Neural Networks and Deep Learning – Summer Term 2019

Exam

1.) Neural networks and deep learning: General concepts

a) For each of the following definitions, find the corresponding technical term.
(For each correct element: 1 point, max: 12 points)

Description	*Technical term*
1.) Technique for iterative learning algorithms (like gradient descent) to stop iterations when the test error starts to increase	Early stoping
2.) Electrical impulse generated at the cell body (axon hillock) of a biological neuron which is propagated along the axon	Action Potential
3.) Ability (e.g. of a machine learning device) to make reasonable guesses for situations that are unknown (in comparison to data seen in training)	overfiting
4.) Layer in a convolutional neural network which combines the activations of several units (within the unit's receptive field in the previous layer) into a single unit of the current layer and reduces the size of its input e.g. by a factor of 2	
5.) Time period after emission of an action potential in which the neuron absolutely cannot emit another action potential, no matter how large a stimulus is applied	Absolute Refractory Period
6.) Amount by which a filter is shifted when computing a convolution	feature Map
7.) Technique (applied mostly in fully connected layers of artificial neural networks) to randomly delete neurons (and their weights), separately for each mini-batch, to avoid co-adaptation of neurons	Batch normalization
8.) Weight vector of a hidden neuron (shared among all neurons in that layer) in a convolutional neural network	
9.) (Non-)linear function in an artificial neuron transforming the neuron's input (postsynaptic potential) to its output	
10.) Artificial neural network with bidirectional data flow, i.e. including feedback loops	recurrent neural network
11.) Connection between two biological or artificial neurons	perception
12.) Phenomenon in learning where the model is not detailed enough to learn the characteristics of the input patterns	overfiting

Scanned with CamScanner

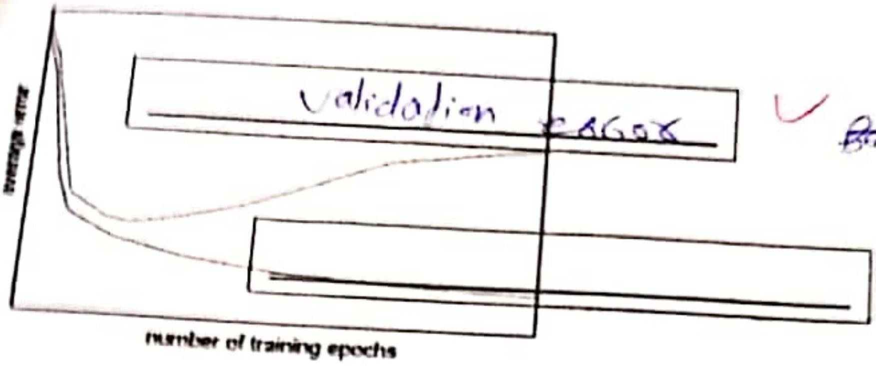
b) For each statement, mark for all alternatives which are true and which are false. Note: There is no limit on the number of true alternatives per statement.
(0,5 point for each correct answer, 0,5 point subtracted for any incorrect answer, Total: 8 points)

	True	False
i) In feedforward neural networks,		
(1) there is a bi-directional data flow,		X
(2) neurons in the same layer can be computed in parallel, whereas neurons of a subsequent layer have to "wait" until the previous layers have been computed,	X	
(3) there can only be connections between neighboring layers (but no "shortcuts"),	X	
(4) the neurons of different layers can have different activation functions.		X
ii) In recurrent neural networks,	True	False
(1) the network state is a function of time, since feedback loops may theoretically lead to "endless" neuron updates,		
(2) the output is computed layer by layer from input to output (and then no update can occur anymore),		
(3) an initial state may lead to oscillations, i.e. a sequence of network states that is periodically repeated,		
(4) there is a uni-directional data flow.		X
iii) Learning in artificial neural networks	True	False
(1) refers to specifying the parameters of the neural network (i.e. synaptic weights) such that a desired network behaviour is obtained,		
(2) is performed by presenting a set of examples to the network from which the parameters are learned,	X	
(3) needs a selection of a loss function and a network architecture for the given problem,	X	
(4) always finds the optimal solution for any problem, independent of the parameters and initial values.		X
iv) What is the advantage of using *multiple layers* in a feedforward neural network?	True	False
(1) The decision boundary can be non-linear.	X	
(2) Learning becomes substantially faster.		
(3) Margins (the "confidence" in classifying an input pattern) become larger.		
(4) Fewer training samples are required.		X

