UNIVERSITY OF OSLO

Faculty of mathematics and natural sciences

Exam in:	IN5400/IN9400 — Machine Learning for Image Analysis
Day of examination:	June 14, 2022
Examination hours:	9:00 – 13:00
This exercise set consists of 19 pages.	
Appendices:	None
Permitted aids:	All

(Continued on page 2.)

It is important that you read the following two (2) cover pages carefully before you start.

General information:

- Read the entire exercise text before you start solving the exercises. Please check that the exam paper is complete.
- Important messages during the exam are given directly from the course teachers on the course's semester page. It is therefore important that you check the course's semester page regularly during the exam.
- Your answer should reflect your own independent work and should be a result of your own learning and work effort.
- All sources of information are allowed. If you reproduce a text from books, online articles, etc., a reference to these sources must be provided to avoid suspicions of plagiarism. This also applies if a text is translated from other languages.
- You are responsible for ensuring that your exam answers are not available to others during the exam period, neither physically nor digitally.
- Remember that your exam answers must be anonymous; do not state either your name or that of fellow students.
- If you want to withdraw from the exam, press the menu at the top right of Inspera and select "Withdraw".

Collaboration during the exam:

It is not allowed to collaborate or communicate with others during the exam. Cooperation and communication will be considered as attempted cheating. A plagiarism control is performed on all submitted exams where text similarities between answers are checked. If you use notes that have been prepared in collaboration with others before the exam, this might be detected in a plagiarism control. Such text similarities will be considered an attempt at cheating.

Cheating / Dishonesty / Unfair advantage during an examination

Read about what is considered cheating on UiO's website: `https://www.uio.no/english/studies/examinations/cheating/index.html`

(Continued on page 3.)

Further notes:

- File upload: You have been given 30 min. extra time for uploading files (e.g. digital hand drawings). Please see also `https://www.uio.no/english/studies/examinations/submissions/options-for-hand-drawings.html`
- You can mix handwriting, hand drawings, digital drawings, and digitally typed documents. Handwritten drawings are okay. Handwritten text on drawings is okay. Digitally typed text is preferred over handwritten text because of less potential for misreading.
- You are responsible for the readability of your handwriting. You are responsible for ensuring that the quality of scans of any drawings or handwritten parts are sufficient to be easily readable. Remember that you can scan also using mobile phones.
- 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.
- **Submit one PDF file in Inspera**. In this file, the exercises should be answered in given order. Any figures or drawings MUST be included at the correct place in the document. You can import scanned drawings into some word processing software with typed text and export as PDF or use e.g. LaTeX. You can concatenate many PDFs into one using Adobe Acrobat Pro available to students and staff at UiO: `https://www.uio.no/english/services/it/home-away/kiosk/`
- 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.

Contact information:

User support for the exam: `https://www.mn.uio.no/english/studies/exam/user-support.html`

(Continued on page 4.)

Exercise 1

1a

Assume you want to predict the number of people in a country supporting that the country is or becomes a member of the European Union. Denote this true number y and the prediction $\hat{y}$. As inputs you have three values; the population count x_1, the country's land area x_2, and the gross domestic product (GDP) per capita x_3. Make an illustration of a neural network depicting the following relation between the inputs and the prediction:

$$\hat{y} = \sum_{k=1}^{4} w_k g\left(\sum_{j=1}^{3} v_{j,k} x_j + b_k\right) + c$$

$v_{j,k}$, w_k, b_k, and c are trainable parameters and g is the logistic sigmoid function, i.e.:

$$g(x) = \frac{1}{\exp(-x) + 1}$$

Solution hint: A sketch of a neural network with one hidden layer containing four nodes.

1b

Without changing the number of trainable parameters, how would you change the network so that it outputs a value between 0 and 1 indicating the proportion of people (instead of the number of people) in a country supporting that the country is or becomes a member of the European Union?

