Chair of Machine Learning of 3D Scene Geometry
School of CIT
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:** Friday 10th February, 2023
Examiner: Prof. Dr. Angela Dai **Time:** 18:30 – 20:00

	P 1	P 2	P 3	P 4	P 5	P 6	P 7	P 8
I								

Working instructions

- This exam consists of **20 pages** with a total of **8 problems**.
  Please make sure now that you received a complete copy of the exam.
- The total amount of achievable credits in this exam is 90 credits.
- Detaching pages from the exam is prohibited.
- **Answers are only accepted if the solution approach is documented.** Give a reason for each answer unless explicitly stated otherwise in the respective subproblem.
- 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.

Left room from ________ to ________ / Early submission at ________

Problem 1 Multiple Choice (18 credits)

Mark correct answers with a cross ☒
To undo a cross, completely fill out the answer option ■
To re-mark an option, use a human-readable marking × ■

Please note:

- For all multiple choice questions any number of answers, i.e. either zero (!), 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.**

1.1 Your model for classifying different cat species is getting a low training set error with a high testing set error. Which of the following are promising things to try to improve your classifier?

- ☐ Use a bigger neural network
- ☐ Get more training data Not a valid of feasable solution
- ☐ Try a different initialization during training
- ☒ Add weight regularization

1.2 Which of the following statements on activation functions are true?

- ☐ The output values should be in the range of 0 to 1
- ☒ Tanh can lead to vanishing gradients
- ☐ Sigmoid outputs are zero-centered
- ☒ Parametric ReLU can handle negative input values

1.3 Which of the following propositions are true about a Conv layer?

- ☐ The total number of parameters depends on padding
- ☐ The total number of parameters depends on the width and height of the input
- ☒ The output depth is the same as the number of filters
- ☐ The number of input channels and the number of filters' channels can differ

1.4 Logistic regression:

- ☒ Allows performing binary classification.
- ☒ Uses a variant of the cross entropy loss.
- ☒ Can be seen as a 1-layer neural network.
- ☐ The output space is between -1 and 1.

1.5 Regularization:

- ☐ Is any technique that aims to reduce your validation error and increase your training accuracy.
- ☒ Is any 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.

1.6 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.

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

1.7 So far we've learned Fully Connected Neural Network (FC), Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). In which architecture does weight sharing occur across an input?

- [ ] FC
- [x] CNN
- [x] RNN
- [ ] None

1.8 Dropout...

- [ ] ... makes your network train faster.
- [x] ... can be seen as an ensemble of networks.
- [x] ... is an efficient way for regularization.
- [ ] ... has trouble with tanh activations.

1.9 Which of the following methods can be used in unsupervised learning?

- [x] Autoencoder.
- [x] PCA.
- [x] K-means.
- [ ] Linear Regression.

Problem 2 Short Questions (19 credits)

0 1 2

2.1 Give one application scenario to use 1x1 convolution.

(+ 2p) Keep the input dimensions and scales / Shrink the number of channels / Change no. of channels / Further adds a non-linearity / Stack more layers (compared to 3x3, 5x5) / Reduce computational cost / Replace fully connected layers Specific application: inception layers / GoogleNet
(0.5p): if only reduce dimensionality / keep the output size but without specific depth or spatial size.
(0p): Change output classes / u-net / segmentation / autoencoder without mentioning reasons

0 1 2

2.2 Explain the differences between binary classification and multiclass classification in terms of the output layer and loss function.

Binary Class.: Activation: Sigmoid **(+ 0.5p)**. Loss: BCE **(+ 0.5p)**. (If stated that the BCE is a special case of CE, so stated Softmax+CE for this, got the point also.)
Multi. Class.: Activation: Softmax **(+ 0.5p)**. Loss: CE **(+ 0.5p)**.
(0p): "output layer" when not explicitly stating sigmoid/softmax **or** for sigmoid/softmax as loss function.

0 1 2

2.3 A convolutional neural network has 3 consecutive layers as follows:
5x5 Conv (stride 2) - 3x3 Conv (stride 2) - 3x3 Conv (stride 2).
How large is the receptive field of a pixel on the output? Note: Give it by MxM.

(+ 2p): 17x17
Method 1: 3*2+3-2=7; 7*2+5-2=17
Method 2: $r_0 = 1 \to r_1 = 5 \to r_2 = 5 + (2)(3-1) = 9 \to r_3 = 9 + (2 \cdot 2)(3-1) = 17$
(- 0.5p): For 17, but not 17×17 / a calculation mistake.
(- 1p): Mistake in formula.