c) Consider a supervised machine learning problem (e.g. a classification task), where the available data is divided into two parts (as is generally being done in any machine learning problem). The following graph shows two error curves for the two parts of the data for some iterative learning algorithm (e.g. backpropagation) as a function of the number of training iterations (epochs).

- Identify the type of error curve for the two graphs (regarding the functional role the corresponding data part plays in the learning problem) and write the corresponding term into the two boxes. (2 points each, max. 4 points)
- Indicate in the figure the training iteration number which would yield the best model (recommended to be used in any future tests). (max. 2 points)
- Indicate for the model corresponding to the *maximal* number of training epochs (last iteration) whether it is an example for successful learning or whether it suffers from overfitting or from underfitting (one answer out of the three possibilities).

Answer (only one item!): Overfitting (correct solution: 2 points)



~~Batch error~~

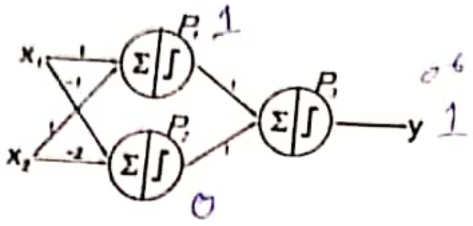
~~GNN~~

Ok 108

2.) Multi-layer perceptrons

Ex 1: 11/28

Consider the multi-layer perceptron in the figure below. The weights are indicated in the figure, the threshold (biases) of perceptrons P_1 and P_2 are 0, the threshold of perceptron P_3 is 1.5, and the Heaviside function is used for all units. The inputs x_1 and x_2 are real.



$x_1w_1 + x_2w_2$

$x_1w_1 + x_2w_2 = \theta$

a)
- Give the equation of the decision boundary (for the specific weights of this exercise) for the perceptron P_1.
- Further, indicate the area of the $x_1 - x_2$-plane that is classified positive by the perceptron P_1 (in the left figure below).

b)

(max. 4 points)

- Give the equation of the decision boundary (for the specific weights of this exercise) for the perceptron P_2.
- Further, indicate the area of the $x_1 - x_2$-plane that is classified positive by the perceptron P_2 (in the middle figure below).

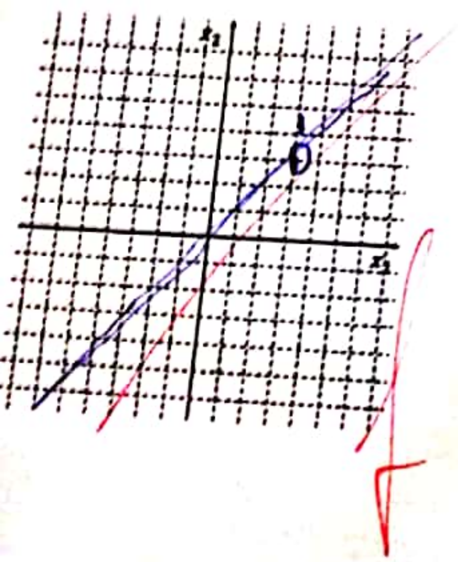
c)

(max. 4 points)

