Chair of Chair of Visual Computing & Artificial Intelligence
Department of Informatics
Technical University of Munich

Esolution
Place student sticker here

Note:
- During the attendance check a sticker containing a unique code will be put on this exam.
- This code contains a unique number that associates this exam with your registration number.
- This number is printed both next to the code and to the signature field in the attendance check list.

Introduction to Deep Learning

Exam:	IN2346 / Endterm	**Date:**	Wednesday 19th February, 2020
Examiner:	Prof. Dr. Matthias Nießner	**Time:**	13:30 – 15:00

	P 1	P 2	P 3	P 4	P 5	P 6	P 7
I							

Working instructions

- This exam consists of **20 pages** with a total of **7 problems**.
  Please make sure now that you received a complete copy of the exam.
- The total amount of achievable credits in this exam is 89 credits.
- Detaching pages from the exam is prohibited.
- Allowed resources: **none**
- Do not write with red or green colors nor use pencils.
- Physically turn off all electronic devices, put them into your bag and close the bag.
- If you need additional space for a question, use the additional pages in the back and properly note that you are using additional space in the question's solution box.

Left room from ________ to ________ / Early submission at ________

Problem 1 Multiple Choice (18 credits)

Multiple Choice Questions:

- For all multiple choice questions any number of answers, i.e. either zero (!) or one or multiple answers can be correct.
- For each question, you'll receive 2 points if all boxes are answered correctly (i.e. correct answers are checked, wrong answers are not checked) and 0 otherwise.

How to Check a Box:

- Please **cross** the respective box: ☒ ☒ (interpreted as **checked**)
- If you change your mind, please **fill** the box: ■ ■ (interpreted as **not checked**)
- If you change your mind again, please put an additional **cross** besides the box.

a) You train a neural network and the loss diverges. What are reasonable things to do?

- ☐ Try a different optimizer.
- ☐ Add dropout.
- ☐ Decrease the learning rate.
- ☐ Increase the number of parameters.

b) Use a neuron to regress the **AND function**. The activation function of the neuron is $f(x) = 0$ if $x < 0$ else 1. What is the weight and bias of the neuron?

x_1	x_2	x_1 AND x_2
1.0	1.0	1.0
1.0	0.0	0.0
0.0	0.0	0.0
0.0	1.0	0.0

- ☐ Bias: 1.5, w_1: 2.0, w_2: 2.0
- ☐ Bias: -1.0, w_1: 1.5, w_2: 1.5
- ☐ Bias: -1.5, w_1: 1.0, w_2: 1.0
- ☐ Bias: -2.0, w_1: 1.4, w_2: 1.2

c) You want to train an autoencoder to overfit a single image with a fixed learning rate. Setting numerical precision aside, which loss function **is able to** reach **zero** loss after training with gradient descent.

- ☐ L2
- ☐ L1

d) Regularization:

- ☐ Is a technique that aims to reduce the generalization gap.
- ☐ Dropout, the use of ReLU activation functions, and early stopping can all be considered regularization techniques.
- ☐ Weight decay (L^2) is commonly applied in neural networks to spread the decision power among as many neurons as possible.
- ☐ Is a technique that aims to reduce your validation error and increases your training accuracy.

e) Which statements are correct for a Rectified Linear Unit (ReLU) applied in a CNN?

- [ ] Despite a small learning rate, without saturation the gradients are very likely to explode.
- [ ] A large negative bias in the previous layer can cause the ReLU to always output zero.
- [ ] Large and consistent gradients allow for a fast network convergence.
- [ ] Max pooling must always be applied after the ReLU.

f) You want to use a convolutional layer to decrease your RGB image size from 2545x254 to 127x127. What parameter triplets achieve this?

- [ ] Kernel size 3, stride 2, padding 1
- [ ] Kernel size 6, stride 2, padding 2
- [ ] Kernel size 2, stride 2, padding 0
- [ ] Kernel size 5, stride 2, padding 2

g) Your train and val loss converge to about the same value. What would you do to increase the performance of your model?

