1 = (\hat{y}_1 - y_1)^2 = (4-5)^2 = 1$$
$$L_2 = (\hat{y}_2 - y_2)^2 = (6-5)^2 = 1$$
$$L = L_1 + L_2 = 2$$

1c

Compute the derivative of the total loss with respect to h_1, $\frac{\partial L}{\partial h_1}$.

$$\frac{\partial L_1}{\partial h_1} = 2(\hat{y}_1 - y_1)\frac{\partial \hat{y}_1}{\partial h_1} = 2(\hat{y}_1 - y_1)W_y = 2 * (4-5) * 2 = -4$$

$$h_2 = W_h h_1 + W_x x_2 = 1 * 2 + 0.1 * 10 = 3$$
$$\hat{y}_2 = W_y(W_h h_1 + W_x x_2) = 2 * (1 * 2 + 0.1 * 10) = 6$$
$$\frac{\partial \hat{y}_2}{\partial h_1} = W_y W_h = 2 * 1 = 2$$
$$\frac{\partial L_2}{\partial h_1} = 2(\hat{y}_2 - y_2)\frac{\partial \hat{y}_2}{\partial h_1} = 2(\hat{y}_2 - y_2)2 = 2(6-5)2 = 4$$
$$\frac{\partial L}{\partial h_1} = \frac{\partial L_1}{\partial h_1} + \frac{\partial L_2}{\partial h_1} = (-4) + (4) = 0$$

1d

Compute the derivative of the total loss with respect to W_h, $\frac{\partial L}{\partial W_h}$.

$$\frac{\partial L}{\partial W_h} = \frac{\partial L_1}{\partial W_h} + \frac{\partial L_2}{\partial W_h}$$

$$\frac{\partial y_1}{\partial W_h} = W_y h_0 = 2 * 1 = 2$$

$$\frac{\partial L_1}{\partial W_h} = 2(\hat{y}_1 - y_1)\frac{\partial y_1}{\partial W_h} = 2(4-5) * 2$$

$$\frac{\partial y_2}{\partial W_h} = \frac{\partial(W_y(W_h h_1 + W_x x_2))}{\partial W_h} = \frac{\partial(W_y W_h h_1)}{\partial W_h}$$

(Continued on page 4.)

$$\frac{\partial y_2}{\partial W_h} = \frac{\partial (W_y W_h h_1)}{\partial W_h} = W_y h_1 + W_y W_h \frac{\partial h_1}{\partial W_h} = W_y h_1 + W_y W_h h_0$$

$$\frac{\partial y_2}{\partial W_h} = 2 * 2 + 2 * 1 * 1 = 6$$

$$\frac{\partial L_2}{\partial W_h} = 2(\hat{y}_2 - y_2)\frac{\partial y_2}{\partial W_h} = 2(6-5) * 6 = 12$$

$$\frac{\partial L}{\partial W_h} = \frac{\partial L_1}{\partial W_h} + \frac{\partial L_2}{\partial W_h} = (-4) + (12) = 8$$

Exercise 2 Convolutional neural networks, pooling and segmentation

2a

Which network architecture would you consider to be most parameter efficient on image data: convolutional neural network or dense neural network? Justify your answer.

A convolutional neural network is more parameter efficient as the filters are reused on the image data by the convolution operator.

2b

Discuss briefly whether convolutional neural networks build view invariance or not.

A CNN is not view invariant. A CNN with a fully convolutional head and no padding is translation invariant. Small rotational/scale invariance seems to be built. To achieve better view invariance we use data augmentation.

2c

Explain the different hyperparameters that define a convolutional layer.

- Stride
- Padding
- Kernel/filter size (heigt, width, number of filters)
- Dilation

(Continued on page 5.)

2d

Give three reasons to use max pooling.

- Max pooling is used to reduce the spatial size for lowering the memory constraint
- Build some scale invariance
- Build some local rotational invariance
- No trainable parameters
- Max pooling is a nonlinear operation which cannot be expressed by a convolutional layer directly

2e

Give two arguments against using max pooling.

- The max pooling layer can remove important information in the forward pass
- With max pooling a lot of the gradients will be zero and therefore the training can be slow

2f

PhD students only Give an example on how you could design a semantic segmentation network. Discuss which loss function you would use.

Suggested network: A u-net with the same number of channels in the output as in classes. Loss function: Cross Entropy Loss with Softmax activation function. Other plausible solutions are also credited.

Exercise 3 Training a neural network

3a

Describe the algorithm for gradient descent with momentum updates.