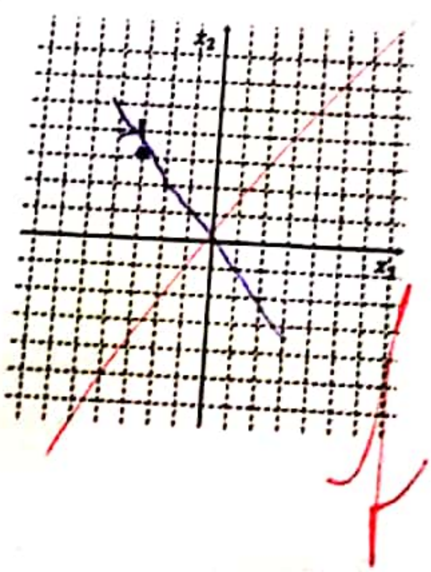
- Indicate the area of the $x_1 - x_2$-plane that is classified positive by the perceptron P_3 (in the right figure below),
- Explain your decision

(max. 4 points)

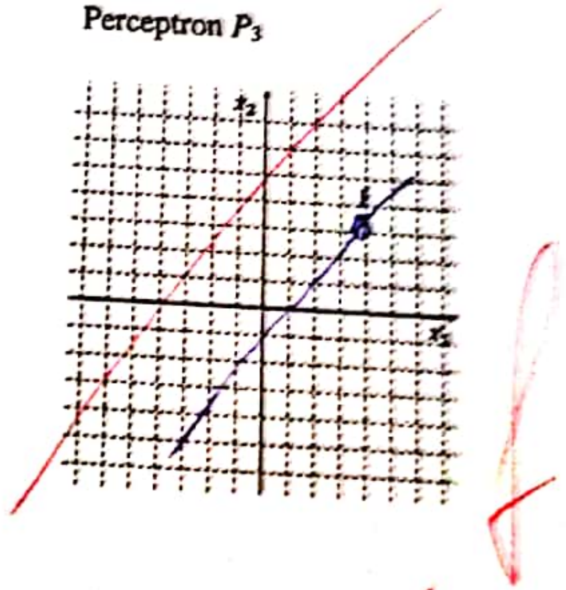
Perceptron P_1



Perceptron P_2



Perceptron P_3



0/12

d) Calculate the perceptron output y (=P_3) for the input $(x_1, x_2) = (3, -1)$, if *all* activation functions (for P_1, P_2 and P_3) are replaced by linear activation functions. Insert your results into the following table: (max. 6 points)

Input (x_1, x_2)	$y_1 (= P_1)$	$y_2 (= P_2)$	Output $y (= P_3)$
(3, −1)	2 mean 1	−1 = 0	2 mean 1

0/6

e) For each statement, mark for all alternatives which are true and which are false. Note: There is no limit on the number of true alternatives per statement.
(0,5 point for each correct answer, 0,5 point subtracted for any incorrect answer, Total: 4 points)

	True	False
i) Assume you want to use a multi-layer perceptron to solve an arbitrary classification problem (using a corresponding activation function and training loss).	True	False
(1) A multi-layer perceptron with two layers of trainable weights (i.e., input layer, single hidden layer, output layer) can solve any arbitrary classification problem.		X
(2) A multi-layer perceptron with three layers of trainable weights (i.e., input layer, two hidden layers, output layer) can solve any classification problem.		X
(3) A multi-layer perceptron with four layers of trainable weights (i.e., input layer, three hidden layers, output layer) or more can be more efficient to solve the classification problem, but may be prone to overfitting.		X
(4) A multi-layer perceptron is not appropriate to solve a classification problem.	X	
ii) Assume you want to use a multi-layer perceptron to solve a regression problem, i.e., approximating an arbitrary, continuous function on a compact interval (using a corresponding activation function and training loss)	True	False
(1) A multi-layer perceptron with two layers of trainable weights (i.e., input layer, single hidden layer, output layer) can match the target function with any desired accuracy.	X	
(2) A multi-layer perceptron with three layers of trainable weights (i.e., input layer, two hidden layers, output layer) can match the target function with any desired accuracy.	X	
(3) A multi-layer perceptron with four layers of trainable weights (i.e., input layer, three hidden layers, output layer) or more can be more efficient to solve the regression problem, but may be prone to overfitting.	X	
(4) A multi-layer perceptron is not appropriate to solve a regression problem.		X