0 1 2

2.4 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. Let's use the order of (Kernels, Channels, H, W) for the shape (0.5p). Write them down explicitly such that this convolutional layer represents the identity for an RGB image input. (1.5p).

(+ 0.5p): Shape of W=(3,3,3,3)
W[0,0,1,1] = W[1,1,1,1] = W[2,2,1,1] = 1, else 0 (Each **(+ 0.5p)**)

0 1 2

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

Advantages: **(1p)**: model any output / Sequential data / input of arbitrary length / Lightweight. **(0.5p)**: process language, video, sound, or other applications / Can make use of past output information / More flexible. **(0p)**: Highway for gradient.
Disadvantages: **(1p)**: Vanishing or exploding gradient / No long-term memory / Complicated architecture / Difficult parallelization. **(0.5p)**: Only close inputs can interact / Long training time. **(0p)**: Difficult to compute.

2.6 Briefly explain the concept of weight initialization of a neural network (1p). Name one bad method of initialization and explain why it is bad (1p). Additionally, name two common initialization strategies (0.5p each).

0 1 2 3

(+ 1p): Initialize weight matrices with random non-zero values.
(1p): Zero-initialization **(+ 0.5p)**. Does not break symmetry / Gradients will be the same **(+ 0.5p)**.
(1p): He Kaiming Initialization **(+ 0.5p)**. Xavier Initialization **(+ 0.5p)**.

Common Mistakes:
(0.5p): Gauss / random initialization are not strategies.

2.7 What is “early stopping”?

0 1 2

(+ 1p): Stop training, if the validation loss (based on some metric/loss) increases or stops decreasing.
(+ 1p): Prevent overfitting to train data.
Common Mistakes: Mentioning only "loss".

2.8 Define “data augmentation" (0.5p), name two common data augmentation techniques used in image classification (0.5p each), and how could data augmentation be problematic in a supervised training scenario (1p)?

0 1 2

(+ 0.5p): Artificially increase the amount of training data by adding transformations.
(+ 0.5p) each (req. 2): Crop / Flip / Any method mentioned on the PyTorch website.
(+ 1p): Can lead to incorrect label, e.g. after cropping, GT label does not match anymore.
Common Mistakes: Just stating "better generalization" / Training time / Only examples / Not referring to wrong labels, or distribution mistmatch / explanation refers to a single example, but not to the general problem.

2.9 Consider two different models for image classification of the MNIST data set.
The models are: (i) a 3 layer perceptron, (ii) LeNet.
Which of the two models is more robust to translation of the digits in the images? Give a short explanation why.

0 1

(+ 0.5p): LeNet.
(+ 1p): LeNet has convolutional layers, which are more robust to translation than fully-connected layers.
(- 0.5p) Students wrote too much text which included wrong statements.
Notes: Conv. translation equivariant accepted as an explanation.

Problem 3 Backpropagation (8.5 credits)

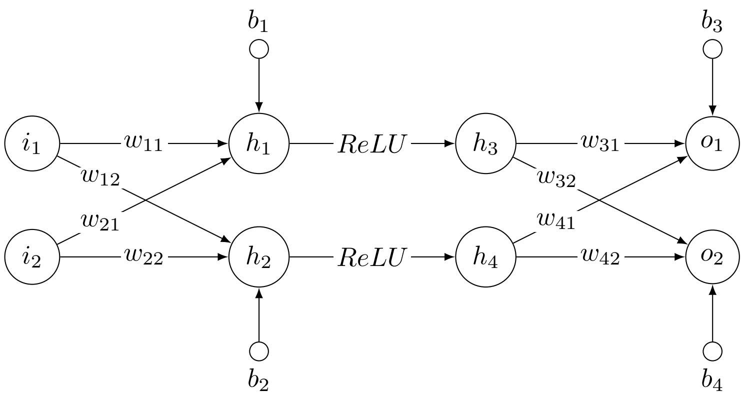


Figure 3.1: Simple network.

