UNIVERSITEIT TWENTE.

Mathematics of Imaging & AI
Department of Applied Mathematics
Prof. Dr. Christoph Brune
Dr. Len Spek, Julian Suk, Dr. Nicolò Botteghi

Exam: Deep Learning - From Theory to Practice (201800177)

Date: Thursday, Jan 26, 2023
Time: 13:45-15:45

- The exam has 28 points in total (grading scheme below).
- An explanation to every answer is required. Answers can be short and focused.
- You can make use of a calculator. No other additional material is allowed.

Good luck and success!

Exercise 1. (Multiple Choice Questions) [5pt] For each of the following questions, provide the answer of your choice. There is only ONE correct choice per question. No explanation is required. There is no penalty for a wrong answer.

(a) **[1pt]** Which of the following activation functions can lead to vanishing gradients for large input values for a single layer neural network?

(i) ReLU
(ii) Tanh
(iii) Leaky ReLU
(iv) All of the above

(b) **[1pt]** After training a neural network, you observe a large gap between the training accuracy (100%) and the test accuracy (42%). Which of the following methods is commonly used to reduce this gap?

(i) Diffusion Network
(ii) Dropout
(iii) RMSprop Optimizer
(iv) Parallelization using GPUs

(c) **[1pt]** Which of the following is a non-iterative method to generate adversarial examples to attack a neural network?

(i) Non-Saturating Cost Method
(ii) Input Optimization Method
(iii) Adversarial Training
(iv) Projected Gradient Descent
(v) Fast Gradient Sign Method

(d) **[1pt]** Which of the following would you consider to be a valid activation function to train a nonlinear neural network using gradient descent?

(i) $f(x) = 0.9x + 1$
(ii) $f(x) = \max(x, .1x)$

(iii) $f(x) = \begin{cases} 0 & | \, x \geq 0 \\ 1 & | \, x < 0 \end{cases}$

(e) **[1pt]** What is the purpose of a value function in reinforcement learning?

(i) A value function is used to determine the future rewards of a given state.

(ii) A value function is used to determine the future rewards of a given action in a given state.

(iii) A value function is used to determine the expected future rewards of a given policy.

(iv) A value function is used to determine the current rewards of a given state.

Exercise 2. (Neural Networks for Time Series) [10pt] We have time series data $x(t_1), \cdots, x(t_n) \in \mathbb{R}^d$ and want to learn a scalar feature $y \in \mathbb{R}$. There are many different architectures one could use.

One architecture is a recurrent neural network, with a hidden neuron $h \in \mathbb{R}^m$, given by

$$\begin{aligned} h_i &= \sigma(Wx(t_i) + Vh_{i-1}) \\ y &= Uh_n \end{aligned} \tag{1}$$

Here σ is some piecewise nonlinearity (for example ReLU).

(a) **[1pt]** Give the dimensions of the weight matrices W, V, U.

(b) **[2pt]** Write a recursive formula for the gradient of $\frac{dy}{dh_{i-1}}$, i.e. the right-hand side should contain the term $\frac{dy}{dh_i}$. Check that the dimensions match.

(c) **[2pt]** Suppose we train the network using weight decay, which makes the weights small. What can happen in this case to the gradient $\frac{dy}{dh_1}$ and why is it a problem?

(d) **[2pt]** We can change this recurrent network into a residual network, by adding skip connections. How does the formula (1) look like when we add skip connections? How does this change the gradient $\frac{dy}{dh_{i-1}}$ of question *(b)*?

(e) **[1pt]** Why does the problem mentioned in question *(c)* for RNNs, not occur in ResNets?

(f) **[2pt]** Suppose the data points $x(t)$ are sampled irregularly, i.e. $t_i - t_{i-1}$ is not constant. Why is this a problem for both the RNN and ResNet defined above? Name an architecture that is suitable for this type of data and explain why. Can you give another advantage of this architecture over RNNs and ResNets?

Exercise 3. (AutoEncoders, Generative Networks) [7pts]

(a) **[2pt]** Give a basic description of an autoencoder and explain why it can find a lower dimensional representation.

(b) **[2pt]** What does a variational autoencoder (VAE) add to this architecture class as key feature.

