Deep Learning: Workbook (SoSe2018)

Dr. Josif Grabocka, Rafael Rego Drumond
HiWi: Manish K. Mihra
Information Systems and Machine Learning Lab
University of Hildesheim

Question 1: Regression and Classification - 5 points

Explain:

a)(1.Point) What is the difference between regression and classification?

b)(1.Point) What is a softmax activation function?

c)(3.Points) What would be the output of:

$$Softmax(x), \quad x = (1, 0, 2)$$

Question 2 Sthocastic gradient descent - 15 points

Consider the data-set below and the function:

Data-set

ID	**Bias**	x_1	x_2	y
#1	1	1	2	2
#2	1	2	4	4
#3	1	4	8	8

$$\hat{y}(x, \theta) = bias * \theta_0 + x_1 * \theta_1 + x_2 * \theta_2$$

With the loss function as, for each sample x:

$$\mathcal{L}(\theta, x, y) = (y - \hat{y}(x, \theta))^2$$

Perform one step of the stochatsic gradient descent with step-size $\alpha = 0.001$ and weights $\theta^{(t=0)}$ initialized as $(0.5, \quad 1.5, \quad 0.5)$. (For the sake of correction: Update in the same order as your samples). What is your new $\theta^{(t=1)}$ and how much your total loss improved? What would happen if we updated the weights again? What would happen if we used $\alpha = 0.1$ in the first update? Comment on your findings.

Question 3: Forward Propagation - 10 points

Given is a neural network with one hidden layer. The network takes as input a 2-dimensional vector $x = (x_1, x_2) \in \mathbb{R}^2$ and outputs a single value $\hat{y} \in \mathcal{Y}$. The number of neurons in the hidden layer is set to be two.

.

a) Make a scetch of the whole network architecture. Do not forget the biases in the input and the hidden layer. To make your life a little bit easier, use different variables for the network inputs and weights by using x and W for the input layer; h and v for the hidden layer and $\hat{y}$ for the final prediction.

.

b) Write down the formulas how to compute h and y^ when using ReLU as activation function for both a regression output and a binary classification output.

.

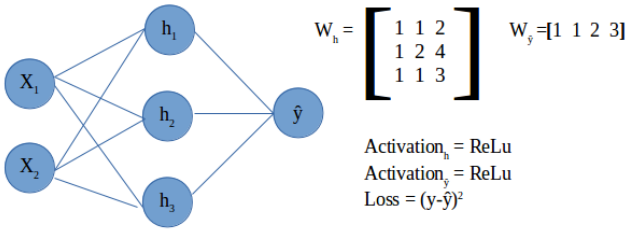
c) Predict the network output for $x = (-1 \quad 1)$ for parameters: $W = \begin{bmatrix} 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$, $v = (1 \quad -1 \quad 2)$

Question 4: Log-likelihood loss Gradient - 10 points

RESEARCH TIME: Take a look at slide 18 of the second lecture. Research and explain how to derive the gradients for the log-likelihood loss.

Question 5: Back Propagation - 20 points

a) [15 points] Given the neural network below (flowing left to right). Compute the functions for updating every weight W. Compute the forward pass for $x = (2 \quad 4)$, $y = 2.5$ and a backward pass. What are your final weights? How much it improved in terms of loss? Use gradient descent with learning-rate of 0.1.

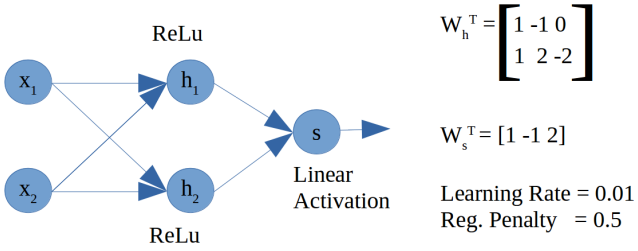


Note that activation of all nodes are ReLu.

b) [5 Points] What would happen in the network above if the loss function was defined as $L(y, \hat{y}) = \hat{y} - y$ after several iterations? And what if it was $L(y, \hat{y}) = y - \hat{y}$. (Consider that y will always be positive).

Question 6: Regularization - 15 points

Consider $x = (-1 \quad 1)$ with output $y = 2$ and the network below:



a) Perform a forward and backward pass without Regularization

b) From scratch, perform a backward pass using L1-Regularization. (Use penalty as described in the picture).

c) Comment on the results for a) and b)

Question 7: Data Augmentation- 5 points

Jane Doe has a data-set that contains speech ".wav" files. However this data-set is not so big. Jane wants to build a model that reads the ".wav" files and classify the speaker as male or female. Suggest data augmentation methods that might help Jane and explain why augmentation helps with regularization.