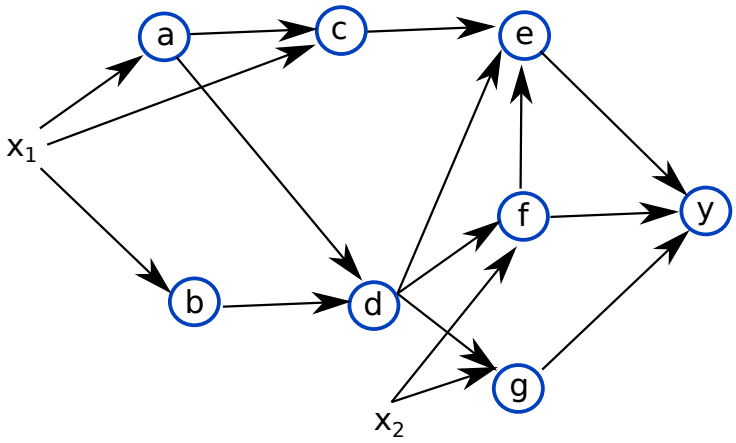
Solution hint: By defining the network's output as:

$$\hat{z} = g\left(\hat{y}\right) = g\left(\sum_{k=1}^{4} w_k g\left(\sum_{j=1}^{3} v_{j,k} x_j + b_k\right) + c\right)$$

(Continued on page 5.)

Exercise 2

Consider the following neural network:



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

$$n\left(s + \sum_l v_l x_l + \sum_k w_k k\right) \tag{1}$$

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:
  - $\frac{\partial e}{\partial k}$, where k is a neuron directly connected to the loss function e,
  - $\frac{\partial k_1}{\partial k_2}$, where k_2 is direct input to k_1, and
  - $\frac{\partial k}{\partial x_l}$, where x_l is input to the network.
- You **do not need** to plug in how $\frac{\partial k_1}{\partial k_2}$ or $\frac{\partial k}{\partial x_l}$ 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:

2a

$\frac{dy}{dx_2}$

(Continued on page 6.)

Solution hint:

$$\begin{aligned}\frac{dy}{dx_2} &= \frac{\partial y}{\partial e}\frac{de}{dx_2} + \frac{\partial y}{\partial f}\frac{df}{dx_2} + \frac{\partial y}{\partial g}\frac{dg}{dx_2} \\ &= \frac{\partial y}{\partial e}\frac{\partial e}{\partial f}\frac{df}{dx_2} + \frac{\partial y}{\partial f}\frac{\partial f}{\partial x_2} + \frac{\partial y}{\partial g}\frac{\partial g}{\partial x_2} \\ &= \frac{\partial y}{\partial e}\frac{\partial e}{\partial f}\frac{\partial f}{\partial x_2} + \frac{\partial y}{\partial f}\frac{\partial f}{\partial x_2} + \frac{\partial y}{\partial g}\frac{\partial g}{\partial x_2}\end{aligned}$$

2b

$\frac{dy}{dx_1}$

Solution hint:

$$\begin{aligned}\frac{dy}{dx_1} &= \frac{\partial y}{\partial e}\frac{de}{dx_1} + \frac{\partial y}{\partial f}\frac{df}{dx_1} + \frac{\partial y}{\partial g}\frac{dg}{dx_1} \\ &= \frac{\partial y}{\partial e}\frac{\partial e}{\partial c}\frac{dc}{dx_1} + \frac{\partial y}{\partial e}\frac{\partial e}{\partial f}\frac{df}{dx_1} + \frac{\partial y}{\partial e}\frac{\partial e}{\partial d}\frac{dd}{dx_1} + \frac{\partial y}{\partial f}\frac{df}{dx_1} + \frac{\partial y}{\partial g}\frac{dg}{dx_1} \\ &= \frac{\partial y}{\partial e}\frac{\partial e}{\partial c}\frac{dc}{dx_1} + (\frac{\partial y}{\partial e}\frac{\partial e}{\partial f} + \frac{\partial y}{\partial f})\frac{df}{dx_1} + \frac{\partial y}{\partial e}\frac{\partial e}{\partial d}\frac{dd}{dx_1} + \frac{\partial y}{\partial g}\frac{dg}{dx_1} \\ &= \frac{\partial y}{\partial e}\frac{\partial e}{\partial c}\frac{dc}{dx_1} + (\frac{\partial y}{\partial e}\frac{\partial e}{\partial f} + \frac{\partial y}{\partial f})\frac{\partial f}{\partial d}\frac{dd}{dx_1} + \frac{\partial y}{\partial e}\frac{\partial e}{\partial d}\frac{dd}{dx_1} + \frac{\partial y}{\partial g}\frac{\partial g}{\partial d}\frac{dd}{dx_1} \\ &= \frac{\partial y}{\partial e}\frac{\partial e}{\partial c}\frac{dc}{dx_1} + ((\frac{\partial y}{\partial e}\frac{\partial e}{\partial f} + \frac{\partial y}{\partial f})\frac{\partial f}{\partial d} + \frac{\partial y}{\partial e}\frac{\partial e}{\partial d} + \frac{\partial y}{\partial g}\frac{\partial g}{\partial d})\frac{dd}{dx_1} \\ &= \frac{\partial y}{\partial e}\frac{\partial e}{\partial c}\frac{dc}{dx_1} + ((\frac{\partial y}{\partial e}\frac{\partial e}{\partial f} + \frac{\partial y}{\partial f})\frac{\partial f}{\partial d} + \frac{\partial y}{\partial e}\frac{\partial e}{\partial d} + \frac{\partial y}{\partial g}\frac{\partial g}{\partial d})(\frac{\partial d}{\partial a}\frac{da}{dx_1} + \frac{\partial d}{\partial b}\frac{db}{dx_1}) \\ &= \frac{\partial y}{\partial e}\frac{\partial e}{\partial c}\frac{dc}{dx_1} + ((\frac{\partial y}{\partial e}\frac{\partial e}{\partial f} + \frac{\partial y}{\partial f})\frac{\partial f}{\partial d} + \frac{\partial y}{\partial e}\frac{\partial e}{\partial d} + \frac{\partial y}{\partial g}\frac{\partial g}{\partial d})(\frac{\partial d}{\partial a}\frac{\partial a}{\partial x_1} + \frac{\partial d}{\partial b}\frac{\partial b}{\partial x_1}) \\ &= \frac{\partial y}{\partial e}\frac{\partial e}{\partial c}(\frac{\partial c}{\partial x_1} + \frac{\partial c}{\partial a}\frac{da}{dx_1}) + ((\frac{\partial y}{\partial e}\frac{\partial e}{\partial f} + \frac{\partial y}{\partial f})\frac{\partial f}{\partial d} + \frac{\partial y}{\partial e}\frac{\partial e}{\partial d} + \frac{\partial y}{\partial g}\frac{\partial g}{\partial d})(\frac{\partial d}{\partial a}\frac{\partial a}{\partial x_1} + \frac{\partial d}{\partial b}\frac{\partial b}{\partial x_1})\end{aligned}$$

(Continued on page 7.)

Exercise 3

3a

The input to a pooling operation is:

1	7	1	1	4	0
0	2	1	7	5	2
6	5	3	6	5	2
6	5	2	3	0	2
5	0	2	0	4	5
3	2	1	2	7	5

and the output is:

7	7	5
6	6	5
5	2	7

Which pooling operation is it and what are the applied parameter values?

Solution hint: Max pooling with 2x2 filter and stride 2.

3b

A 2-dimensional convolutional layer applies filters with spatial size $(5, 9)$, stride 2, and standard padding and provides an output with spatial size $(30, 20)$. What is the upper and lower bound for the spatial size of the input to this layer?

Solution hint: If m is the spatial size of the input and n is the spatial size of the output of a convolutional layer that applies stride s and standard padding, then their relation can be described as:

$$n = \text{floor}\left(\frac{m-1}{s} + 1\right)$$

With $n_1 = 30$ and $s = 2$, we get:

$$30 = \text{floor}\left(\frac{m_1 - 1}{2} + 1\right)$$

The equation $29 = \text{floor}\left(\frac{m_1-1}{2}\right)$ implies that $m_1 = 59$ or $m_1 = 60$ if requiring integer m_1. Similarly, for $n_2 = 20$ and $s = 2$, we get:

$$20 = \text{floor}\left(\frac{m_2 - 1}{2} + 1\right)$$

(Continued on page 8.)

The equation $19 = \text{floor}\left(\frac{m_2-1}{2}\right)$ implies that $m_2 = 39$ or $m_2 = 40$ if requiring integer m_2.

3c

Assume the 2-dimensional convolutional layer (applying filters with spatial size $(5, 9)$, stride 2, and standard padding) has 200 input channels and 100 output channels. What is the number of trainable parameters in this layer?

Solution hint: Each filter has $200 * 5 * 9 + 1 = 9001$ parameters. There is one filter for each output channel, thus 100 filters with 9001 parameters each, giving a total of 900100 parameters in this layer.