- [ ] Add more input features.
- [ ] Increase the capacity of your model.
- [ ] Add more training data.
- [ ] Add regularization.

h) An autoencoder

- [ ] has no loss function.
- [ ] has a bottle neck layer.
- [ ] is often used for fine-tuning pre-trained models.
- [ ] requires class labels for training.

i) What is the correct order of operations for an optimization with gradient descent?

(a) Update the network weights to minimize the loss.

(b) Calculate the difference between the predicted and target value.

(c) Iteratively repeat the procedure until convergence.

(d) Compute a forward pass.

(e) Initialize the neural network weights.

- [ ] bcdea
- [ ] ebadc
- [ ] eadbc
- [ ] edbac

Problem 2 Short Questions (22 credits)

0 1 2

a) Kaiming initialization corresponds to Xavier initialization with the variance multiplied by two. In which case (1p) and why (1p) would you chose this initialization?

0 1 2

b) You are given a convolutional layer with kernel size 3, number of filters 3, stride 1 and padding 1. Compute the shape of the weights (0.5p) and write them down explicitly such that this convolutional layer represents the identity for an RGB image input (1.5p).

0 1 2

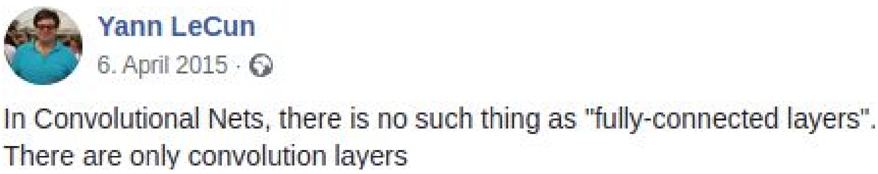
c) Name two reasons to use an inception layer in favor of a standard convolutional layer.

0 1 2

d) What are the definitions of *bias* and *variance* in the context of machine learning?

e) In Generative Adversarial Networks the generator and discriminator play a two player min-max game. Why is this hard to optimize and what heuristical method is used instead of the default min-max formulation.

f) Consider the quote below. Demonstrate how a fully connected layer with an input size of 512 neurons and an output of 10 neurons can be modeled as a convolutional layer.

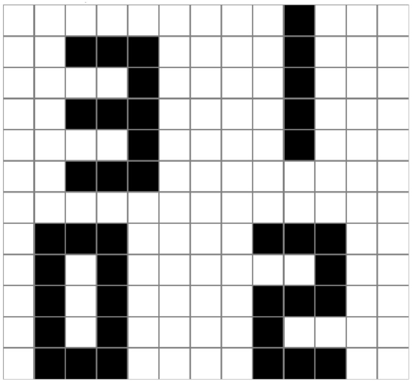

Yann LeCun
6. April 2015 ·

In Convolutional Nets, there is no such thing as "fully-connected layers".
There are only convolution layers

g) Explain the Markov Assumption in reinforcement learning.

h) Where do we use neural networks in Q-learning (1p) and why are they needed (1p)?

i) The following image shows a rectangular image of size 16 $\times$ 16. Design a 5 $\times$ 5 convolutional filter that is maximally activated when sliding over the '3' in the image below (Black pixels are 1s and white are -1. For simplicity use only -1s and 1s in your designed filter).



j) Name one advantage and one disadvantage of Recurrent Neural Networks in general.

k) Long-Short Term Memory Units suffer less from the vanishing gradient problem than vanilla RNNs. What two design changes make this possible?

Problem 3 House Prices and Backpropagation (11 credits)

You are tasked to predict house prices for your first job, i.e. for a given input vector of numbers you have to predict a single floating point number that indicates the house price (e.g., between 0 and 1m euros).