Solution proposal:

(Continued on page 6.)

$$v = \rho v - \lambda \partial w$$

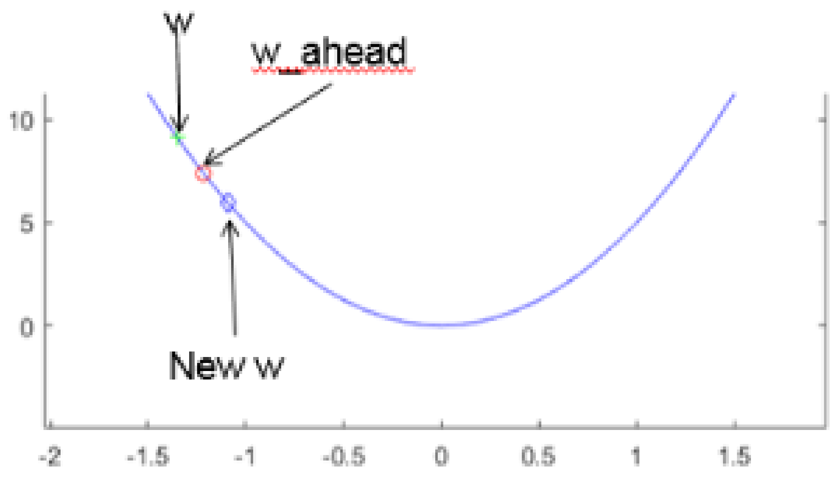
$$w = w + v$$

3b

PhD students only

Make a small sketch that illustrates Nesterov momentum updates

Solution sketch:
Compute the gradient at w_{ahead}, not at w.



3c

Gradients that are close to zero can be a problem when training a network. Suggest three different things that can reduce the problem.

- Use activation function such as ReLU and Leaky ReLU.
- Normalizing the input data
- Weight initialization such as Xavier and He
- Batch normalization

(Continued on page 7.)

- Using skip connections as in ResNet
- In RNN, using cells such as GRU and LSTM

Exercise 4 Generalization and regularization

4a

Give an example of how you could use transfer learning.

We can use transfer learning when our dataset is small. Select a network trained on another dataset and use it as a baseline. We normamlly reinitialize/change the first and/or the last layer(s) depending on the data format and the number of classes. Train the changed layers on the new datset.

4b

Describe briefly a few data augmentation techniques.

- Rotation
- Crop
- Exposure/contrast
- "random" noise
- Mirroring

4c

Explain region (1) and region (2) in the figure below. Explain also the double arrow (3).

- Region (1): Underfitting, the model capacity is too small to fit the data.
- Region (2): Overfitting, the model capacity is large and fits the data too well.
- Double arrow (3): generalization gap

(Continued on page 8.)