3d

A second 2-dimensional convolutional layer is applied on top of the 2-dimensional convolutional layer applying filters with spatial size $(5, 9)$, stride 2, and standard padding. The second 2-dimensional convolutional layer applies filters with spatial size $(3, 3)$, stride 1, and no padding and has 100 input channels and 50 output channels. On top of this second 2-dimensional convolutional layer is a third 2-dimensional convolutional layer applying filters with spatial size $(7, 7)$, stride 3, and zero-padding 2 and having 50 input channels and 1 output channel.

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 can assume that the input is sufficiently large and that the neuron is located in the spatial center.

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

Solution hint: The height and width of the region used by layer $k \in \{1, 2, 3\}$ can be computed recursively using:

$$R_k = R_{k-1} + (F_k - 1) \prod_{i=1}^{k-1} S_i$$

with $R_0 = 1$. Relevant characteristics of the layers are:

$$F_{1,h} = 5, F_{1,w} = 9, S_1 = 2$$
$$F_{2,h} = 3, F_{2,w} = 3, S_2 = 1$$
$$F_{3,h} = 7, F_{3,w} = 7, S_3 = 3$$

(Continued on page 9.)

Inserting the height characteristics into the general equation gives:

$$R_{1,h} = 1 + (5-1) * 1 = 5$$
$$R_{2,h} = 5 + (3-1) * 2 = 9$$
$$R_{3,h} = 9 + (7-1) * 2 = 21$$

Inserting the width characteristics into the general equation gives:

$$R_{1,w} = 1 + (9-1) * 1 = 9$$
$$R_{2,w} = 9 + (3-1) * 2 = 13$$
$$R_{3,w} = 13 + (7-1) * 2 = 25$$

Thus, a $21x25$ region of the input to the first layer will be used to compute the output of a single neuron in the third layer.

Exercise 4

4a

Modify the following code so that it applies batch normalization.

```
class Model(nn.Module):
  def __init__(self):
    super(Model, self).__init__()
    self.conv1 = nn.Conv2d(in_channels=1,  out_channels=16, kernel_size=3)
    self.conv2 = nn.Conv2d(in_channels=16, out_channels=16, kernel_size=3)
    self.maxpool1 = nn.MaxPool2d(kernel_size=2)
    self.conv3 = nn.Conv2d(in_channels=16, out_channels=32, kernel_size=3)
    self.conv4 = nn.Conv2d(in_channels=32, out_channels=32, kernel_size=3)
    self.maxpool2 = nn.MaxPool2d(kernel_size=2)
    self.fc1 = nn.Linear(in_features=32*4*4, out_features=10)

  def forward(self, x):
    out = F.relu(self.conv1(out))
    out = F.relu(self.conv2(out))
    out = self.maxpool1(out)
    out = F.relu(self.conv3(out))
    out = F.relu(self.conv4(out))
    out = self.maxpool2(out)
    out = out.view(out.size(0), -1)
    out = self.fc1(out)
    return out
```

You can but do not need to include unchanged code in your answer. If you do not include unchanged code, you should make sure that it is clear

(Continued on page 10.)

where code lines you propose adding should be included and which lines you propose changing or deleting.

Solution hint: Could e.g. define and apply an `nn.BatchNorm2d` layer after the last `nn.Conv2d` layer, i.e. change to:

```
class Model(nn.Module):
  def __init__(self):
    super(Model, self).__init__()
    self.conv1 = nn.Conv2d(in_channels=1,  out_channels=16, kernel_size=3)
    self.conv2 = nn.Conv2d(in_channels=16, out_channels=16, kernel_size=3)
    self.maxpool1 = nn.MaxPool2d(kernel_size=2)
    self.conv3 = nn.Conv2d(in_channels=16, out_channels=32, kernel_size=3)
    self.conv4 = nn.Conv2d(in_channels=32, out_channels=32, kernel_size=3)
    self.conv4_bn = nn.BatchNorm2d(32)
    self.maxpool2 = nn.MaxPool2d(kernel_size=2)
    self.fc1 = nn.Linear(in_features=32*4*4, out_features=10)

  def forward(self, x):
    out = F.relu(self.conv1(out))
    out = F.relu(self.conv2(out))
    out = self.maxpool1(out)
    out = F.relu(self.conv3(out))
    out = F.relu(self.conv4_bn(self.conv4(out)))
    out = self.maxpool2(out)
    out = out.view(out.size(0), -1)
    out = self.fc1(out)
    return out
```

