UNIVERSITY OF OSLO

Faculty of mathematics and natural sciences

Exam in:	IN3310/IN4310 — Deep Learning for Image Analysis
Day of examination:	June 4, 2024
Examination hours:	9:00 – 13:00
This problem set consists of 17 pages.	
Appendices:	None
Permitted aids:	None

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

The course teachers will visit the exam room once around the middle of the exam time.

Appendices: None

Permitted aids: None

An on-screen calculator is available.

Information about hand drawings (Scantron)

In this question set you have the opportunity to answer with hand drawings (Problem 3a and 3b optionally). You use the handed out sketch sheets. It is possible to use several sheets per assignment. See instructions on how to fill in the sketch sheets in the link below the assignment overview.

You are not supposed to submit hand drawings for any other questions than Problem 3a and 3b. You will NOT be given extra time to fill in the information boxes on the sketch sheets (assignment codes, candidate number, etc.)

(Continued on page 2.)

Problem 1

Suppose you want to predict the value of a house as a function of the following inputs: 1) Size of the house x_1; 2) Number of bedrooms x_2; Age of the house x_3; 4) Monthly energy usage of the house x_4. Let y denote the true value and $\hat{y}$ denote the prediction. Suppose you use a neural network that predicts $\hat{y}$ as follows:

$$\hat{y} = \sum_{j=1}^{5} v_j g\Big(\sum_{i=1}^{4} w_{i,j} x_i + b_j\Big) + b$$

where $v_j, w_{i,j}, b_j, b$ are trainable parameters and g is an activation function.

1a

Is this problem classification or regression? Why?

Regression since the output takes continuous values.

1b

How many hidden layers does the neural network have? How many hidden nodes (neurons in hidden layers) does it have?

One hidden layer with 5 hidden nodes.

1c

Suppose you want to slightly modify the final layer of the neural network so the output becomes non-negative regardless of the value of trainable parameters and without changing the number of parameters. Note that the desired output can be any and possibly large positive number but cannot be a negative number. How do you modify the final (output) layer?

$$\hat{y} = \text{ReLU}\Big(\sum_{j=1}^{5} v_j g\Big(\sum_{i=1}^{4} w_{i,j} x_i + b_j\Big) + b\Big)$$

(Continued on page 3.)

Problem 2

2a

A two-dimensional convolutional layer applies filters with spatial size $(5, 5)$, stride 2, and standard padding to an input of size (32,32). What will be the spatial output size?

$$N^{[2]} = \left\lfloor \frac{N^{[1]} - 1}{2} + 1 \right\rfloor$$

We note that

$$\left\lfloor \frac{32 - 1}{2} + 1 \right\rfloor = 16$$

The output size is $(16, 16)$.

2b

Suppose that a number of above filters with spatial size $(5, 5)$, stride 2, and standard padding, 10 input channels and 100 output channels, and no bias terms are applied in a layer. What is the number of trainable parameters in this layer?

$$5 \times 5 \times 10 \times 100 = 25000$$

2c IN4310 students only

Suppose we use a second two-dimensional convolutional layer having filters with spatial size $(3, 3)$, stride 3, and standard padding on top of the first convolutional layer described above and a third convolutional layer having filters with spatial size $(3, 3)$, stride 1, and standard padding on top of the second convolutional layer.

What is the spatial size of the region of the input to the first layer that will be used to compute the output of a single neuron in the third layer? You need to answer only for neurons in the center of the feature maps and ignore padding effects at the boundaries.

Include your calculations in your answer even if you also include drawings

*Hint $R^{[0]} = 1$ and $R^{[1]} = R^{[0]} + (k^{[1]} - 1)$ where $k^{[1]}$ is the spatial size of the first layer filter

$$R^{[1]} = 1 + (5 - 1) = 5$$
$$R^{[2]} = 5 + (3 - 1) \times 2 = 9$$
$$R^{[3]} = 9 + (3 - 1) \times 2 \times 3 = 21$$