(c) **[2pt]** Recall that a GAN could, in principle, be trained using a min-max formulation, where G is the generator function, D is the probability the discriminator assigns to the sample being data, and $\mathcal{J}_D$ and $\mathcal{J}_G$ are the cost functions for the discriminator and generator, respectively.

$$\begin{aligned} \mathcal{J}_D &= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[-\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}}[-\log(1 - D(G(\mathbf{z})))] \\ \mathcal{J}_G &= -\mathcal{J}_D \\ &= \text{const} + \mathbb{E}_{\mathbf{z}}[\log(1 - D(G(\mathbf{z})))] \end{aligned}$$

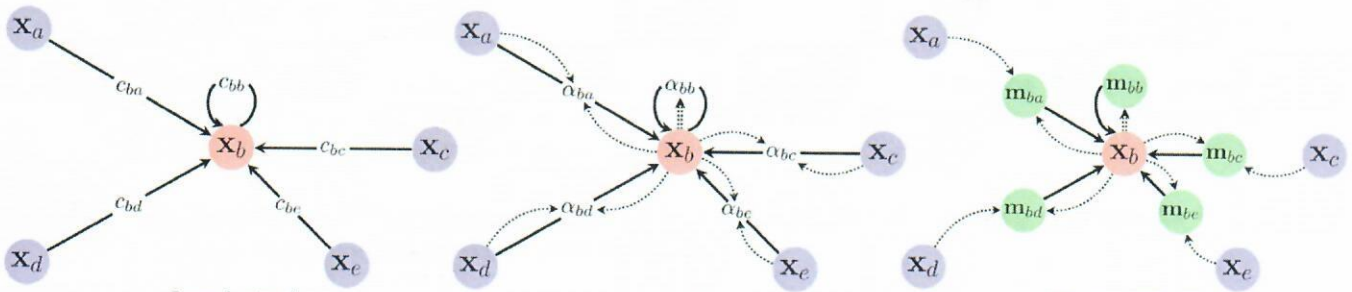
If we use those loss functions in a min-max optimization problem why does it help for the discriminator and the generator to solve their goals in this way.

(d) **[1pt]** In practice however, the generator in a GAN is usually trained with a different loss function, namely

$$-\log D(G(\mathbf{z}))$$

What is the reason to use this cost function rather than the one given above in (c)? (Hint: Consider what happens if the discriminator can easily recognize its samples as fake.)

Exercise 4. (Graph Neural Networks) [6pts] There are three main 'flavours' to construct a graph neural network layer, each of which balance the need to be expressive with having a scalable implementation.



Convolutional *Attentional* *Message-passing*

$$\mathbf{h}_i = \phi\left(\mathbf{x}_i, \bigoplus_{j\in\mathcal{N}_i} c_{ij}\psi(\mathbf{x}_j)\right) \qquad \mathbf{h}_i = \phi\left(\mathbf{x}_i, \bigoplus_{j\in\mathcal{N}_i} a(\mathbf{x}_i,\mathbf{x}_j)\psi(\mathbf{x}_j)\right) \qquad \mathbf{h}_i = \phi\left(\mathbf{x}_i, \bigoplus_{j\in\mathcal{N}_i} \psi(\mathbf{x}_i,\mathbf{x}_j)\right)$$

(a) **[2pt]** All three of these layers are permutation equivariant. What is permutation equivariance and what are the benefits for learning on graphs?

(b) **[2pt]** Which layer leads to the most scalable implementation of a graph neural network layer (in general)? Explain your reasoning.

(c) **[2pt]** Which layer is the most expressive, i.e. it can represent the most functions.

Grading scheme:

Ex 1.	(a) 1pt	**Ex 2.**	(a) 1pt	**Ex 3.**	(a) 2pt	**Ex 4.**	(a) 2pt
	(b) 1pt		(b) 2pt		(b) 2pt		(b) 2pt
	(c) 1pt		(c) 2pt		(c) 2pt		(b) 2pt
	(d) 1pt		(d) 2pt		(d) 1pt		
	(e) 1pt		(e) 1pt				
			(f) 2pt				

Total: 28 points

Extra Credit (but no extra points): Which question has been generated by ChatGPT?