Question 8: Dropout-Regularization in Neural Networks (10 Points)

a) In your own words, briefly explain the dropout regularization scheme for neural networks!

b) Assume we have a neural network with ReLU activation function and want to perform a regression task, the weights (and therefore the structure) is given through:

$$W^{1T} = \begin{pmatrix} 1 & 2 \\ -1 & 1 \\ 2 & -1 \end{pmatrix} \qquad W^{2T} = \begin{pmatrix} 1 & 2 & -2 \\ 2 & -1 & 1 \end{pmatrix}$$

$$W^{3T} = \begin{pmatrix} 2 & 1 \\ -1 & -1 \end{pmatrix} \qquad v^T = (1 - 1)$$

and

$$b^1 = \begin{pmatrix} -1 \\ 1 \\ 2 \end{pmatrix} \quad b^2 = \begin{pmatrix} 2 \\ -1 \end{pmatrix} \quad b^3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad b^4 = 1$$

Predict for the instance $x^T = (1 \quad 1)$ twice, using the following dropout masks:

$$\mu^0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad \mu^1 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \qquad \mu^2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

and

$$\mu^0 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \qquad \mu^1 = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \qquad \mu^2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

c) Explain, why dropout is not used for bias nodes!

Question 9: Solving the XOR Problem (10 Points)

Implement the backpropagation learning algorithm using L2 regularization for a network with one hidden layer and two hidden neurons. The size of the input layer is also set to two. Do not forget biases. Initialize the weights by drawing from a Gaussian distribution centered around zero, and learn the weights for the XOR data:

x_1	x_2	y
1	1	1
-1	1	0
1	-1	0
-1	-1	1

Please print your source code and the convergence of your algorithm and put it to the pdf submission.

Question 10: AdaGrad and Friends! (10 Points)

a) In your own words, explain the fundamental difference between normal gradient descent, AdaGrad and Adam.

b) Assume we have a single layer neural network with a linear activation function with the following configuration and want to perform a regression task, the weights (and therefore the structure) is given through (bias included):

$$X = \begin{bmatrix} 1 & 1 & 2 \\ 1 & 1 & 1 \\ 1 & 1 & 2 \end{bmatrix} Y = \begin{bmatrix} 11 \\ 8 \\ 10 \end{bmatrix}$$

And initial weights

$$w = (1 \quad 1 \quad 1)$$

Do two epochs using stochastic gradient descent with a step size of $\mu = 0.1$ and report the errors and total loss after each epoch, with an initial $\beta = (1, 1, 1)$. Please go over the instances in order, i.e. first line, second line, third line of X.

c) Repeat the same procedure by using a stochastic gradient descent with Adagrad for an initial step size of $\mu = 0.1$. Does Adagrad help?

Question 11: Solving the XOR Problem with Adagrad (10 Points)

Implement AdaGrad, as well as the normal momentum and the Nesterov's momentum in your implementation of last week's exercise and learn the network again. Which one performs best in your experience?

x_1	x_2	y
1	1	1
-1	1	0
1	-1	0
-1	-1	1

Please print your source code and the convergence of your algorithm and put it to the pdf submission.

Question 12: Convolution Kernels (10 Points)

Given the kernels (filters) K^i and Matrices M^i perform the convolution and display the output. Remember to use padding to keep the output the same size as the input M^i.

a)

$$M^1 = \begin{bmatrix} 1 & 2 & 3 & 4 & -4 & -2 & -1 & 0 \end{bmatrix}$$

$$K^1 = \begin{bmatrix} -1 & 1 & -1 \end{bmatrix}$$

b)

$$K^2 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} M^2 = \begin{bmatrix} 1 & -10 & 10 \\ 5 & 1 & -5 \\ 1 & -10 & 0 \end{bmatrix}$$

c)

$$K^3 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} M^3 = \begin{bmatrix} -1 & 1 & -1 \\ -1 & 1 & -1 \\ -1 & 1 & -1 \end{bmatrix}$$

d)

$$K^4 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} M^4 = \begin{bmatrix} 5 & 5 & -10 \\ -5 & 1 & -1 \\ 5 & -1 & -10 \end{bmatrix}$$

e) Peform a max pooling of size $\phi = 2$ on the resulting matrices 2 to 4 (question b to d) without using padding.