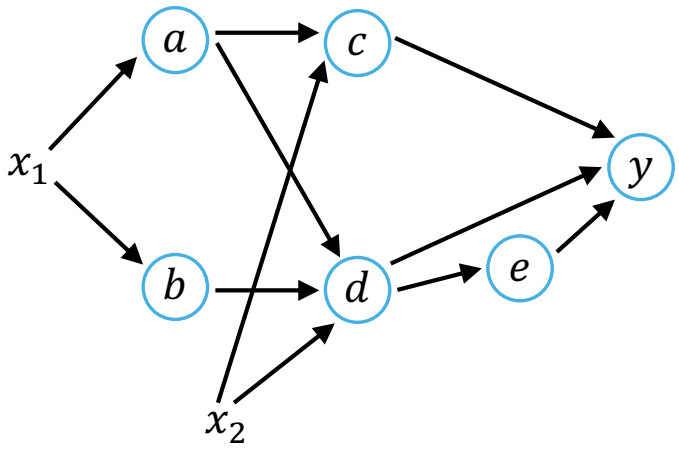
(Continued on page 4.)

The spatial size of the receptive field is $(21, 21)$.

(Continued on page 5.)

Problem 3

Consider the following neural network:



The output (activation) for any neuron $q \in \{a, b, c, \ldots\}$ is given by:

$$g\left(b + \sum_l v_l x_l + \sum_u w_u u\right) \tag{1}$$

where $g(\cdot)$ is some activation function, b is the bias term, $v_l = 0$ for neurons that have no direct inputs x_l, and $w_u = 0$ for neurons that have no inputs from neuron u.

This task asks you to write down expressions for gradients of this network. Please consider the following advices before doing so:

- Write the expression in terms of:
  - $\frac{\partial q_1}{\partial q_2}$, where q_2 is direct input to q_1, and
  - $\frac{\partial q}{\partial x_l}$, where the network input x_l is direct input to neuron q.
- This task is about the sequences of partial derivatives along the edges.
- You **do not need** to plug in or compute how $\frac{\partial q_1}{\partial q_2}$ or $\frac{\partial q}{\partial x_l}$ looks like.
- You **do not need** to multiply out terms in parentheses, so $(a + b)c$ or $((a + b)c + (d + e)f)g$ is fine to keep like that.

Write down an expression for the following gradients of this network:

3a

$\frac{\partial y}{\partial x_2}$

(Continued on page 6.)

$$\begin{aligned}\frac{\partial y}{\partial x_2} &= \frac{\partial y}{\partial c}\frac{\partial c}{\partial x_2} + \frac{\partial y}{\partial d}\frac{\partial d}{\partial x_2} + \frac{\partial y}{\partial e}\frac{\partial e}{\partial x_2} \\ &= \frac{\partial y}{\partial c}\frac{\partial c}{\partial x_2} + \frac{\partial y}{\partial d}\frac{\partial d}{\partial x_2} + \frac{\partial y}{\partial e}\frac{\partial e}{\partial d}\frac{\partial d}{\partial x_2}\end{aligned}$$

3b

$\frac{\partial y}{\partial x_1}$

$$\begin{aligned}\frac{\partial y}{\partial x_1} &= \frac{\partial y}{\partial c}\frac{\partial c}{\partial x_1} + \frac{\partial y}{\partial d}\frac{\partial d}{\partial x_1} + \frac{\partial y}{\partial e}\frac{\partial e}{\partial x_1} \\ &= \frac{\partial y}{\partial c}\frac{\partial c}{\partial x_1} + \frac{\partial y}{\partial d}\frac{\partial d}{\partial x_1} + \frac{\partial y}{\partial e}\frac{\partial e}{\partial d}\frac{\partial d}{\partial x_1} \\ &= \frac{\partial y}{\partial c}\frac{\partial c}{\partial a}\frac{\partial a}{\partial x_1} + \left(\frac{\partial y}{\partial d} + \frac{\partial y}{\partial e}\frac{\partial e}{\partial d}\right)\frac{\partial d}{\partial x_1} \\ &= \frac{\partial y}{\partial c}\frac{\partial c}{\partial a}\frac{\partial a}{\partial x_1} + \left(\frac{\partial y}{\partial d} + \frac{\partial y}{\partial e}\frac{\partial e}{\partial d}\right)\left(\frac{\partial d}{\partial a}\frac{\partial a}{\partial x_1} + \frac{\partial d}{\partial b}\frac{\partial b}{\partial x_1}\right)\end{aligned}$$

(Continued on page 7.)