1/4

3.) Learning in neural networks

a) For each network type (with indicated activation function of the output layer), give an appropriate loss function (2nd column in table below) and indicate (3rd column in table below) whether the learning algorithm has to be applied in an iterative fashion ("iterative") or whether a solution in closed form exists ("closed form", i.e., the network parameters can be calculated directly from the training data, without iterating a learning algorithm). (max. 8 points)

Network type	Loss function	Iterative / closed form?	
Single-layer perceptron, activation function: Heaviside	$\phi(w_1 x_1 + w_2 x_2 - \Theta)$?		
Single-layer perceptron, activation function: Linear	Loss Mean square error ✓ $f(z) = z$	Regression	Wrong question
Multi-layer perceptron, output activation function: Logistic	~~Log Likelihood~~ Cross entropy $f(z) = \frac{1}{1+e^{-z}}$	Binary	
Multi-layer perceptron, output activation function: Softmax	Log-Likelihood $f(z) = \frac{e^{z_v}}{\sum_{k=1}^{m} e^{z_k}}$ ✓	classification	

3/8

b) Indicate (by checking the appropriate column) whether the following statements are true or false: (For each correct answer: 1 point, for each incorrect answer: 1 point subtracted, min.: 0 points, max: 8 points)

Statement	true	false	
A "flat" neuron activation function or error function may pose problems to neural network training since the resulting weight modification may be small.	X		✓
The stochastic gradient descent learning algorithm always finds the global minimum of the loss function.	X		✗
Using weight regularization in stochastic gradient descent (e.g. L1 or L2) guarantees to find network weights yielding a low generalization error (e.g., smaller than 0.05).	X		✗
Using a ReLU activation function may help to reduce the effect of the vanishing gradient problem.		X	✗
The size of the mini-batch in stochastic gradient descent may influence the convergence of the learning algorithm (i.e., whether and how fast it converges).	X		✓
The initial values of the weights and bias parameters may have an influence on the result of the learning algorithm.		X	✗
Second order learning methods are especially suited for non-convex loss functions and only few training data.			-
Learning with momentum generally helps to stabilize learning (by reducing the effect of parameter oscillations) and to speed up learning.			-

0

c) Indicate (by checking the appropriate column) whether applying the following method may help to reduce the *test error* of a neural network (on a very large test corpus; mark "yes") or not ("no"). (For each correct answer: 1 point, for each incorrect answer: 1 point subtracted, min.: 0 points, max: 6 points)

Method (does applying this method may reduce the test error?)	Yes	No
Data augmentation	X	
Normalizing the input data only on the training data (bot not on the test data)		X
Weight regularization	X	
Reducing the size of the training set		X
Applying dropout to fully connected layers	X	
Initializing weights and biases to the same value for all neurons of a given layer		X

6

d) Name three extensions of the standard stochastic gradient descent learning algorithm.
(2 points for each correct answer; max. 6 points)

Learning with momentum ✓

adapted learning rate ✓

Second order method ✓

6/6

75/7

4.) General questions about neural networks (multiple choice)

a) For each question, mark a *single* answer that represents the *best* possible reply to the question (sometimes there might be no obvious "wrong" answer, but one answer should always be better than the others). (1 point for each correct alternative)

i) Training and testing: An artificial neural network may be trained on one data set and tested on a second data set. The system designer can then experiment with different numbers of hidden layers, different numbers of hidden units, etc. For real world applications, it is therefore important to use a *third* data set to evaluate the final performance of the system. Why?

(a) The error on the third data set provides a better (unbiased) estimate of the true generalization error.
(b) The error on the third data set is used to choose between lots of different possible systems.
(c) It's not important – testing on the second data set indicates the generalization performance of the system.
(d) The error on the third data set is used to update the system parameters via backpropagation.

Answer: b,c & a

ii) Training and testing: Which one of the following statements is the best description of leaving-one-out training (*k*-fold cross-validation, where *k* corresponds to the number of training samples)?

(a) Randomly pick some of the training data for training (e.g., 70%) and use the rest for testing.
(b) Calculate estimates for the mean vectors of each class using the training data.
(c) Use one of the training samples for testing and the remaining samples for training. Repeat for all samples of the training set.
(d) Apply stochastic gradient descent with a mini-batch size of 1 .