The values of variables are given in the following table:

Variable	i_1	i_2	w_{11}	w_{12}	w_{21}	w_{22}	w_{31}	w_{32}	w_{41}	w_{42}	b_1	b_2	b_3	b_4	t_1	t_2
Value	2.0	-1.0	1.0	-0.5	0.5	-1.0	0.5	-1.0	-0.5	1.0	0.5	-0.5	-1.0	0.5	1.0	0.5

3.1 Compute the outputs (o_1 and o_2) of this network. Therefore, you will need to calculate the following variables: $h_1, h_2, h_3, h_4, o_1, o_2$.

$$h_1 = i_1 \times w_{11} + i_2 \times w_{21} + b_1 = 2.0 \times 1.0 - 1.0 \times 0.5 + 0.5 = 2.0$$

$$h_2 = i_1 \times w_{12} + i_2 \times w_{22} + b_2 = 2.0 \times -0.5 + -1.0 \times -1.0 - 0.5 = -0.5$$

$$h_3 = max(0, h_1) = h_1 = 2$$

$$h_4 = max(0, h_2) = 0$$

$$o_1 = h_3 \times w_{31} + h_4 \times w_{41} + b_3 = 2 \times 0.5 + 0 \times -0.5 - 1.0 = 0$$

$$o_2 = h_3 \times w_{32} + h_4 \times w_{42} + b_4 = 2 \times -1.0 + 0 \times 1.0 + 0.5 = -1.5$$

(+ 0.5p): For each correcnt result.
Notes: Follow-up errors are accepted.

3.2 Write down the formula of the Mean Squared Error, and calculate the loss using your results in the previous question and the target values (t_1 and t_2). In case you have not solved the previous question, use the following values: $o_1 = 2$ and $o_2 = 0.5$.

$$MSE = \frac{1}{2} \times (t_1 - o_1)^2 + \frac{1}{2} \times (t_2 - o_2)^2 = 0.5 \times 1.0 + 0.5 \times 4.0 = 2.5$$

(+ 1p): Correct formula.
(+ 0.5p): Correct result.

3.3 Please update the weight w_{21} using gradient descent with learning rate $\alpha = 0.1$ as well as the loss computed previously. (Please write down all your computations.)

0 1 2 3 4

Backward pass (Applying chain rule):

$$\frac{\partial MSE}{\partial w_{21}} = \frac{\partial \frac{1}{2}(t_1 - o_1)^2}{\partial o_1} \times \frac{\partial o_1}{\partial h_3} \times \frac{\partial h_3}{\partial h_1} \times \frac{\partial h_1}{\partial w_{21}} + \frac{\partial \frac{1}{2}(t_2 - o_2)^2}{\partial o_2} \times \frac{\partial o_2}{\partial h_3} \times \frac{\partial h_3}{\partial h_1} \times \frac{\partial h_1}{\partial w_{21}}$$

$$= (o_1 - t_1) \times w_{31} \times 1.0 \times i_2 + (o_2 - t_2) \times w_{32} \times 1.0 \times i_2$$

$$= (0 - 1.0) \times 0.5 \times -1.0 + (-1.5 - 0.5) \times -1.0 \times -1.0$$

$$= 0.5 + -2.0 = -1.5$$

Update using gradient descent:

$$w_{21}^{+} = w_{21} - lr * \frac{\partial MSE}{\partial w_{21}} = 0.5 - 0.1 * -1.5 = 0.65$$

(+ 1p): Correct formula.
(+ 1p): Correct derivation.
(+ 1p): Correct result.
(+ 1p): The gradient descent update.

Problem 4 Optimization (6 credits)

0 1

4.1 Explain the concept behind momentum in SGD.

(+ 1p): Full answer: accumulating previous gradients / use weighted average, **and** goal: Avoid getting stuck in saddle points or accelerate optimization.
Only **(0.5p)**: Only "bigger steps" w/o an outcome / stating that SGD+Momentum was not mentiond / giving only half of the solution. **Accepted**: Previous gradients w/o weighted average / faster **training** / accumulated velocity.
Common Mistakes: GD vs SGD / RMSProp instead of Momentum /
/ Damping oscillations / global minima.
Notes: Wrong statements remove **0.5p** if the full answer is given.