Question 13: CNN Backpropagtion Part 1- FOWARD PASS (10 Points)

Given is an input image of size 3×3:

$$V^0 = \begin{pmatrix} 2 & 3 & 1 \\ 4 & 6 & 9 \\ 4 & 7 & 1 \end{pmatrix}$$

and a Convolutional neural network, where the first filter/Kernel has dimensionality $K^{(1)} \in^{1 \times 2 \times 3 \times 3}$

$$K^{(1)}_{1,1} = \begin{pmatrix} 2 & -3 \\ -3 & 2 \end{pmatrix} \qquad K^{(1)}_{1,2} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

and uses a ReLU activation function

$$f(x) = \max(x, 0)$$

The next layer uses a filter of dimensionality $K^{(2)} \in^{2 \times 1 \times 2 \times 2}$ and is given by

$$K^{(2)}_{1,1} = \begin{pmatrix} 1 & -2 \\ -4 & 3 \end{pmatrix} \qquad K^{(2)}_{2,1} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

again, using a ReLU activation function. Finally, the prediction layer employs the output of the convolutions and incorporates a bias to come up with the final prediction:

$$P(y=1) = \sigma(Z^{(2)}_{1,1} + \theta)$$

where $\theta = -2$ and σ is the sigmoid function.

Perform a complete network pass and predict the output class for V^0. Do not use padding, do not use strides, and don't be confused, there is no pooling layer involved here.

Question 14: CNN Back-propagtion Part 2- BACK-WARD PASS (10 Points)

Consider the forward pass from question 13. Using the logistic loss and label $y = 0$, backpropagate the error and compute the gradient for θ, and for the top left parameters on every K (as bold-faced below).

$$K^{(1)}_{1,1} = \begin{pmatrix} \mathbf{2} & -3 \\ -3 & 2 \end{pmatrix} \qquad K^{(1)}_{1,2} = \begin{pmatrix} \mathbf{1} & -1 \\ -1 & 1 \end{pmatrix}$$

And,

$$K^{(2)}_{1,1} = \begin{pmatrix} \mathbf{1} & -2 \\ -4 & 3 \end{pmatrix} \qquad K^{(2)}_{2,1} = \begin{pmatrix} \mathbf{2} & 1 \\ 1 & 2 \end{pmatrix}$$

$$\boldsymbol{\theta} = \mathbf{-2}$$

Do not use padding, do not use strides, and don't be confused, there is no pooling layer involved here.

Question 15: Recurrent Neural Networks - Introduction (10 Points)

Worried about their future, students from deep learning have developed a Recurrent neural network that takes into consideration the novelty of the topics from the last lectures and the difficulty of the same and output the chances of having a Backpropagation question in the next exercise. Consider the following data-set:

$Time(t)$	$Novelty(x_1)$	$Difficulty(x_2)$
1	0.1	0.1
2	0.7	0.4
3	0.5	0.2
4	0.1	0.1
5	0.8	0.8

Consider that we have a sigmoid layer at the very end and that:

$$U = \begin{pmatrix} 0.5 & 0.1 \\ 0.4 & 0.01 \end{pmatrix} \qquad W = \begin{pmatrix} 0.1 & 0.3 \\ 0.3 & 0.1 \end{pmatrix} \qquad b = (0, 0)$$

$$V^{(t)} = \begin{pmatrix} 2 & 3 \end{pmatrix} \quad c = 1 \qquad h^{(t)} = ReLU(a^{(t)}) \quad h^{(0)} = (0, 0)$$

Draw a graph of the output for each time-step and interpret the results.

Question 16: RNN recap (10 Points)

Answer the following questions with a maximum of two phrases:

a) Consider the standard RNN with some activation function $f(x)$, and matrices W and U and an input $X = (x_1, x_2, x_3, x_4, x_5)$. Write down the unrolled formula for the final output of the RNN given X.
b) Comparing LSTMs with GRUs, if we consider that LSTM and GRUs have the same learning power, what is the advantage of GRU over LSTM?
c) What is Gradient Clipping? Why is it important?
d) What is a vanishing gradient?
e) What are the main contributions of LSTMs and GRUs to RNNs.