Answer: b,c & c

iii) Biological neurons: Which of the following statements are true for typical neurons in the human brain?

→ (a) Electrical potential is summed in the neuron.
→ (b) When the potential is bigger than a threshold, the neuron emits an action potential through the axon.
→ (c) The neurons are connected to each other via synapses, which transmit the action potential to dendrites of other neurons.
(d) All of the above answers.

iv) Perceptron ... w is the weight ... the target ...

iv) Perceptrons: Which of the following equations is the best description of perceptron learning? $\mathbf{w}$ is the weight vector, t the iteration index, $\mathbf{x}^{(\mu)}$ the input vector, η the learning rate, $y^{(\mu)}$ the target output, $\hat{y}^{(\mu)}$ the perceptron output, and $\Delta\mathbf{w}(t) = \mathbf{w}(t+1) - \mathbf{w}(t)$:

(a) $\Delta\mathbf{w}(t) = \eta \cdot \hat{y}^{(\mu)} \cdot \mathbf{x}^{(\mu)}$
(b) $\Delta\mathbf{w}(t) = \eta \cdot (y^{(\mu)} - \hat{y}^{(\mu)}) \cdot \mathbf{x}^{(\mu)}$
(c) $\Delta\mathbf{w}(t) = \eta \cdot (\mathbf{x}^{(\mu)} - \mathbf{w}(t))$
(d) $\Delta\mathbf{w}(t) = \eta \cdot y^{(\mu)} \cdot (\mathbf{x}^{(\mu)} - \hat{y}^{(\mu)} \mathbf{w}(t))$

$w(t+1) = w(t) + \eta \cdot (d^i - y^i) \cdot x$

learning rate   target output   network op.

Answer: b ✓

v) Neural network learning: What is *backpropagation*?

(a) It is the transfer of error back through the network to adjust the inputs.
(b) It is the transfer of error back through the network to allow the weights to be adjusted.
(c) It is the transfer of error back through the network using a set of recurrent connections.
(d) It is the transfer of outputs back from the hidden layer to the input layer using a set of recurrent connections.

Answer: d f b

vi) Neural network learning: Which of the following statements is the best description of *early stopping*?

(a) Train the network until a local minimum in the error function is reached.
(b) Evaluate the network on the test data after every training iteration (epoch). Stop training when the generalization error starts to increase.
(c) Add a momentum term to the weight update in the update equation for the weights, so that training converges more quickly.
(d) Use a faster version of backpropagation, such as the "Quickprop" algorithm.

Answer: b ✓

vii) Neural network learning: Which of the following statements is the best description of *overfitting*?

(a) The network becomes "specialized" and learns the training set too well.
(b) The network cannot predict the correct outputs for test examples which are outside the range of the training examples.
(c) The network does not contain enough adjustable parameters (e.g., hidden units) to find a good approximation to the unknown function which generated the training data.
(d) The network assigns wrong outputs to test examples which lie in between most of the training examples.

Answer: C f a

viii) Multi-layer perceptrons: What is the most general type of decision region that can be formed by a multi-layer perceptron with no hidden layers?

(a) Decision regions separated by a line, plane or hyperplane.
(b) Convex decision regions – for example, the network can approximate any Boolean function.
(c) Arbitrary decision regions – the network can approximate any function (the accuracy of the approximation depends on the number of hidden units).
(d) None of the above answers.

Answer: a ✓

ix) Which of the following neural networks would you use for time series prediction, e.g. weather forecasting (where the input at test time consists of a single time step to predict the next time step)?

(a) Multi-layer perceptron.
(b) Autoencoder.
(c) Long short-term memory.
(d) None of the above answers.

Answer: b,c & c

x) Which of the following techniques is NOT a strategy for dealing with local minima in the backpropagation algorithm?

(a) Add random noise to the weights or input vectors during training.
(b) Training with adaptive learning rate (which is large at the beginning of training).
(c) Test with a committee of networks.
(d) Train and test using cross-validation.