0 1

4.2 Which optimizer introduced in the lecture uses second but not first order moment?

(+ 1p): RMSProp, or with a small typo.
(0.5p): It was clear that the student knew that it is RMSProp, but invented a whole new name for it, as long as it doesn't contain any of the other known optimizers.
Common Mistakes: Adam / Adagrad / Nestrov / Gauss-Newton

0 1 2

4.3 Name a disadvantage of a small minibatch/batch size and a disadvantage of a large minibatch/batch size.

Dis. small batch **(+ 1p)**:
- Noisy updates / high variance / stochastic / Slower training / Many more training steps / Disable the usage of the Gaus-newton optimizer.
- **(-0.5p)**: Overfitting / diverge / make result worse

Dis. Big batch **(+ 1p)**:
- Requires more memory / Heavy usage of GPU / Might be stuck at a saddle point / longer iterations (update steps).
- **(-0.5p)**: More computation time, with no reference to which time variable.

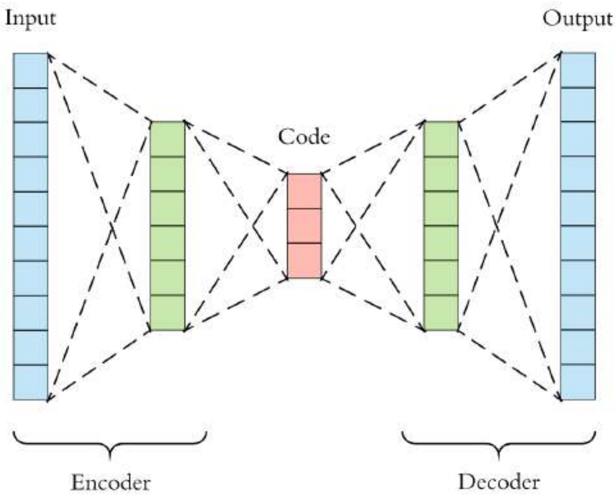
Notes: Points were reduced only for the relevant sub problem and only if a correct answer was given additionally.

0 1 2

4.4 Why is Newton's method not commonly used in training a deep model (1p)? What would be an advantage of using it (1p)?

(+ 1p) Not commonly used:
Option 1: Basis **(0.5p)** - Comutatinally complex / expansive / any variant of that. **(+ 0.5p)** if stated the hessian matrix / second derivative. If stated "inverted", additional wrong statements do not take off points.
(+ 1p) Advantage:
Option 1: Doesn't require a learning rate. Option 2: Basis **(0.5p)** - converges fast. **(+ 0.5p)** if stated "fewer iterations" / "mathematically" / "In theory" / curvature usage of the second derivative / other reasonable explanation or consequence. **(-0.5p)** for claiming that the algorithm would converge in 1 iteration. This is true in general only for linear models. **(-0.5p)** for stating that the optimizer guarantees the global optimum. (Only for convex, and that's true for all the optimizers. Irrelevnat).

Problem 5 Autoencoder (10 credits)



5.1 How do each of the elements (encoder, code, decoder) of autoencoders function?

0 1 2 3

(+ 1p): Encoder: performs dimensionality reduction feature for input.
(+ 1p): Code: This portion of the network only represents the compressed (low dimensionality) input.
(+ 1p): Decoder: Using a lossy reconstruction and the latent space representation, this decoder layer restores the encoded image to its original dimension. **(-0.5p)**: Given correct answers, for additional incorrect information.

5.2 You want to perform a semantic segmentation task on a small labeled dataset, and you also have access to a larger unlabeled image dataset. Explain how an autoencoder can help in that given task.

0 1 2

(+ 1p): Train an autoencoder to reconstruct the inputs using the whole data collection; Learn important features.
(+ 1p): Use the trained Encoder and the labeled data to train a segmentation model.
(-0.5p): Given correct answers, for additional incorrect information
Common Mistakes: use clustering for prediction on the labeled dataset.

0 1

5.3 If you use U-Net as your autoencoder model for semantic segmentation, what is a skip connection in the U-Net architecture? Don't forget to also offer some reasoning.

(+ 1p): encoder-decoder or down-up sampling layers + "concaticnation" / reasoning, e.g. "highway for gradients", pass fine-grained features, Avoid the vanishing gradient.
(0.5p) Only one part of the full answer (connection **xor** reasoning).
Common Mistakes: skip connections are **added** / incorrect sketch, e.g. connection not horizontal / Explaining ResNet skip connections / skip connections in general / Saying they would use Residual Blocks / Not mentioning encoder and decoder (text/sketch)

0 1 2

5.4 What are the differences between the autoencoder and the variational autoencoder in terms of the goal and loss?

Goal:
(+ 0.5p): AE - recovers input / Learn efficient embeddings of unlabeled data / Learn representation of input in latent space.
(+ 0.5p): VAE - provides a range of data in the latent space which is helping to generate new data / Generate new data that is from the same distribution as the input.