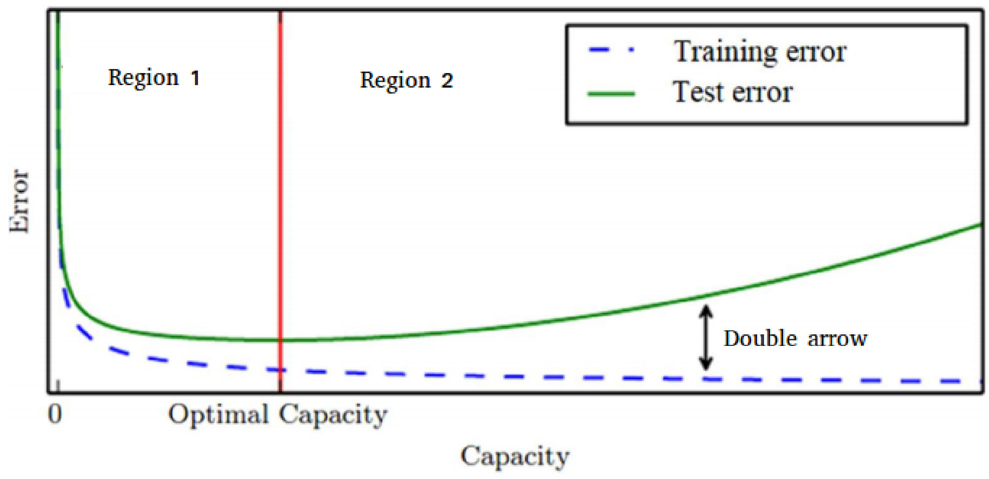


Figure 2

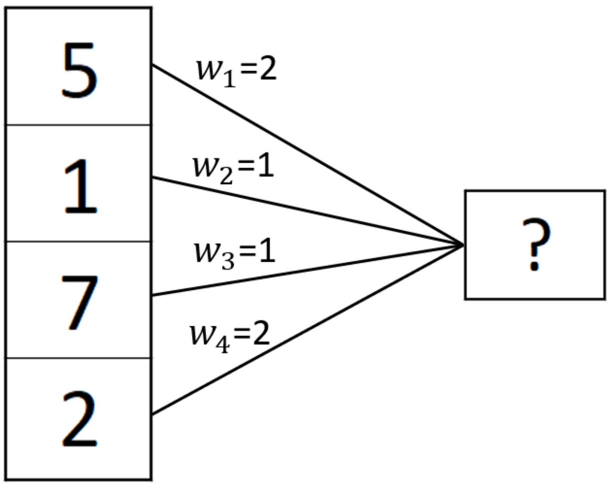


Figure 3

4d

Given the dense layer shown in figure 3. What would the **maximum** value of the output be assuming a dropout probability of p=0.5?

Here two interpretations are possible:

1. Only 50% of the nodes will be active, so finding the maximum is equivalent to finding the two nodes that results in the maximum output. Maximum value: $(5 \cdot 2 + 0 \cdot 1 + 7 \cdot 1 + 0 \cdot 2)/0.5 = 34$

2. Over time with dropoupt 0.5, all nodes will on, so the maximum value can be

(Continued on page 9.)

Maximum value: $(5 \cdot 2 + 1 \cdot 1 + 7 \cdot 1 + 2 \cdot 2)/0.5 = 44$

Note: Assuming we are in the training phase, we scale with 1/p as we want the expected value to be equal during test time. Alternatively, we can omit the 1/p during training and multiple with p during testing.

4e

Given the dense layer shown in figure 3. What would the **minimum** value of the output be assuming a dropout probability of p=0.5?

Here two interpretations are possible:

1. Only 50% of the nodes will be active, so finding the minimum is equivalent to finding the two nodes that results in the minimum output. Minimum value: $(0 \cdot 2 + 1 \cdot 1 + 0 \cdot 1 + 2 \cdot 2)/0.5 = 10$

2. Over time with dropoupt 0.5, all nodes will be dropped, so the minimum value can be 0.

Note: Assuming we are in the training phase, we scale with 1/p as we want the expected value to be equal during test time. Alternatively, we can omit the 1/p during training and multiple with p during testing.

Exercise 5 Visualization and adverserial fooling

5a

Consider the last layer before the softmax layer in a convolutional network. Suggest a method for visualizing the information learned by the set of features or feature vectors in this layer.

This has the features that the net has extracted, and the softmax acts as a classifier of these features. At least two approaches are common: if we take one particular image, we can find the closest images measured by distance in feature space. This will indicate the type of invariance we have learned. We can also use t-SNE to visualize a lower-dimensional projection of the feature space.

5b

Explain shortly how we can use backpropagation with respect to the input image to visualize a filter.

(Continued on page 10.)

To visualize how **one filter** works, use Guided backprop. Given a filter in a layer, forward propagate an image to the given layer, set all gradients except for the given filter to zero, and backpropagate to change the input image (set negative gradients to zero and do a special step for backpropagation through ReLU). A less detailed answer that describes the main principles would also be OK here.

5c

Explain briefly what an adversarial image is.

An adversarial image is an image that has been modified slightly such that it looks like the original class, but is classified as a different class.

5d

Describe briefly the fast gradient sign method for generating adversarial examples.

Select an image and a class. Forward propagate to get the score for class y. Backpropagate with respect to the class y to get the gradients with respect to class y. Update the image x by taking a step in the direction of the sign of the gradient.

Exercise 6 Unsupervised learning

6a

t-SNE is often used to vizualize high dimentional data by embedding the data to a lower dimention (2D/3D). How would you inculde a new sample in the t-SNE?

We don't have an explicit formula for the t-SNE. To include a new sample, we will need to minimize the objective function over again. This is different than PCA which is defined by a matrix multiplication.

6b

How does t-SNE mitigate the crowding problem?

t-SNE uses the student-t distribution with one degree of freedom (in the low dimention) when calculating the probability of points being

(Continued on page 11.)

neighbours. The student-t distribution is a heavy-tailed distrbution allowing points which are far apart in the high dimention to be even further apart in the low dimention.