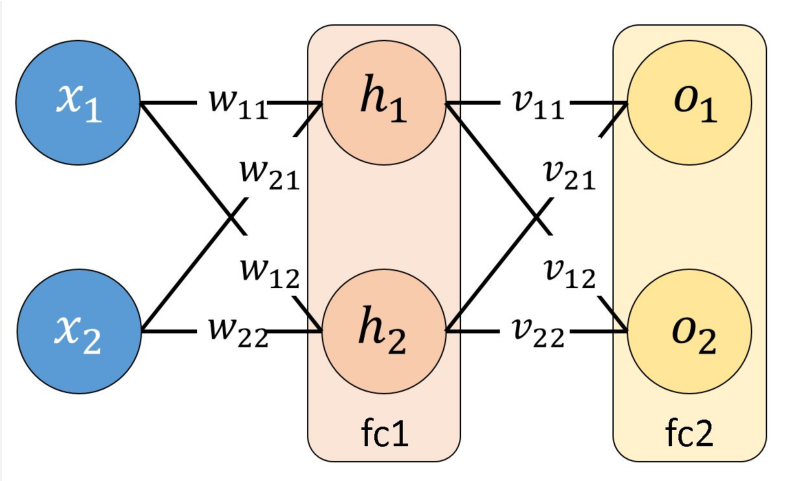
a) What network loss function would you suggest for this problem (0.5p) and why (0.5p)?

0
1

b) How would you approximate the task as a classification problem (0.5) and which loss function would you propose in this situation (0.5)?

0
1

After collecting some data you start off with a simple architecture



Your architecture is composed of two fully-connected layers (fc1 and fc2) which both contain

- a linear layer with weights and biases as outlined on the next page,
- followed by a Dropout with a probability parameter of 0.5,
- as well as a Leaky ReLu with a parameter of 0.5 as your non-linearity of choice at the end.

c) How many trainable parameters does your network have?

0
1

Weights and biases of the linear layers:

Layer	fc1						fc2					
Nodes	h_1			h_2			o_1			o_2		
Variable	w_{11}	w_{21}	b_{h_1}	w_{12}	w_{22}	b_{h_2}	v_{11}	v_{21}	b_{o_1}	v_{12}	v_{22}	b_{o_2}
Value	1.0	-1.0	1.0	-1.0	1.0	-1.0	0.5	-1.0	1.0	1.0	1.0	0.0

0 1 2 3 4

d) You are experiencing difficulties during training and thus decide to check the network in test mode manually. In your test case the input x values are

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

What is the output of your network for that input? Write down all computations for each step.

0 1

e) As you were unsure of your loss function, you phrase the task as a binary classification problem for each of your two outputs independently. Calculate the binary cross-entropy with respect to the natural logarithm for the labels

$$y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

You may assume $0 \cdot \ln(0) = 0$. (Write down the equations and keep the simplified logarithm.)

f) Please update the weight v_{21} using gradient descent with a learning rate of 1.0 with respect to the binary cross-entropy loss as well as labels from e) and the forward computation from d). (Please write down all your computations. Writing down formulas and values in general form is encouraged)

0
1
2
3

Problem 4 Optimization (11 credits)

0
1

a) Why can't we expect to find a global minima while training neural networks?

0
1

b) Why is finding a local minimum often enough?

0
1
2

c) What is a saddle point (1p)? What is the advantage/disadvantage of Stochastic Gradient Descent (SGD) in dealing with saddle points (1p)?

0
1

d) Explain the concept behind momentum in SGD.

0
1

e) Which optimizer introduced in the lecture uses second but not first order momentum?

0
1
2

f) Explain the beta (β_1 and β_2) hyperparameters of Adam with respect to the gradients.

g) What would be an advantage of a second order optimization method such as the Newton method (1p) ? Why is it not commonly used in the context of neural networks?

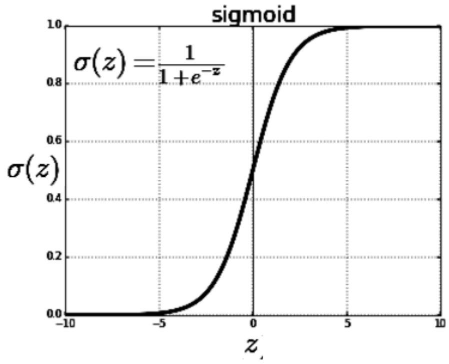
0 1

h) Name the key idea of Newton's method (1p) and write down the update step formula (1p).