4b

Assume the following code should include everything necessary to be applicable for training a model for one epoch (if `is_training` is `True`) and evaluating a validation or test set (if `is_training` is `False`). You can assume that the user have defined `data_loader` correctly.

```
total_loss = 0
for batch_idx, data_batch in enumerate(data_loader):
  images = data_batch[0].to('cuda')
  labels = data_batch[1].to('cuda')
  if not is_training:
    with torch.no_grad():
      prediction = model(images)
      loss = loss_fn(prediction, labels)
      total_loss += loss.item()
  elif is_training:
    prediction = model(images)
```

(Continued on page 11.)

```
        loss = loss_fn(prediction, labels)
        total_loss += loss.item()
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
    loss_avg = total_loss / len(data_loader)
```

If the model applies batch normalization, what needs to be included in the code so that the batch normalization will work as intended? Why is this needed?

Solution hint: The model needs to be set to training or evaluation mode depending on the value of `is_training`, e.g. by including `model.train(is_training)` before the `for` loop or alternatively adding:

```
if is_training:
  model.train()
else:
  model.eval()
```

The model needs to be set to evaluation mode at validation and testing time to avoid the running estimates of mean and variance in the batch normalization layer to be updated. These estimates should be updated when training, thus the model needs to be in training mode then.

4c

Batch normalization and related normalization techniques such as group, instance, or layer normalization are frequently used to ensure good gradient flow even for deep networks. Briefly suggest two techniques that facilitate networks without any such normalization to perform similarly as state-of-the-art networks using batch normalization.

Solution hint: Two of the following techniques could be suggested:

- Include skip connections and modify the standard residual branch $f_l(x_l)$ to $\alpha f_l(x_l/\beta_l)$ (α and β_l are trainable parameters) in order to allow rescaled variance.
- Use scaled weight standardization.
- Bound the gradient norm by including adaptive gradient clipping.

Exercise 5

5a

What are the potential advantages of using a momentum term when running stochastic gradient descent (SGD) to learn parameters?

(Continued on page 12.)

Solution hint: The momentum term when applied to SGD is a memory of gradients from the past. It can speed up the motion through parameter space in flat valley, e.g. of a part which is similar to the surface of a quadratic function, because it remembers the more steep gradients at the entrance of the valley.

5b

How would you change the value of the momentum parameter (default is frequently 0.9) in order to put more weight on the more recently computed gradients (from more recent iterations)? Explain in at most two sentences why.

Solution hint: Increase it to values above 0.9.

Exercise 6

Consider these three vectors, which are three different gradients g_t from three timesteps $t = 0, 1, 2$, obtained from some loss with respect to two input variables:

$$g_0 = (-6, 4),\ g_1 = (2, \sqrt{6}),\ g_2 = (-\sqrt{40}, \sqrt{14})$$

For this task, consider an exponential moving average (EMA) with a parameter α over a sequence of values $(g_0, g_1, g_2, \ldots)$ as given below:

$$\begin{aligned} e_0 &= (1-\alpha)g_0 \\ e_1 &= \alpha e_0 + (1-\alpha)g_1 \\ e_k &= \alpha e_{k-1} + (1-\alpha)g_k \end{aligned}$$

6a

Calculate the whole gradient update resulting from g_2 which is used in RMSprop. Use $\alpha = 0.5$ as the EMA parameter and $\epsilon = 6e-3$ as the stabilizing epsilon for the EMAs.

Solution hint: Let e_2 be the EMA of the sequence $(\|g_0\|, \|g_1\|, \|g_2\|)$:

$$\begin{aligned} e_2 &= 0.5(0.25\|g_0\|^2 + 0.5\|g_1\|^2) + 0.5\|g_2\|^2 \\ &= 0.125\|g_0\|^2 + 0.25\|g_1\|^2 + 0.5\|g_2\|^2 \\ &= (4.5 + 2) + (1 + 1.5) + (20 + 7) \\ &= 36 \end{aligned}$$

(Continued on page 13.)

Then the gradient update which is used in RMSprop is:

$$\frac{g_2}{\sqrt{e_2+\epsilon}} = \frac{(-\sqrt{40}, \sqrt{14})}{\sqrt{36+0.006}} \approx \frac{(-\sqrt{40}, \sqrt{14})}{6.0005}$$

Exercise 7

7a

Let f be a logit classifier output, x the original input sample, and x^* be an adversarial sample created from x.

