Introduction to Deep Learning
Exam SS18, 16.07.2018

This collection of questions is purely based on memories of students attending the exam and not on any written notes in any form whatsoever.

General Notes

- ➢ It was 17 pages of mostly short questions (90 points in total, mostly 2 per question, sometimes 4, sometimes 1)
- ➢ time was tight, but not undoable
- ➢ The exam consisted of 8 different question blocks ranging from 7 to 18 points. The following collection might have missed some blocks, not know the given points or have assigned questions to the wrong block. Yet, you might get an idea of how the questions look and what to expect in which form.
- ➢ Multiple-/Single-Choice questions. It was given that some exercises are meant to have several solutions or only one single solutions. This was illustrated graphically.
  - ❑ Means: several answers could be correct, tick all correct ones
  - ● Means: only one answer is correct
- ➢ Recommendation: Tick the box that you allow the chair publishing your grade to get your grade earlier in an not-official published list of matricle numbers with corresponding exam points, grades and grades with bonuses.

Grade Key	Exam Points
1.0	85 - 90.0
1.3	80 - 84.5
1.7	75 - 79.5
2.0	70 - 74.5
2.3	65 - 69.5
2.7	60 - 64.5
3.0	55 - 59.5
3.3	50 - 54.5
3.7	45 - 49.5
4.0	40 - 44.5
5.0	00 - 39.5

Part I: Multiple Choice (18 points)

1.) Which of the following introduce nonlinearities?

- ❏ Convolutions
- ❏ Pooling
- ❏ Batch Normalization
- ❏ ReLUs

2.) Which of the following contain learnable parameters

- ❏ Dropout
- ❏ Batch Normalization
- ❏ Max-Pooling
- ❏ ...

3.) And Regression

Given is the following and-operation between inputs x_1 and x_2.

X_1	x_2	and(x_1, x_2)
1	1	1
1	0	0
0	1	0
0	0	0

Which of the following weights and biases represents a valid linear regression for that task. Given that non-linearity is applied afterwards: f(x) = 1 for x > 0, 1 else

- W = (1,1), b = 2
- W = (1,1), b = -0.5
- W = (1, 1), b = -1.5
- None of the above

4.) Regularization: Which is true for the established L1 and L2 Norm Regularization?

- ❏ add additional resdiuals to cost functions
- ❏ Typically not applied to biases
- ❏ Makes training harder
- ❏ ...

5.) Why do we use multiples of 2 for the mini batch size?

- ❏ GPUs are optimized for that
- ❏ Makes computation faster
- ❏ Converges faster
- ❏ …

6.) You train a network and the loss diverges. Which of the following is reasonable?

- ❏ Reduce the learning rate
- ❏ Change optimizer
- ❏ Add dropout
- ❏ Add more parameters

7.) Logistic Regression

- ❏ Can not be used for binary classification
- ❏ Uses hinge loss
- ❏ Outputs values in the range [-1, 1]
- ❏ Is optimized through a cross-entropy loss

8.) ….

9.) ...

Part II: Short Questions

1.) What kind of architecture would you choose for a network to translate english to german? (RNN as it considers sequential order of sentence structure, maybe also mention LSTM as well-performing architecture)

2.) Draw a ResNet block. What is the main advantage of using such a block?

- Skip connection + sum
- Advantage: Fast gradient flow counters vanishing gradient. Easier training of very deep networks.

3.) Given a net with tanh activations, 50 x 50 input image size and 1024 hidden layer size. What is the coefficient for the variance for Xavier initialization? How does it change for ReLU activations?

- 1/n, where n=50x50
- ReLU: 2/n

4.) You have an input of size 50x50x200 and a convolutional layer with kernel 10x10, your output should be size 10x10x10, how many multiplications does one forwards pass involve? What would you do, to reduce this number?

- Kernel size * input channels * output feature map size * output channels
- 1x1 conv (no pooling because the output dim should remain the same)

5.) You have a CNN with 4 layers with convolution masks of 3x3. What is the receptive field of a neuron in the last layer. 9x9

6.) Why is minimizing the cross entropy between desired output distribution q and and actual estimation distribution p (H(q,p) the same as minimizing their KL divergence KL(q,p) = H(q,p) - H(p)?

7.) You decrease your mini batch size. Do you increase or decrease your learning rate? Explain why.

Part III: tanh-network

Given is a fully-connected network with 15 layers using tanh as activation functions.

1.) Explain and describe the behaviour of such a network with respect to large outputs
2.) Which problems might occur during training on large inputs?
3.) Now the network is initialized to small random numbers. Which problems do foresee?
4.) Name a method that makes the network robust against initialization. (1 point) Batch norm
5.) If one now uses a ReLU as activation-function, which problems are solved? Which problems do still exist?
6.) How to initialize weights better than using small random numbers?

Part IV: Calculations

Batch Normalization
Given is the Batch-Normalization-operation
$y = \frac{(x-\mu)}{\sigma}\xi + \phi$

1. Compute the gradients of the loss function with respect to the parameters ξ, ϕ given the gradient $\partial L / \partial y_{ij}$.
2. How can you accumulate the backward gradients? ?? Very unsure about that one

Convolution
Given is a simple convolutional operation. You have an input of dimension 2x2x2 and apply a convolutional kernel with filter size 1 and one output channel. This convolutional layer is followed by a max-pooling layer of size 2x2 with stride 2 and a padding of 1.

Given are the following values:
$x = ([1\ -1.5;\ 1\ -2], [-2\ \ 1;\ -1.5\ \ 1]),\ ytarget = [\,0\ \ 1;\ 1\ \ 0]$

1. Compute the output.
2. Compute the binary cross-entropy loss.
3. Compute the weight update given a gradient with respect to the outputs consisting of ones and a learning rate of 1.

Part V: Training

You have two Models, *Model 1* and *Model 2* and train them the same way. You find out, that *Model 1* has too few parameters, to perform well.

1.) What is this called. Draw test and validation loss. → underfitting
2.) You now severely increase the model complexity of *Model 1*, but *Model 2* still performs better. What happened now? Draw again training and validation loss.
3.) You use Model2 now - there is a Diagramm given with 3 learning rates - order the the learning rates based on their size. (highest learning rate increased fastest but diverges very fast. Small learning rate smallest at beginning then also diverges)

Part VI: Optimization details

1. Explain difference between vanilla SGD and RMSProp.
2. Write down the formula of momentum update step.
3. Explain difference between Newton and quasi-Newton-methods, like BFGS.
4. Explain difference between Newton methods and the Gauss-Newton method.
5. Explain beta_1 and beta_2, the parameters of the Adam optimizer.
6. Name two techniques for learning rate scheduling.

Part VII: Other questions (not assigned to blocks)

1. Name two techniques used in Deep learning where you use a dataset separate from the training dataset