0 1 2

Problem 5 **Network Architectures and Training (10 credits)**

You are training a neural network with 10 convolutional layers and the activation function shown below:



0 1

a) What is the purpose of activation functions in neural networks?

0 1

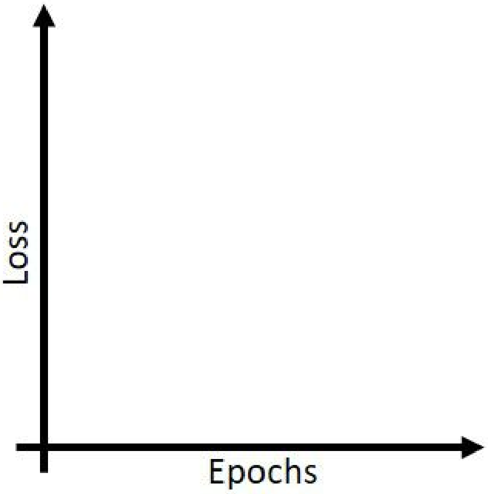
b) You find that your network still is not training properly and discover that your network weights have all been default initialized to 1. Why might this cause issues for training?

0 1 2

c) How can you resolve the problems with weight initialization and provide stable training in most scenarios for the proposed network? Explain your solution.

0 1 2

d) After employing your solutions, you are ready to train your network for image segmentation. After 50 epochs, you come to the conclusion that the network is too large for such a task. What is this effect called? How do you make this observation? Make a plot of the corresponding training (regular line) and validation losses (dashed line) and name them appropriately.

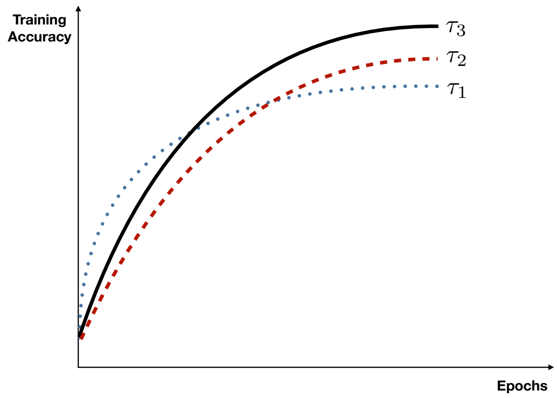


e) Without changing the convolutional layers of your network, name two approaches to counteract the problems encountered in (f).

0
1
2

0
1
2

f) You adapt your network training accordingly, and now are performing a grid search to find the optimal hyperparameters for vanilla stochastic gradient descent (SGD). You try three learning rates τ_i with $i \in \{1, 2, 3\}$, and obtain the following three curves for the training accuracy. Order the learning rates from larger to smaller.


Training
Accuracy
τ_3
τ_2
τ_1
Epochs

Problem 6 **Batchnormalization (5 credits)**

A friend suggested you to use Batchnormalization in your network. Recall the batch normalization layer takes values $x = (x(1), ..., x(m))$ as input and computes $x = (x^{(1)}_{norm}, ..., x^{(m)}_{norm})$ according to:

$$x^{(k)}_{norm} = \frac{x^{(k)} - \mu}{\sqrt{\sigma^2}} \qquad \text{where } \mu = \frac{1}{m}\sum_{k=1}^{m} x^{(i)}, \quad \sigma^2 = \frac{1}{m}(x^{(k)} - \mu)^2.$$

It then applies a second transformation to get $y = (y^{(1)}, ..., y^{(m)})$ using learned parameters $\gamma^{(k)}$ and $\beta^{(k)}$:

$$y^k = \gamma^{(k)}(x)^k_{norm} + \beta^{(k)}.$$

a) How would you make the formulation above numericly stable?

0 1

b) How can the network undo the normalization operation of Batchnorm? Write down the exact parameters.