Which of the following could be described as a performance metric for adversarial attacks for classification? Explain in at most 5 sentences, why your chosen one is suitable, and why the others are not.

- $\frac{\|f(x)-f(x^*)\|^2}{\|x-x^*\|^2} \to 0$
- $\frac{\|f(x)-f(x^*)\|^2}{\|x-x^*\|^2} \to \infty$
- $\frac{(f(x)\cdot f(x^*))^2}{\|x-x^*\|^2} \to 1$

Solution hint: It is the second, because we want the difference between the logits $\|f(x) - f(x^*)\|^2$ to be large, while the samples in input space should be close, thus $\|x - x^*\|^2 \to 0$. The first does exactly the opposite, it can be minimized when the predictions are equal. The third is satisfied when the inner product of logits is close to the sample distance, which has no adversarial meaning.

7b

The following code tries to create an adversarial attack. Explain in at most two sentences what the following code does wrong. Explain in at most two sentences what should have been done instead.

```
def advattack():

  stepsize = 0.0001

  model  = getmodel().to(torch.device('cpu'))

  loader = getloader()
  #gets a single image
  sampleindex = 0
```

(Continued on page 14.)

```
targetimage = loader[sampleindex][0].unsqueeze(0).to(torch.device('cpu'))

targetimage.requires_grad = True

with torch.no_grad():
  y = model(targetimage)
  maxind = torch.argmax(y[0,:])

done = False
stepcounter = 100
curstep = 0
while False == done:

  with torch.enable_grad():
    y = model(targetimage)

  torch.sum(y[0,maxind]).backward()

  with torch.no_grad():
    for nm, param in model.named_parameters():
      param.data -= stepsize*param.grad.data
      print( nm, 'norm of gradient:', torch.norm(param.grad.data).item() )

  newmaxind = torch.argmax(y[0,:])
  if newmaxind != maxind:
    done = True
    print('converged at step', curstep)

  curstep +=1
  if curstep >= stepcounter:
    done = True
    print('not converged, terminated after max number of steps reached')
```

Solution hint: The code optimizes the model parameters until it mispredicts but should have optimized the `targetimage`.

Exercise 8

You are working for security firm that wants to have a model that detects humans in a live closed-circuit television (CCTV) footage. Your training data consists of images with ground truth labels for a) coordinates of boxes around humans and b) a binary label (human or not-human) for each pixel.

(Continued on page 15.)

8a

You decide to start experimenting with a two-stage object detection model that makes use of anchor boxes. Initially, you are only required to detect humans who appear to be standing/walking in an image. What shape (a rough height to width ratio) will you choose for anchor boxes? Explain briefly.

Solution hint: The anchor boxes should be rectangular with height greater than width as humans are more tall than they are wide. The height to width ratio could be 4:1 for instance.

8b

Humans closer to the camera would look bigger than the ones far away. How will you make modifications to the neural network architecture to tackle this problem without using anchor boxes of varying sizes?

Solution hint: Possible solutions:

- Single-Shot multibox Detector (SSD). Add some layers with decreasing resolution at the end of the network. Predict boxes for each of those layers separately.
- Feature Pyramid Network (FPN). Upsample the last low-resolution layer multiple times in stages. At each stage, add the output of earlier layers in the network (ideally after processing the output a bit). Make prediction at each stage.

8c

You are now required to also produce a mask for each human detected. How will you modify the architecture of your network to incorporate this new requirement?

Solution hint: Add a branch (a subnetwork) in the network towards the end to predict the mask of the human as done in Mask R-CNN.

Exercise 9

9a

Suppose you obtain a plot of clusters using t-SNE. It looks plausible to you. Similarly looking images are close together in the plot.

(Continued on page 16.)

source: Andrej Karpathy, the 1k plot at
`https://cs.stanford.edu/people/karpathy/cnnembed/`

Why is it suggested as good practice to validate the findings in a way different from the t-SNE embedding itself? Three sentences at most please.

Solution hint:
One answer direction would be that the embeddings produced are sensitive to parameters used, and thus can vary highly. Therefore, one should check that one has not simply selected the one parameter which matches ones own expectations.

Another answer direction would be that it is a low-dimensional embedding of high dimensional features and therefore the distances in 2D are an approximation of the distances in higher dimensions. As approximation they might not be fully representative of the distances in the original space.