Loss:
(+ 0.5p): AE - Reconstructed output is close to the input / MSE/L1/L2 between encoder input and decoder output.
(+ 0.5p): VAE - Reconstruct output that is close to the input AND latent space is close to a Gaussian or normally distributed / KL-divergence + L1/L2.

0 1 2

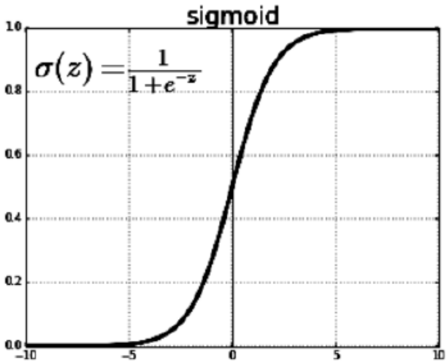
5.5 The decoder part of an autoencoder can also be used in a Generative Adversarial Network (GAN). What is the difference between an autoencoder and a GAN in terms of network architecture? (0.5p each) What is the goal of using the discriminator loss in GAN? (1p)

Network Architecture:
(+ 1p): AE: has an encoder **(0.5p)**, GAN: has a discriminator. **(0.5p)**
(0.5p): AE: cooperative networks vs. GAN: adversarial networks / Just mentioning one of them.
(0p): GANs: only decoder (info in the question desc) / supervised vs. unsupervised (Irrelevnat here) / Generate new data (Not archi related).

Discriminator loss:
(+ 1p): Output image look real in general / Correctly classifying fake and real data
(0.5p): Supervise generator / Improve generator / Tells "how good the generator is" / The goal of the loss is to classify real and fake.
(0p): Only talking about minmax game / Saying the discriminator's sole goal is to classify generated images as fake.

Problem 6 CNNs (10 credits)

You are training a neural network with 10 convolutional layers with the non-linearity shown below:



6.1 Explain the behavior of the gradient of the non-linearity with respect to very large inputs.

0 1 2

(+ 1p): gets almost zero, flat, plateaus, no slope / Sigmoid saturates
(+ 1p): Gradients get very small.
(-0.5p): Neurons instead of sigmoid / Gradients dies / killed / is zero
(0p): Only mentioning that sigmoid output is close to 1, without implication.

6.2 Why might this be a problem for training neural networks?

0 1 2

(+ 1p): Slows down training (or similar) **(+ 1p)**: No meaningful updates / prevents earlier layer from training. **(- 0.5p)**: Mentioning backprop, but not slow training
(0p): Mentioning multiplication with zero cause Vanishing Gradients in Sigmoid / Any inference answers (poor predictions etc.) / neurons / weights don't get updated.

6.3 Due to the problem mentioned in (6.2), modern architectures commonly adopt a different type of non-linearity. Name and draw this non-linearity, and explain why it helps solve the problem.

0 1 2

(1p): Name **(+ 0.5p)** + drawing **(+ 0.5p)**: ReLU, Leaky-ReLU, Parametric-ReLU
(+ 0.5p) for large, consistent, stable, no vanishing gradients, or similar
(+ 0.5p) for does not saturate for **large positiv** values/input (For Leaky/parametric - "large values" was enough.)

(0p): ReLU drawing if no visual left bold bar/line next to x-axis / for anything else that does not address the problems (vanishing gradients, saturation of positiv input.)

6.4 You are training the network for image segmentation. After 50 epochs, you come to the conclusion that the network is too large for such a task. Name two approaches to counteract the problem, without changing the convolutional layers of your network.

0 1 2

The model is too BIG! What problems could we detect?:

Vanishing/exploding gradients, Run out of memory, overfit

Solutions:

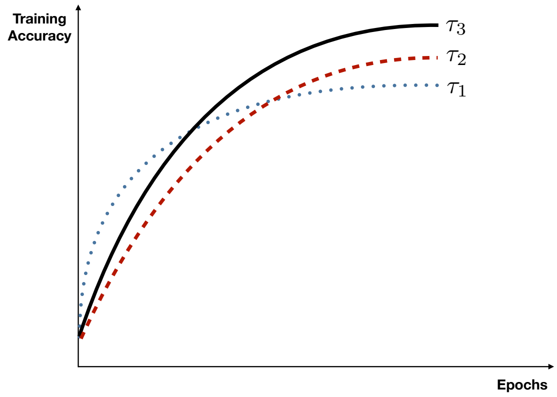
1. BN, Skip connecitons

2. Smaller batch sizes

3. Regularization (Dropout, wieght decay)

0 1 2

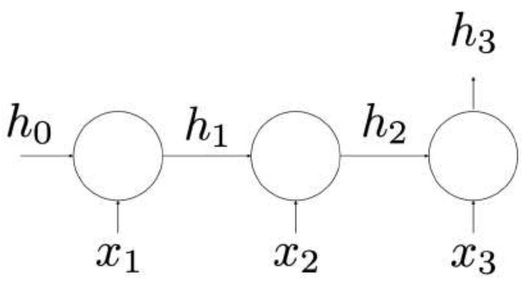
6.5 You adapt your network training accordingly, and now you are performing a grid search to find the optimal hyperparameters for vanilla stochastic gradient descent (SGD). You try three learning rates τ_i with $i \in \{1, 2, 3\}$, and obtain the following three curves for the training accuracy, all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



(+ 2p): $\tau_1 > \tau_3 > \tau_2$

Problem 7 LSTMs (9 credits)

7.1 Consider a vanilla RNN cell of the form $h_t = \tanh(V \cdot h_{t-1} + W \cdot x_t)$. The figure below shows the input sequence x_1, x_2, and x_3.



Given the dimensions $x_t \in \mathbb{R}^4$ and $h_t \in \mathbb{R}^{12}$, what is the number of parameters in the RNN cell? Neglect the bias parameter.

$4 \times 12 + 12 \times 12$ **(+ 1p)** $= 48 + 144 = 192$ **(+ 1p)**
(0.5p): Some reasonable formula, but just not the correct one.
Accepted: $4 \times 12 + 12 \times 12 + bias$ / $3 \times (4 \times 12 + 12 \times 12 + bias)$

7.2 If x_t is the 0 vector, then $h_t = h_{t-1}$. Discuss whether this statement is correct.

Case 1: **(+ 1p)**: False. **(+ 1p)**: After transformation with V and non-linearity $x_t = 0$ does not lead to $h_t = h_{t-1}$. / $h_t = Vh_{t-1}$ / a counterexample. **(0.5p)**: Only true when $h_{t-1} = 0$ / Hidden state updates every time.
Case 2: **(0p)** Yes + any statements.
Case 3: **(+ 2p)** Uncertain + correct statements, e.g. $\exists V, h : h = tanh(Vh)$ **and** $\exists V, h : h \neq tanh(Vh)$. **(+ 1p)** only if no valid statement.

7.3 Now consider the following **one-dimensional** ReLU-RNN cell.

$$h_t = \mathrm{ReLU}(V \cdot h_{t-1} + W \cdot x_t)$$

(Hidden state, input, and weights are scalars)
Calculate h_1, h_2 and h_3 where $V = 1$, $W = 2$, $h_0 = -3$, $x_1 = 1$, $x_2 = 2$ and $x_3 = 0$.

$$h_0 = -3$$
(+ 1p) $: h_1 = \mathrm{relu}(1 \cdot (-3) + 2 \cdot 1) = 0$
(+ 1p) $: h_2 = \mathrm{relu}(1 \cdot 0 + 2 \cdot 2) = 4$
(+ 1p) $: h_3 = \mathrm{relu}(1 \cdot 4 + 2 \cdot 0) = 4$

0 1 2

7.4 A Long-Short Term Memory (LSTM) unit is defined as

$$\begin{aligned} g_1 &= \sigma\left(W_1 \cdot x_t + U_1 \cdot h_{t-1}\right), \\ g_2 &= \sigma\left(W_2 \cdot x_t + U_2 \cdot h_{t-1}\right), \\ g_3 &= \sigma\left(W_3 \cdot x_t + U_3 \cdot h_{t-1}\right), \\ \tilde{c}_t &= \tanh\left(W_c \cdot x_t + u_c \cdot h_{t-1}\right), \\ c_t &= g_2 \circ c_{t-1} + g_3 \circ \tilde{c}_t, \\ h_t &= g_1 \circ c_t, \end{aligned}$$

where g_1, g_2, and g_3 are the gates of the LSTM cell.
1) Assign these gates correctly to the **forget** f, **update** u, and **output** o gates. (1p)
2) What does the value c_t represent in a LSTM? (1p)