0 1 2

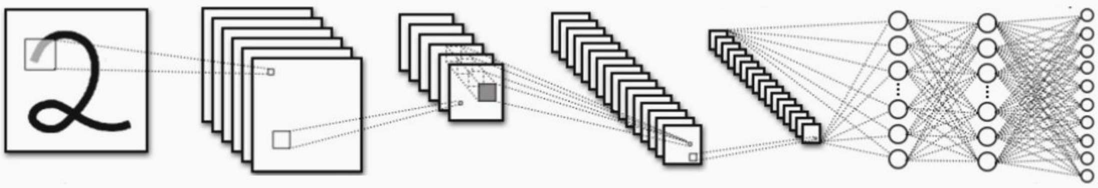
c) Name the main difference of Batchnormalization during training or testing and note down eventual parameters that need to be stored.

0 1 2

Problem 7 Convolutional Neural Networks (12 credits)

You are contemplating design choices for a convolutional neural network for the classification of digits. LeCun et. al suggest the following network architecture:

Layer:	input	conv1	maxpool1	conv2	maxpool2	f1	f2	out
Shape:	1x32x32	6x28x28	6x14x14	16x10x10	16x5x5	120	84	10



For clarification: the shape **after** having applied the operation ‘conv1’ (the first convolutional layer in the network) is 6x28x28.
All operations are done with stride 1 and no padding. For the convolution layers, assume a kernel size of 5x5.

a) Explain the term ‘receptive field’ (1p). What is the receptive field of one pixel after the operation ‘maxpool1’(1p)? What is the receptive field of a neuron in layer ‘f1’ (1p)?

b) Instead of digits, you now want to be able to classify handwritten alphabetic characters (26 characters). What is the **minimal** change in network architecture needed in order to support this?

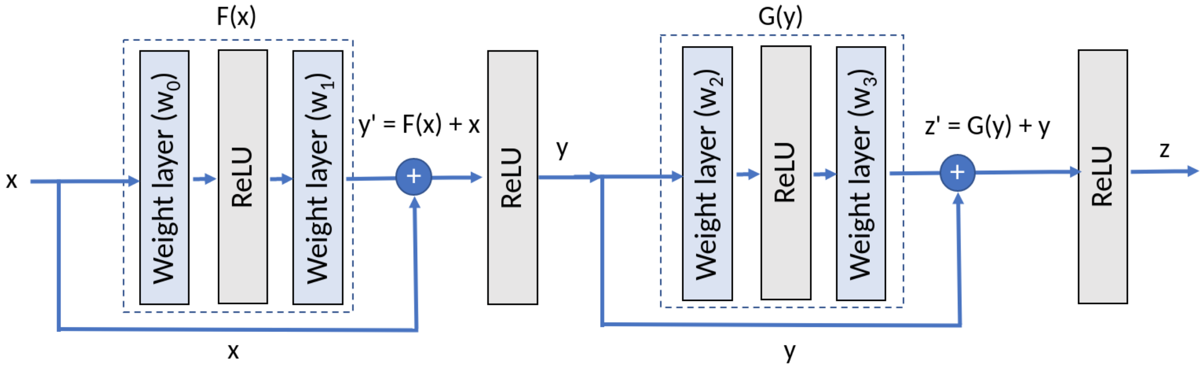
c) Instead of taking 32×32 images, you now want to train the network to classify images of size 68×68. List two possible architecture changes to support this?

d) Your architecture works and you manage to classify characters fairly well. After reading many online blogs, you decide to try out a much deeper network to boost the network's capacity. Name 2 problems that you might encounter when training very deep networks.

0 1 2

e) You read that skip connections are beneficial for training deep networks. In the following image you can see a segment of a very deep architecture that uses skip connections. How are skip connections helpful? (1p). Demonstrate this mathematically by computing gradient of output z with respect to 'w_0' for the network below in comparison to the case without a skip connection (3p). For simplicity, you can assume that gradient of ReLU, $\frac{d(ReLU(p))}{dp} = 1$.

0 1 2 3 4



Additional space for solutions–clearly mark the (sub)problem your answers are related to and strike out invalid solutions.