9b

How to complete this code in order to obtain a tensor called `rscores` which contains a gradient times input explanation? Hint: A couple of lines should be sufficient.

```
def gxiexplanation():

  model  = getmodel().to(torch.device('cpu'))

  loader = getloader()
```

(Continued on page 17.)

```
#gets a single image
sampleindex = 0
targetimage = loader[sampleindex][0].unsqueeze(0).to(torch.device('cpu'))

targetimage.requires_grad = True

with torch.enable_grad():
  y = model(targetimage)
  maxind = torch.argmax(y[0,:])

#TODO to be continued by students HERE
```

Solution hint: Add the following lines at the end:

```
torch.sum(y[0,maxind]).backward()
rscores = targetimage.grad * targetimage
```

Exercise 10

You are working on an application to detect the direction in which eyes are looking. Your application receives an image of eyes as input, and it outputs an angle (0 to 360). You have data for some of the angles but lack it for others. Since it is difficult to collect data for all possible angles, you decide to generate images of eyes synthetically and then train a Generative Adversarial Network (GAN) to convert synthetic images into realistic ones.

10a

What will be the inputs and outputs of the generator network and the discriminator network in your GAN?

Solution hint:

- Generator's input: a synthetic image of eyes.
- Generator's output: the realistic version of the synthetic image provided as input.
- Discriminator's input: either an image generated by the generator or an image from the dataset.
- Discriminator's output: a score between 0 and 1 indicating the probability of the image being from the dataset (output close to 1) or from the generator (output close to 0).

(Continued on page 18.)

10b

You decide to use a generator network that has an architecture containing two main components and use transposed convolutions (i.e. fractionally-strided convolutions) in one of them. What are these two main components and in which would you use transposed convolutions?

Solution hint: The generator should have an encoder-decoder architecture where encoder encodes the input image to a latent vector and the decoder decodes the vector back to an image of the required resolution. Transposed convolutions should be used in the decoder to scale up the latent vector to the required resolution.

Exercise 11

Consider the space of all distributions $\{P : P(a) + P(b) = 0\}$
over a space with two elements $M = \{a, b\}$ with a metric (i.e. a distance function) $d(a, b) = 1$. Then, the 1-Wasserstein distance is defined as:

$$W(P,Q) = \inf_{R:R\geq 0,\ \sum_{c\in\{a,b\}} R(\cdot,c)=P(\cdot),\ \sum_{c\{a,b\}} R(c,\cdot)=Q(\cdot),\ \sum_{c,c'\in\{a,b\}} R(c,c')=1} \int_{M\times M} d(x,y)dR(x,y)$$

Because we are dealing with probabilities over a finite number of points, the integral can be rewritten as:

$$\int_{M\times M} d(x,y)dR(x,y) = \sum_{c,c'\in\{a,b\}} d(c,c')R(c,c')$$

11a

Rewrite the integral in the Wasserstein distance as a sum of terms depending on a, b, and $R(\cdot,\cdot)$ (and not $d(\cdot,\cdot)$). Hint: You might want to use some general property or axiom of metrics.

Solution hint:

$$\begin{aligned}\int_{M\times M} d(x,y)dR(x,y) &= d(a,a)R(a,a) + d(a,b)R(a,b) + d(b,a)R(b,a) + d(b,b)R(b,b)\\ &= d(a,b)R(a,b) + d(b,a)R(b,a)\\ &= R(a,b) + R(b,a)\end{aligned}$$

(Continued on page 19.)

Exercise 12 PhD students only

Answers from MSc students will be ignored :)

12a

Name one possible advantage of using a learning rate reduction scheme over training with a fixed learning rate.

Solution hint: A learning rate reduction scheme may lead to better prediction performance on validation and test data, in particular if a fixed learning rate causes the training loss to decrease too slowly.

12b

Define a learning rate reduction scheme by a formula and draw its learning rate as a function of number of epochs.

Solution hint: In step-wise learning rate decay, the learning rate is reduced by a constant multiplicative factor every K epochs:

$$\eta_t = \begin{cases} \eta_{t-1} & \text{if } t\%K \neq 0 \\ c\eta_{t-1} & \text{if } t\%K = 0,\ c \in (0,1) \end{cases}$$

$$\text{or } \eta_t = \eta_0 c^{\lfloor \frac{t}{K} \rfloor},\ c \in (0,1)$$

A drawing is omitted in this solution hint.