Problem 4

Let $h_\theta : \mathcal{X} \to \mathcal{Y}$ denote a predictor parameterized by $\theta \in \mathbb{R}^d$. Suppose we have collected N independently and identically distributed samples (x_n, y_n) to train our predictor and use some loss function ℓ so $\ell(h_\theta(x_n), y_n)$ is the loss on sample n.

4a

Suppose the optimization problem for empirical risk minimization (ERM) is denoted by $\min_\theta R_N(\theta)$. What is $R_N(\theta)$ in terms of training samples?

$$R_N(\theta) = \frac{1}{N} \sum_{n=1}^{N} \ell(h_\theta(x_n), y_n)$$

4b

Instead of training the predictor with Gradient Descent (GD), your friend suggests you to use Stochastic Gradient Descent (SGD). Why does your friend think SGD is better than GD?

Computational complexity when N is large

4c

When you implement SGD by drawing one sample from training samples uniformly at random and computing stochastic gradients, you note that stochastic gradients are too noisy and variance is huge. How can you control the variance?

Use a mini-batch instead of drawing one sample

(Continued on page 8.)

Problem 5

5a

What is true for "Dropout Regularization"? Briefly explain your selection(s)!

1. Dropout removes neurons out of the network at inference time.
2. Dropout is proposed to improve generalization.
3. If a neuron is active and its activation dominates the output of the network, dropout ensures its activation throughout training.

2 is Correct

5b

Consider the following code:

```
class SimpleCNN(nn.Module):
    def __init__(self, num_classes=10):
        super(SimpleCNN, self).__init__()
        self.conv1 = nn.Conv2d(...)
        self.conv2 = nn.Conv2d(...)
        self.conv3 = nn.Conv2d(...)
        self.conv4 = nn.Conv2d(...)
        self.conv5 = nn.Conv2d(...)
        self.fc1 = nn.Linear(...)

    def forward(self, x):
        x = F.relu(self.conv1(x))
        x = F.relu(self.conv2(x))
        x = F.relu(self.conv3(x))
        x = F.relu(self.conv4(x))
        x = F.relu(self.conv5(x))
        x = self.fc1(x)

        return x
```

By modifying, adding, removing **at most three lines of code** in total, how do you change the code such there there exists a residual connection between the "input to self.conv3" and the "output of self.conv5"?

```
class SimpleCNN(nn.Module):
    def __init__(self, num_classes=10):
        super(SimpleCNN, self).__init__()
        self.conv1 = nn.Conv2d(...)
        self.conv2 = nn.Conv2d(...)
        self.conv3 = nn.Conv2d(...)
        self.conv4 = nn.Conv2d(...)
        self.conv5 = nn.Conv2d(...)
        self.fc1 = nn.Linear(...)
```

(Continued on page 9.)

```
    def forward(self, x):
        x = F.relu(self.conv1(x))
        z = F.relu(self.conv2(x))
        x = F.relu(self.conv3(z))
        x = F.relu(self.conv4(x))
        x = z+F.relu(self.conv5(x))
        x = self.fc1(x)

        return x
```

(Continued on page 10.)

Problem 6

For a model $f : \mathcal{X} \to \mathcal{Y}$ and a transformation $T : \mathcal{X} \to \mathcal{X}$, we want the following property to hold:

$$f(T(x)) = f(x), \quad \forall x \in \mathcal{X}.$$

6a

Explain how this property is used for training neural networks with data augmentation in a *supervised* context. Discuss how this affects a model's ability to generalize.

The **invariance property** ensures that the label y assigned to the original input x remains the same after transformation $T(x)$, forming the basis for data augmentation techniques. Invariance improves generalization by exposing the model to augmented data, making it robust to variations and preventing overfitting. Transformations that do not preserve the label will however degrade model performance.

6b

Discuss how this is applied for training models without supervised labels.

