UNIVERSITEIT TWENTE.

Applied Analysis Group
Department of Applied Mathematics
Dr. Christoph Brune
Len Spek

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

Date: Tuesday, Jan 21, 2020
Time: 13:45-16:45

- The exam has 40 points in total (grading scheme below).
- An explanation to every answer is required. Answers can be short and focussed.
- You can make use of a calculator. No other additional material is allowed.

Good luck and success!

Exercise 1. (Backpropagation, Optimization) [8pts]

(a) **[2pts]** *Describe shortly with words what backpropagation means? How is it related to the loss function of a neural network?*

(b) **[4pts]** Suppose we have a prediction problem where the target t corresponds to an angle, measured in radians. A reasonable loss function we might use is

$$L(y,t) = 1 - cos(y-t).$$

Suppose we make predictions using a linear model,

$$y = \mathbf{w}^T\mathbf{x} + b.$$

where $\mathbf{w}$ is a vector and b is a scalar. The cost is the average loss over the training set:

$$C = \frac{1}{N}\sum_{i=1}^{N} L(y^{(i)}, t^{(i)}).$$

Derive a sequence of vectorized mathematical expressions for the gradients of the cost with respect to $\mathbf{w}$ *and* b. The inputs are organized into a matrix $\mathbf{X}$ with one row per training example. Your answer should look like:

$$\begin{aligned}\mathbf{y} &= \cdots \\ \frac{dC}{d\mathbf{y}} &= \cdots \\ \frac{dC}{d\mathbf{w}} &= \cdots \\ \frac{dC}{db} &= \cdots\end{aligned}$$

You can use $sin(\mathbf{A})$ to denote the sin function applied elementwise to $\mathbf{A}$. Remember that $\frac{dC}{d\mathbf{w}}$ denotes the gradient vector,

$$\frac{dC}{d\mathbf{w}} = \begin{pmatrix} \frac{dC}{dw_1} \\ \vdots \\ \frac{dC}{dw_D} \end{pmatrix}$$

(c) **[2pts]** Suppose we are performing gradient descent on a quadratic objective:

$$J(\theta) = \frac{1}{2}\theta^T A\theta$$

The dynamics of the gradient descent with learning rate α can be analyzed in terms of the spectral decomposition $A = Q\Lambda Q^T$, where Q is an orthogonal matrix containing the eigenvectors of A, and $\Lambda = diag(\lambda_1, \cdots, \lambda_d)$ in ascending order, namely

$$\theta_t = Q(I - \alpha\Lambda)^t Q^T \theta_0.$$

Now, based on this formula, what is the value of C such that the gradient descent iterates diverge for $\alpha > C$ but converge for $\alpha < C$? Briefly justify your answer.

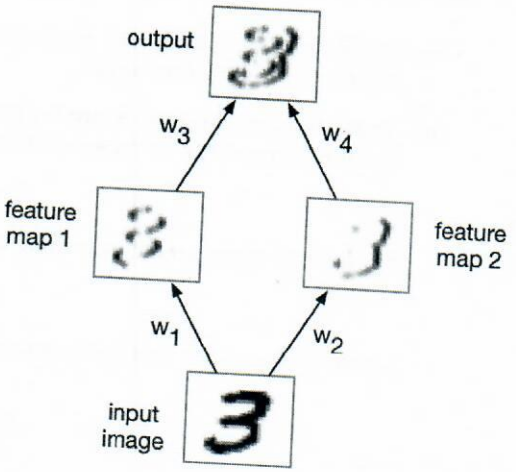
Exercise 2. (Convolutional Neural Networks, Depth, MLP) [9pts]

(a) **[2pts]** In this question, you will design a convolutional network to detect vertical boundaries in an image. The architecture of the network is as shown on the right.

The ReLU activation function is applied to the first convolution layer. The output layer uses the linear activation function.

You may assume a standard definition of convolution from the course with standard boundary conditions.

In order to make the figure printable for this exam paper, we use white to denote 0 and darker values to denote larger (more positive) values.



Design two convolution kernels for the first layer of size 3×3. One of them should detect dark/light boundaries, and the other should detect light/dark boundaries. (It doesn't matter which is which.) You don't need to justify your answer.

$w_1 =$ $\quad$ $w_2 =$

Design convolution kernels of size 3×3 for the output layer, which computes the desired output. You don't need to justify your answer.

$w_3 =$ $\quad$ $w_4 =$

(b) **[2pts]** Assume you have a CNN architecture with a convolution layer of size 5×5 on the one hand, and a 'deeper' CNN architecture with two convolution layers of size 3×3 on the other hand. The convolutions are with no pooling. *Why are the two architectures similar?*

(c) **[3pts]** *What are the three key properties of CNNs? Please explain them shortly.*

(d) **[2pts]** *Assume you have a fully-connected Multi-Layer-Perceptron (MLP) architecture and a Convolutional Neural Network (CNN) architecture with the same input dimension and the same depth. What can you say about the scaling of the parameter space (i.e. amount of parameters) when you increase the input size in the limit to infinity?*

Exercise 3. (Regularisation in Video Classification) [7pts]