(0.5p) for 2 correct $g_i, i \in \{1,2,3\}$.
(1p) for 3 correct $g_i, i \in \{1,2,3\}$.

$$\begin{aligned} g_1 &= \text{output gate} \\ g_2 &= \text{forget gate} \\ g_3 &= \text{update gate} \end{aligned}$$

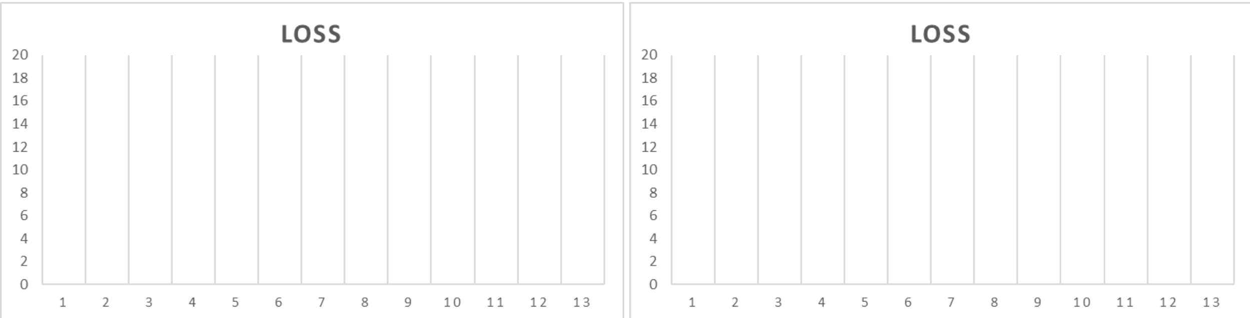
(+ 1p): c_t: cell state.

Problem 8 Training & Evaluation (9.5 credits)

8.1 A common way to divide your data is by splitting it into a train, validation, and test split. Explain the purpose of each split in detail and how we use each split (1p for each split). How much percentage of data do you commonly assign to each split (0.5p)?

Split: **(+ 0.5p)** examples: (60-20-20) (80-10-10) (90-5-5) (70-15-15) (70-20-10) (60-30-10) (50-30-20) or any acceptable split (training portion is $> 50\%$).
Training: **(+ 1p)**: used to train the parameters (weights) of the model. **(0.5p)**: train the model / calculate the gradients / optimize the loss.
Validation: **(+ 1p)**: hyperparameters tuning(searching/finding) / check the generalization performance / compute loss (performance) on unseen data / sanity check. **(0.5p)**: check the performance but do not emphasize "on unseen data". **(0p)**: only state "check if it is overfitting" / only state "validate the model".
Test: **(+ 1p)**: Final evaluation(verification)/ overall performance (to test hyperparameters). **(0.5p)**: only used once. **(0p)**: only state "test the model".

8.2 Explain the issues of overfitting and underfitting (1p each). Additionally, describe how your loss curves look like in each of the cases - draw the corresponding plots (1p each). (Make sure to label your curves).



(a) Underfitting (b) Overfitting

Curves:
(+ 1p): Undefitting: if both curves descend from the start to the end (could keep going) / both curves stay at a high position and gap between them is small.
(+ 1p): Overfitting: The training curve continues descending. The val curve first decreases then increases after some critical point / The val curve plateaus. **(0p)**: Only one curve each is drawn / if "test" curve is drawn / if val curve is below the training curve without sufficient explanations / if curves are drawn without labels val and training.

Explanations:

- Underditting: The model doesn't perform well on the training set.
- Overfitting: The model performs very well on the training set, but underperforms on unseen data.

0
1

8.3 A friend tries a new learning method and shows you this training loss plot. Name the method that was applied.

(+ 1p): learning rate scheduling (decay) / reducing learning rate / step (stepwise) decay.
(0.5p): Any imprecise term implying reducing the learning rate dynamically.
(0p): Adaptive learning rate method without further descriptions recognised as Adam / Momentum / RMSProp.

0
1

8.4 You successfully trained your model on the task of Image Classification with product images you collected from Amazon. It achieves good classification accuracy on your collected data. Now, you took pictures of objects yourself, however, your model misclassifies most objects. Give one reason, why your model performs poorly on these images you took.

(+ 1p): Domain gap between data distribution.

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