Without supervised labels, transformations T can be used for self-supervised learning where the model learns ensure consistency in embeddings of original and transformed inputs. This forms the basis for contrastive learning.

(Continued on page 11.)

Problem 7

7a

Discuss the impact of image compression on adverserial attacks.

Solution hint:

- **Impact on Adversarial Perturbations:** Adversarial perturbations are often subtle and rely on small, precise changes to pixel values. Compression algorithms, such as JPEG, typically introduce quantization and other artifacts that can disrupt these small changes, thereby diminishing or eliminating the adversarial effect.
- **Defense Mechanism:** Compression can act as a form of defense against adversarial attacks by smoothing out high-frequency noise and reducing the precision of pixel values, making it harder for adversarial examples to maintain their malicious influence after compression.
- **Trade-offs:** While compression can mitigate adversarial attacks, it may also degrade the overall image quality and potentially impact the performance of the model on legitimate inputs, especially if the compression is aggressive.

7b

Explain the concept of *gradient masking* as an adverserial defence strategy, and relate this to the effect of image compression on adverserial attacks.

Solution hint: Gradient masking involves limiting the usefulness of gradients to generate adversarial examples. By obscuring or altering gradient information, it becomes more challenging to compute effective perturbations that can deceive the model. One technique for gradient masking is to use non-differentiable layers. Compression algorithms often use non-differentiable steps (e.g., quantization) which effectively obscures the gradient information.

(Continued on page 12.)

Problem 8

Consider a dataset for a classification task. It consists of 1000 samples. 970 samples have label A, and the remaining 30 have label B.

8a

Is accuracy a good metric to evaluate the performance of a classifier on this dataset? If not, then which other metric would be a good choice? Explain briefly.

Solution hint: Accuracy is not a good choice because the dataset is imbalanced. Accuracy can be misleading for imbalanced datasets because it doesn't take the class distribution into account. A model can achieve high accuracy simply by correctly classifying most of the class A samples, even if it performs poorly on class B. Balanced accuracy is a good choice for an imbalanced dataset as it treats all classes equally and provides a more balanced view of the model's performance across classes.

8b

Suppose we care more about class B than class A. We don't want our classifier to misclassify class B as A, but classifying some instances of class A as class B is tolerable. Which of the following metrics would you look at for class B? Briefly justify your choice in at most 3 sentences. **Note:** class B is the positive class for these metrics.

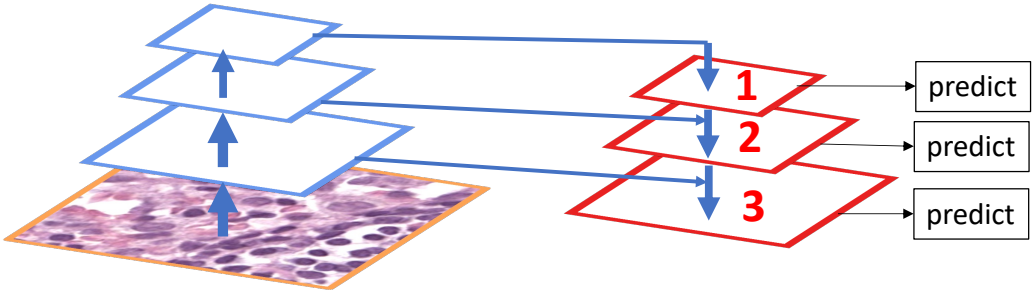
1. Precision
2. Recall / Sensitivity
3. Specificity

Solution hint: Option 2. Recall gives us the percentage of positive elements found. High recall would indicate that our classifier is predicting class B correctly.

(Continued on page 13.)

Problem 9

Consider the Feature Pyramid Network shown in the image below:



This network's predictions are made using anchor boxes on the 3 red feature maps numbered 1, 2, and 3.

9a

As seen in the image, the 3 feature maps have different resolutions. Furthermore, each feature map uses a different set of anchor boxes to make predictions. The resolution of the feature map and the number of anchor boxes used by it are listed below:

- Feature map 1: Resolution = 2×2. Anchor boxes = 5.
- Feature map 2: Resolution = 3×3. Anchor boxes = 4.
- Feature map 3: Resolution = 4×4. Anchor boxes = 2.

