UNIVERSITY OF OSLO

Faculty of mathematics and natural sciences

Exam in: IN3310/IN4310/IN5400/IN9400 –

Day of examination: June 15, 2023

Examination hours: 9:00 – 13:00

This exercise set consists of 18 pages.

Appendices: None

Permitted aids: None

General information:

- Read the entire exercise text before you start solving the exercises. Please check that the exam paper is complete.
- Remember that your exam answers must be anonymous; do not state either your name or that of fellow students.
- Most of your text answers should include a discussion and be brief, typically at most a few sentences.
- When you do calculations to answer an exercise, include the intermediate steps in your answer.
- 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 justifiable within the context of the exercise. In such a case, you should make it clear what assumptions you have made.
- Plan your time so that you can try to answer as many subtasks as possible. Each subtask is normally weighted equally.

Cheating on school exams

"For school exams, it is considered as cheating to use support materials, unless it is explicitly stated in the exam question that this is permitted. Having access to illegal support materials may also be considered cheating, even if they are not used.

All communication between candidates in the exam room is prohibited. Contact with other candidates or other persons during trips to the lavatory and breaks is also prohibited. Mobile phones and other electronic equipment should be turned off and packed away."

`https://www.uio.no/english/studies/examinations/cheating/index.html`

(Continued on page 2.)

Exercise 1

Consider the logistic regression classifier:

$$f(x) = \sigma(\sum_{d=0}^{D-1} w_d x_d + b) \tag{1}$$

where:

$$\sigma(x) = 1/(1 + e^{-x}) \tag{2}$$

1a

The following question is multi-select question, that is, **more than one choice can be true**. What is true? Briefly explain your selection(s)!

1. The set $\{x : f(x) = const\}$ is parallel to w
2. The set $\{x : f(x) = const\}$ is orthogonal to w
3. If x is such that $\sum_{d=0}^{D-1} w_d x_d = -3b$, and we consider $z = x + 3bw/\|w\|$, then $f(z) = 0.5$
4. If x is such that $\sum_{d=0}^{D-1} w_d x_d = -3b$, and we consider $z = x + 2bw/\|w\|^2$, then $f(z) = 0.5$
5. If x is such that $\sum_{d=0}^{D-1} w_d x_d = -3b$, and we consider $z = x + 3bw/\|w\|^2$, then $f(z) = 0.5$

Solution hint: Options 2 and 4.

(Continued on page 3.)

Exercise 2

2a

What is the essential difference between a fully connected layer and a 1-D convolution layer with respect to computing outputs? Answer in 3 sentences at most, but name the difference specific to a 1-D convolution.

Solution hint: Convolution applies the inner product in a sliding window moving by a stride, implying both locality and weight sharing which is not found in fully connected layers. For a 1-D convolution layer specifically, the sliding window moves along 1 dimension.

2b

A 2-dimensional convolutional layer applies filters with spatial kernel size $(5,5)$, stride 4, and a padding of 5 to an input of shape (112,224). What will be the spatial output shape?

Solution hint:

$$floor((M+2r-ksize)/s+1) = floor(((112,224)+2*5-5)/4+1) \tag{3}$$

$$= floor((29.25,57.25)+1) = (30,58) \tag{4}$$

2c

Assume the 2-dimensional convolutional layer, applying filters with spatial size $(4,4)$, stride 2, and a padding of 3, has 100 input channels and 30 output channels, and no bias terms. What is the number of trainable parameters in this layer?

Solution hint: $100*4*4*30 = 48000$

(Continued on page 4.)

Exercise 3

3a

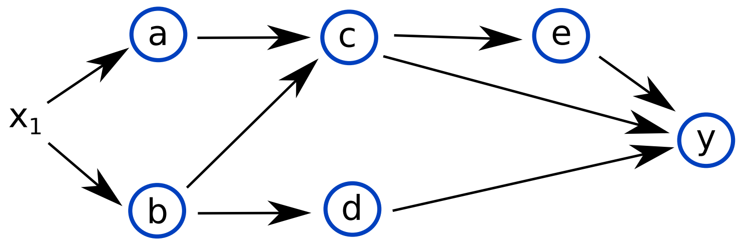
What is true? Briefly explain your selection(s)!

1. Backpropagation is the method of finite differences executed on a graph.
2. Backpropagation computes directional derivatives on a graph.
3. Backpropagation is chainrule executed on a graph.

Solution hint: Option 3.

3b

Consider the following neural network:



The equation for any neuron $k \in \{a, b, c, \dots\}$ is:

$$n\left(s + \sum_l v_l x_l + \sum_k w_k k\right) \tag{5}$$

where $n(\cdot)$ is some activation function, s is the bias term, $v_l = 0$ for neurons that have no direct inputs x_l, and $w_k = 0$ for neurons that have no inputs from other neurons.