Question 17: LSTM feed-foward (10 Points)

Consider the following sequence:

Time(t)	x
1	0.7
2	0.5
3	0.7

And the following configuration of an LSTM:

$$U^g = 0.3 \quad W^g = -0.3 \qquad b^g = 0.1$$

$$U^f = 0.25 \qquad W^f = 0.3 \qquad b^f = 0$$

$$U^q = 0.4 \qquad W^q = 0.3 \qquad b^q = 0.2$$

$$U = 0.01 \qquad W = 0.1 \qquad b = 1$$

$$h^{(0)} = 0 \qquad s^{(0)} = 0$$

where: $h^{(t)} = ReLU(s^{(t)}) * q^{(t)}$

and

$output_t = \sigma(h^{(t)} * V + c)$

$V = 5 \qquad c = -1$

Perform the foward pass on the whole sequence with an LSTM. Plot your results for each step. Is the output from $t = 1$ the same as $t = 3$? Why?

Question 18: Autoencoders Basics (10P)

Answer the following questions with a maximum of two phrases:

a) Is an Autoencoder for supervised learning or for unsupervised learning? Explain briefly.
b) What is the difference between an Undercomplete and Overcomplete Autoencoder?
c) Why do we need sparse autoencoder? Explain briefly.
d) What is the objective of Denoising Autoencoders and Contractive Autoencoders?
e) What is the similarity between autoencoder and PCA method? How are the two different? What constrain would you apply to an autoencoder to make it similar to a PCA method?RNNs.

Question 19: Sparse Autoencoders (10 Points)

For some $x \in \mathbb{R}^3$, we want to learn an autoencoder that encodes x as a two-dimensional vector $h \in \mathbb{R}^2$ using some parameters $W \in \mathbb{R}^{2\times 3}$ and nonlinearity g

$$h = g(Wx)$$

and a decoder that from h is supposed to predict the original x using parameters $V \in \mathbb{R}^{3\times 2}$:

$$\tilde{x} = Vh$$

Additionally, we want to regularize the hidden encoding h, so we overall want to minimize the loss:

$$\mathcal{L}(x, W, V) = \sum_{i=1}^{3} (\tilde{x_i} - x_i)^2 + \lambda \sum_{i=1}^{2} |h_i| \qquad \lambda > 0$$

Compute the gradients of the loss function for all model parameters!

Question 20: GAN Basics (5 Points)

Answer the following questions with a maximum of two bullet points each (third bullet points and follow ups will be disconsidered):

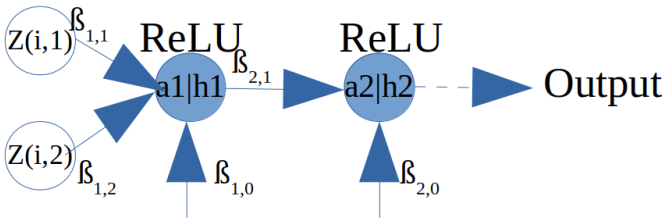
a) Give two examples of a Minimax Scenario outside Deep Learning and Computer Games. (1 point)
b) Why the discrimantor is trained with gradient ascent and not descent by default? (1 point)
c) Why, in practice, it is not a good idea to learn parameters from the generator and discriminator at the same time? (3 Points)

Question 21: GAN Back-Propagation (15 Points)

Given the two networks below, write down the udpate equations (given some learning rate λ) for every weighta β of the Generator and every weight θ of the discriminator. Consider Z as input noise for the generator. And X as a sample to be discriminated. Explain how those update equations would be used while training this network.

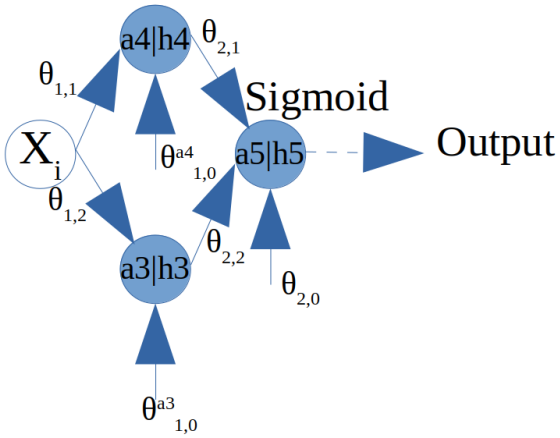
.
.
.

Generator G



Discriminator D

Relu



.

Warning!

If we detect **Plagiarism** on your solution, you will receive no points for it. If a second plagiarism attempt is detected, you might fail the class or be expelled from your program.

You are allowed to discuss solutions, but if you work on a group, you must indicate on your sheet with whom are you working with.

Group submissions earn 0 points, but counts as participation.

Bonus Points!

During the tutorials, you will have the chance of earning up to 10% of extra points to your final exam. Submission grades represent 75% of this bonus. The rest (25%) is earned by attending to at least 70% of the tutorial sessions OR submitting at least 70% of the sheets.