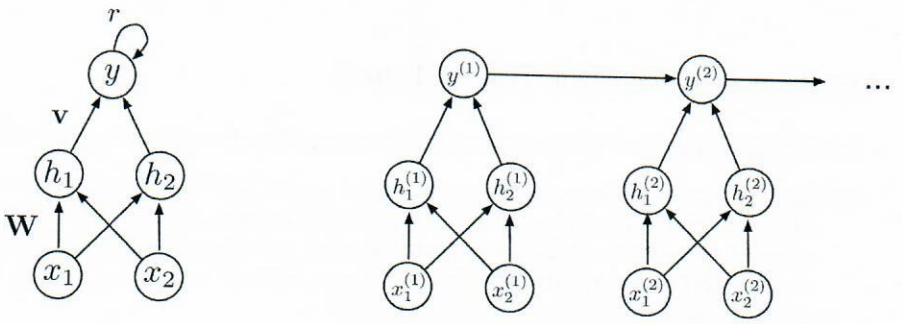
(a) **[2pts]** Assume you have a video classification problem given, e.g. classifying image sequences of 'cats' and 'dogs' into their corresponding categories. *Choose two deep learning architectures which could solve the problem and draw a rough sketch of your chosen architectures.*

(b) **[3pts]** *Mention three challenges where regularization would be needed or could help?*

(c) **[2pts]** *How could you explicitly apply regularization to your chosen deep learning architectures from part (a)?*

Exercise 4. (Recurrent Neural Networks) [6pts]

(a) **[5pts]** Suppose we receive two binary sequences $x_1 = (x_1^{(1)}, \cdots, x_1^{(T)})$ and $x_2 = (x_2^{(1)}, \cdots, x_2^{(T)})$ of equal length, and we would like to design an RNN to determine if they are identical. We will use the following architecture, drawn with self-loops on the left and as an unrolled architecture on the right:



The computation in each time step is as follows:

$$h^{(t)} = \phi(Wx^{(t)} + b)$$

$$y^{(t)} = \begin{cases} \phi(v^T h^{(t)} + r y^{(t-1)} + c) & \text{for } t > 1 \\ \phi(v^T h^{(t)} + c_0) & \text{for } t = 1, \end{cases}$$

where ϕ denotes the hard threshold activation function

$$\phi(z) = \begin{cases} 1 & \text{if } z > 0 \\ 0 & \text{if } z \leq 0\,. \end{cases}$$

The parameters are a 2×2 weight matrix W, a 2-dimensional bias vector b, a 2-dimensional weight vector v, a scalar recurrent weight r, a scalar bias c for all but the first time step, and a separate bias c_0 for the first time step.

We'll use the following strategy. We'll proceed one step at a time, and at time t, the binary-valued elements $x_1^{(t)}$ and $x_2^{(t)}$ will be fed as inputs. The output unit $y(t)$ at time t will compute whether all pairs of elements have matched up to time t. The two hidden units $h_1^{(t)}$ and $h_2^{(t)}$ will help determine if both inputs match at a given time step. *Hint: have $h_1^{(t)}$ determine if both inputs are 0, and $h_2^{(t)}$ if both inputs are 1.*

Give parameters which correctly implement this function,

$$W = \cdots$$
$$b = \cdots$$
$$v = \cdots$$
$$r = \cdots$$
$$c = \cdots$$
$$c_0 = \cdots$$

Exercise 5. (AutoEncoders, Generative Networks) [6pts]

(a) **[2pts]** *Sketch the architecture of an AutoEncoder network. What is the relationship between the Principal Component Analysis (PCA), based on the Singular Value Decomposition (SVD), and the square reconstruction error of an AutoEncoder?*

(b) **[2pts]** *In which situation would a Generative Adversarial Network (GAN) be added value compared to an AutoEncoder?*

(c) **[2pts]** *Assume you train a GAN based on the MNIST database of hand-written digits and suddenly obtain only generated images of one digit class. What is the problem and how could you solve it?*

Exercise 6. (Reinforcement Learning) [4pts]

(a) **[1pts]** *Sketch the diagram explaining the main idea of reinforcement learning.*

(b) **[2pts]** Suppose we have a Markov Decision Process (MDP) with two time steps. It has an initial state distribution $p(s_1)$, transition probabilities $p(s_{t+1}|s_t, a_t)$, and a deterministic reward function $r(s, a)$. The agent is currently following a stochastic policy $\pi_\theta(a|s)$ parameterized by θ. *Give the formula for the probability $p(\tau)$ of a 'rollout' $\tau = (s_1, a_1, s_2, a_2)$, i.e. the likelihood of obtaining this sequence of actions and states.*

(c) **[1pt]** Recall that the discounted return is defined as

$$G_t = \sum_{i=0}^{\infty} \gamma^i r_{t+i},$$

where γ is the discount factor and r_t is the reward at time t. *Give the definition of the action-value function $Q^\pi(s, a)$ for policy π, state s, and action a. You can either give an equation or explain it verbally.*

Grading scheme:

Ex 1.	(a) 2pt (b) 4pt (c) 2pt	**Ex 2.**	(a) 2pt (b) 2pt (c) 3pt (d) 2pt	**Ex 3.**	(a) 2pt (b) 3pt (c) 2pt	**Ex 4.**	(a) 6pt	**Ex 5.**	(a) 2pt (b) 2pt (c) 2pt	**Ex 6.**	(a) 1pt (b) 2pt (c) 1pt

Total: 40 points