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:
  - $\frac{\partial k_1}{\partial k_2}$, where k_2 is direct input to k_1, and
  - $\frac{\partial k}{\partial x_l}$, where x_l is input to the network.
- This task is about the sequences of partial derivatives along the edges.
- You **do not need** to plug in or compute how $\frac{\partial k_1}{\partial k_2}$ or $\frac{\partial k}{\partial x_l}$ looks like.

(Continued on page 5.)

- 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:

- $\frac{dc}{dx_1}$
- $\frac{dy}{dx_1}$

Solution hint:

- Either:

$$\frac{dc}{dx_1} = \frac{\partial c}{\partial a}\frac{\partial a}{\partial x_1} + \frac{\partial c}{\partial b}\frac{\partial b}{\partial x_1} \tag{6}$$

  or:

$$\frac{dc}{dx_1} = \frac{\partial c}{\partial a}\frac{da}{dx_1} + \frac{\partial c}{\partial b}\frac{db}{dx_1} \tag{7}$$

-

$$\frac{dy}{dx_1} = \frac{\partial y}{\partial e}\frac{de}{dx_1} + \frac{\partial y}{\partial d}\frac{dd}{dx_1} + \frac{\partial y}{\partial c}\frac{dc}{dx_1} = \frac{\partial y}{\partial e}\frac{\partial e}{\partial c}\frac{dc}{dx_1} + \frac{\partial y}{\partial d}\frac{dd}{dx_1} + \frac{\partial y}{\partial c}\frac{dc}{dx_1} \tag{8}$$

$$= \frac{\partial y}{\partial e}\frac{\partial e}{\partial c}\frac{dc}{dx_1} + \frac{\partial y}{\partial d}\frac{\partial d}{\partial b}\frac{\partial b}{\partial x_1} + \frac{\partial y}{\partial c}\frac{dc}{dx_1} \tag{9}$$

$$= (\frac{\partial y}{\partial e}\frac{\partial e}{\partial c} + \frac{\partial y}{\partial c})(\frac{\partial c}{\partial a}\frac{\partial a}{\partial x_1} + \frac{\partial c}{\partial b}\frac{\partial b}{\partial x_1}) + \frac{\partial y}{\partial d}\frac{\partial d}{\partial b}\frac{\partial b}{\partial x_1} \tag{10}$$

(Continued on page 6.)

Exercise 4

4a

Let $C(x)$ be the output of a stack of convolution layers C, computed from an input feature map x. What does a residual connection compute? Briefly explain your selection(s)!

1. $\sigma(C(x)) * C(x)$ where $\sigma(x) = \frac{1}{1+e^{-x}}$ is element-wise sigmoid weighting
2. $C(x) + x$
3. `avgpool( torch.cat( (C(x),x), dim=(1) ) )`
4. $C(x) - x$
5. $C(x) * x$

Solution hint: Option 2.

4b IN9400 students only

Answers from students in other courses will be ignored :-)

Name one advantage of using residual connections with respect to gradients. 3 sentences max.

Solution hint: Residual connections allow to flow gradients through the shortcut without a change in scale. This reduces the vanishing gradient problem. As such they contribute to gradients being more similar in scale for different neurons.

(Continued on page 7.)

Exercise 5

Consider the following optimizers:

1. SGD
2. SGD with momentum
3. RMSProp
4. AdamW

5a

Which of these are using normalization of gradients? Briefly explain your selection(s)!

Solution hint: Options 3 and 4.

5b

Which of these are using moving averages of gradients which are not applied as normalization? Briefly explain your selection(s)!

Solution hint: Options 2 and 4.

(Continued on page 8.)

Exercise 6

6a

Consider the following code:

```
def train_epoch(model,  trainloader,  losscriterion, device, optimizer):

        model.train()
        model = model.to(device)

        losses = []
        for batch_idx, data in enumerate(trainloader):

                inputs=data['image'].to(device)
                labels=data['label'].to(device)

                output = model(inputs)
                loss = losscriterion(output, labels)

                loss.backward()
                optimizer.step()

                losses.append(loss.item())
        return losses
```

Which one line of code is missing in the code above?

Solution hint: `optimizer.zero_grad()`

(Continued on page 9.)

Exercise 7

7a

- How is a test subset obtained?
- How does a test subset differ from an external dataset?

Solution hint:

- It is obtained by splitting the development dataset into two disjoint subsets – train and test.
- Unlike test subset, external dataset is NOT derived from the development dataset. It is external to the development data.