How many predictions will the network make in total? Show your calculations.

Solution hint: Feature map 1 has $2 \times 2 = 4$ locations to make predictions on. It will make 5 predictions per location. So, the total number of predictions for feature map $1 = 2 \times 2 \times 5 = 20$. Similarly, #predictions for feature map $2 = 3 \times 3 \times 4 = 36$ and for feature map $3 = 4 \times 4 \times 2 = 32$. So, the total number of predictions $= 20 + 36 + 32 = 88$

9b

To train the network, we need to match the predictions to the ground truth boxes. Given a prediction box and a list of ground truth boxes, how will you decide which ground truth box (or none of them) should be used as the ground truth for the prediction box?

Solution hint: Check the overlap between the prediction box P and the ground truth boxes via IoU (Intersection over Union). Pick the ground truth box with the maximum IoU. If the IoU is greater than a threshold, use it as the ground truth box for P.

(Continued on page 14.)

Problem 10

You are creating a Convolutional Network for image segmentation to segment out tumorous regions in the given input image. Your task is to not just identify the pixels corresponding to the tumour but also further grade the tumour as either grade 1, grade 2, grade 3, or grade 4.

10a

To reduce the number of parameters in your network, you would like to replace transposed convolution with nearest neighbour upsampling. If we want to upsample the matrix $\begin{bmatrix} 3 & 4 \\ 2 & 1 \end{bmatrix}$ by $2\times$ (make it twice as large), what would be the output of using nearest neighbour upsampling for that?

Solution hint: $\begin{bmatrix} 3 & 3 & 4 & 4 \\ 3 & 3 & 4 & 4 \\ 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \end{bmatrix}$

(Continued on page 15.)

Problem 11

11a

The cell state in LSTM is updated using the following equation:

$$\mathbf{C_t} = \mathbf{f_t} \odot \mathbf{C_{t-1}} + \mathbf{i_t} \odot \hat{\mathbf{C}}$$

Where f_t is the forget gate. Why is the forget gate's bias initialised to a high positive value? Explain briefly in 2-3 sentences.

Solution hint: High bias leads to the forget gate being open. This allows the gradients to flow through the cell state as it is, without vanishing. That enables LSTM to learn long-range dependencies.

11b

If a bidirectional LSTM layer has 10 LSTMs for each direction separately, what would be its output size? Justify your answer in 1-2 sentences.

Solution hint: A bidirectional RNN concatenates the output of both directions. Therefore, the output size will be 10 + 10 = 20.

(Continued on page 16.)

Problem 12

12a

Self-attention computes dot products between keys and queries. If a given self-attention has 4 attention heads and its one-dimensional input size is 3, how many dot products will it compute? Show your calculations.

Solution hint: Each attention head will compute $3 \times 3 = 9$ dot products. There are 4 such attention heads, so the total number of dot products $= 9 \times 4 = 36$.

12b

Swin Transformer limits the computation self-attention to a local window. Patches within a window do not interact with patches from other windows in a given block. This prevents the layer from capturing the global context in that block. What does Swin Transformer do to solve this problem when going from one block to the next? Briefly explain in 2-3 sentences.

Solution hint: Swin Transformer shifts its windows when going form one block to the next. This allows patches that were not in the same local window to interact with each other.

(Continued on page 17.)

Problem 13

Suppose you are developing a deep learning model to classify emails as either "spam" or "not spam" based on various features such as the content of the email, the sender's address, and the presence of certain keywords. Within your training date, you notice that 40% of emails are classified as spam. Over time, the behavior of email users and spammers changes. For the test data, you notice that the proportion of spam emails has decreased significantly to 10% because email providers have implemented more stringent spam filters and security measures. Meanwhile, the distribution of features given spam and not spam labels does not change.

13a

What is the type of distribution shift you are facing in this example?

Label shift

13b

What is the expression for importance weight (density ratio) in terms of train and test distributions?

$$\frac{p^{te}(y)}{p^{tr}(y)}$$