7b

Consider the following scenario: You train a model and get a good enough accuracy on the validation subset, but the accuracy on the test subset is lower than expected. So you change your training settings and retrain the model. You keep changing settings and retraining until the accuracy is good enough on the test subset. Do you expect that the accuracy is similarly better when the model is applied on new data as you observed for the test subset? Argue in at most 3 sentences.

Solution hint: No, because this procedure may lead to an overly optimistic accuracy on the test subset. The accuracy in the real world will likely not be as good as the performance on the test subset would suggest.

7c IN5400 and IN9400 students only

Answers from students in other courses will be ignored :-)

One can facilitate a neural network to generalize better by controlling its capacity. Mention any two methods to control a neural network's capacity.

Solution hint: Any 2 of the following methods:

- Dropout.
- Weight decay.
- Depthwise separable convolutions.
- Reduce width.

(Continued on page 10.)

- Reduce depth.
- Smaller kernel sizes.

(Continued on page 11.)

Exercise 8

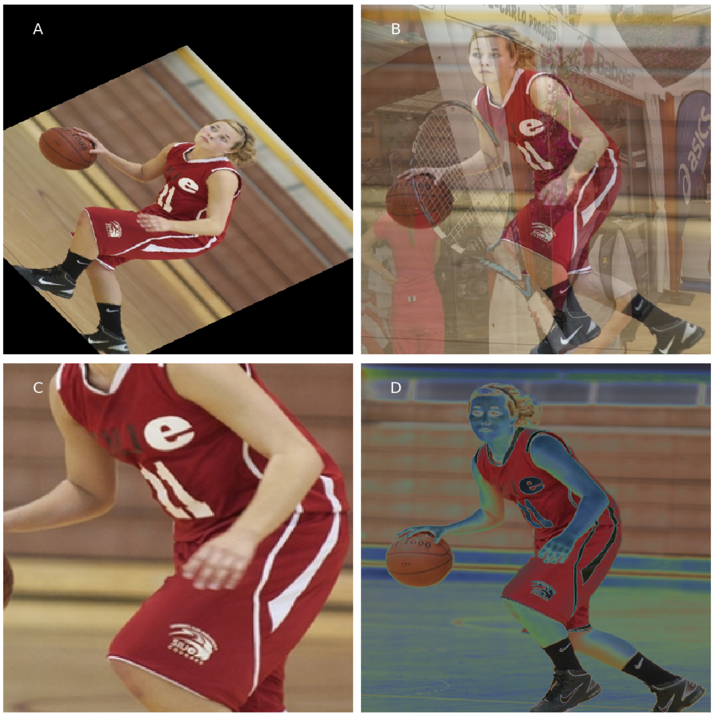


Figure 1: Various Image Augmentations

Figure 1 illustrates the following image transformations:

- Mixup.
- Solarization.
- Affine transformation.
- Crop.

8a

Match the names of the transformations with the corresponding image.

Solution hint:

- **Mixup** - B
- **Solarization** - D
- **Affine transformation** - A
- **Crop** - C

(Continued on page 12.)

8b

Classify each transformation as either **geometric**, **photometric**, or **other**. Provide a brief justification for your classification.

Solution hint:

- **Mixup**: Other, linearly interpolates between two different images and, importantly, *also blends the labels*.
- **Solarization**: Photometric, inverts the colors in an image based on a threshold, changing the pixel values but leaving the spatial arrangement of the pixels unchanged.
- **Affine transformation**: Geometric, an affine transformation includes scaling, translation, rotation, and shearing, all of which alter the spatial arrangement of the pixels.
- **Crop**: Geometric, the spatial dimensions of the image are altered, in this case, a portion is cropped.

(Continued on page 13.)

Exercise 9

9a

Describe two methods used to overcome the exploding gradients problem.

Solution hint: Gradients can be clipped if they explode. Two gradient clipping methods are:

- Clipping by value: If a gradient is greater than a threshold, then its value is set to be equal to the threshold.
- Clipping by norm: If the L2-norm of a vector of gradients is greater than a threshold, then each gradient in the vector is multiplied by the threshold and divided by the L2-norm of the vector.

9b

Describe any two input-output structures of RNNs. Give an example/application (one for each input-output structure) of where they can be used.

Solution hint: Any 2 of the following:

- One to one: RNN receives one input and emits one output. E.g., Image classification. NOTE: this is not a desirable answer to the question but since it is mentioned in the slides, it is acceptable.
- One to many: RNN receives one input and gives multiple outputs. E.g., Image captioning.
- Many to one: RNN receives a sequence of inputs and gives a output. E.g., sentence classification.
- Many to many: RNN receives a sequence as input and gives an output for each element in the input sequence. E.g., part of speech (POS) tagging.
- Many to many (encoder-decoder): RNN receives a sequence as input, encodes it and then returns a sequence as output. The length of the output can be different from that of the input. E.g., translating one language to another language.

(Continued on page 14.)

Exercise 10

Adversarial attacks can broadly be classified as:

- white box vs. black box attacks,
- targeted vs. untargeted attacks.

10a

Describe what characterizes these attack types.

Solution hint:

- **White box attacks**: Assumes attacker has complete access to the model, including its architecture, inputs, outputs, and weights when crafting adversarial examples.
- **Black box attacks**: Assumes attacker has no knowledge of the model's architecture or weights, and only has access to the inputs and outputs of the model.
- **Targeted attacks**: Attacks where the adversary not only wants the model to make a mistake, but also aims to manipulate the specifics of erroneous outputs.
- **Untargeted attacks**: Attacks where the adversary only wants the model to make a mistake, but does not care what the incorrect output is.

10b

Briefly explain the Projected Gradient Descent (PGD) method for constructing adverserial attacks, and discuss the applicability of PGD for the aforementioned types of attacks.

Solution hint:

- Projected Gradient Descent (PGD) is a gradient based method which constrains the perturbation to be within a predefined limit. Broadly speaking, this is achieved by projecting the gradient in a ball given by some predefined Lebesgue p-norm.
- PGD can be applied in both targeted and untargeted attacks, but is typically associated with white-box attacks where the attacker has full access to the model, as it requires model gradients.
- In black-box settings, applying PGD is challenging due to the absence of direct access to gradients. Techniques like transferability with surrogate models can be used with reduced effectiveness.

(Continued on page 15.)

Exercise 11

11a

Why do CNN architectures like single-shot multibox detector (SSD) and Feature Pyramid Network (FPN) use many features maps from many different layers of the neural network to detect object of different sizes?

Solution hint: They do so because the layers closer to the image have small receptive field and can be used to detect small objects while the deeper layers have larger receptive fields and can be used to detect bigger objects.

11b

Briefly explain how non-maximum suppression (NMS) tries to solve the problem of an object detector predicting multiple overlapping boxes for an object. (There is no need to write the NMS algorithm in detail.)

Solution hint: NMS tries to solve the problem by eliminating all but one box for every object in the image. It does so by repetitively selecting a box with highest score(confidence) and removing all the boxes that overlap with the selected box.

(Continued on page 16.)

Exercise 12

12a

Given an input matrix of size 3x2 (3 rows and 2 columns), what would be the dimensions of the output when applying transposed convolution with kernel size 2x2, stride=2 and no padding?

Solution hint: 6x4

12b

What is the difference between semantic segmentation and instance segmentation?

Solution hint: In semantic segmentation, we DO NOT differentiate between multiple instances of the same class while in instance segmentation we do. (Not required as part of the answer: In semantic segmentation, we classify every pixel in the image while in instance segmentation we do not need to classify every single pixel)

(Continued on page 17.)

Exercise 13

13a IN3310 and IN4310 students only

Answers from students in other courses will be ignored :-)

How does a transformer decoder block use the transformer encoder's output?

Solution hint: The decoder block uses the encoder's output as key and value in a multihead attention in it. (Not required as part of the answer: The query comes from the previous layer in the decoder block.)

13b IN4310 students only

Answers from students in other courses will be ignored :-)

What problem would we face if we were to use Transformers directly on image pixels by treating each pixel as a separate token, i.e., by treating an image as a sequence of pixels? How does the Vision Transformer (ViT) overcome that problem?

Solution hint: It is too expensive (both computationally and memory-wise) to run transformer by treating each pixel as a separate token as the self-attention layer computes N^2 dot-products for an input of length N. ViT overcomes this problem by dividing the image into patches and treating a whole patch as a single token.

(Continued on page 18.)

Exercise 14

14a IN5400 and IN9400 students only

Answers from students in other courses will be ignored :-)

GANs can overfit to data especially when the dataset is small enough. One way to solve that problem is to use data augmentation. But data augmentation can lead to a problem. What is that problem and how can we overcome that problem?

Solution hint: The problem is that the augmentation can leak into the generator's output, i.e., the generator's output can start looking like an augmented image. To solve this problem, the augmentation is added only with a probability p. The augmentation does not leak into the output if p is small enough.

14b IN9400 students only

Answers from students in other courses will be ignored :-)

In progressive growing of GANs, how is each new convolution layer incorporated (phased in) slowly?

Solution hint: The output of the new conv layer is multiplied by a factor alpha and added to its input. The value of alpha is low in the beginning and is slowly increased to phase in the new layer slowly.