Answer: d ✓

xi) A (fully connected) single-layer perceptron has 6 input units and 3 output units. How many learnable parameters does this network have?

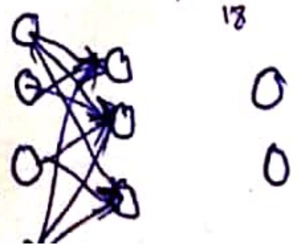
(a) 9
(b) 12
(c) 18
(d) 21

Answer: 18 c & d

xii) A (fully connected) simple recurrent network (no LSTM) has 4 input units, 3 hidden units and 2 output units; the recurrent connections are within the hidden layer. How many learnable parameters does this network have?

(a) 20
(b) 23
(c) 32
(d) 37

$4 \times 3 + 3 \times 2$
$12 + 6$
18



Answer: 32 c ✓

formed

Boolean

xiii) When does a feedforward neural network with two hidden layers become a deep learning model?

(a) When you add more hidden layers and increase the depth of the neural network.
(b) When you increase the dimensionality of the training data.
(c) When the problem is an image recognition problem.
(d) When you add recurrent connections to the hidden layers.

Answer: a ✓

xiv) Which of the following is true about model capacity (where model capacity means the ability of a neural networks to approximate arbitrary functions)?

(a) As the number of hidden layers increases, the model capacity increases.
(b) As the dropout rate increases, the model capacity increases.
(c) As the learning rate increases, the model capacity increases..
(d) None of these.

Answer: b f a

xv) In a neural network, which of the following techniques is used to deal with overfitting?

(a) Dropout.
(b) Regularization.
(c) Batch normalization.
(d) All of the above.

Answer: C f d

xvi) Which of the following gives non-linearity to a neural network?

(a) Stochastic gradient descent.
(b) Rectified linear unit.
(c) Convolution function.
(d) None of the above.

Answer: a f b

xvii) Which of the following architectures has feedback connections?

(a) Variational autoencoder.
(b) Convolutional neural network.
(c) Long short term memory.
(d) Deep convolutional generative adversarial network.

Answer: b f c

7/17

b) For each question, mark a single answer that represents the best possible reply to the question (sometimes there might be no obvious "wrong" answer, but one answer should always be better than the others).
(2 points for each correct alternative)

i) What is the correct order regarding the following steps in using a gradient descent algorithm?

1. Calculate error between the actual value and the predicted value.
2. Reiterate until you find the best weights of network.
3. Pass an input through the network and get values from output layer.
4. Initialize random weight and bias.
5. Go to each neurons which contributes to the error and change its respective values to reduce the error.

(a) 1 -> 2 -> 3 -> 4 -> 5
(b) 5 -> 4 -> 3 -> 2 -> 1
(c) 3 -> 2 -> 1 -> 5 -> 4
(d) 4 -> 3 -> 1 -> 5 -> 2

Answer: d ✓

ii) Batch normalization is helpful because

(a) it normalizes (changes) all the input before sending it to the next layer.
(b) it returns back the normalized mean and standard deviation of the weights.
(c) It is a very efficient backpropagation technique.
(d) None of the above.

Answer: a, b a

iii) For a classification task, instead of random weight initializations in a neural network, se set all the weights and biases to zero. Which of the following statements is true?
we

(a) There will not be any problem and the neural network will train properly.
(b) The neural network will train, but all the neurons (in the same layer) will end up with the same parameters, so they will recognize the same thing.
(c) The neural network will not train as there is no net gradient change.
(d) None of the above.

Answer: c b

2/6

5.) Convolutional neural networks

a) Mention 5 *different* specific types of convolutional neural networks in the order of appearance (which roughly corresponds to the depth / complexity of the network), together with one characteristic element for each architecture (which is used by this architecture, but not by earlier architectures).
(1 point for each correct name, 1 point for each correct characteristic element plus max. 2 additional points for correct order; max. 12 points)

Order	Name	Characteristic element
1.)	LeNet ✓	hand written digit recognition 32×32×1 input ✓
2.)	Alex Net ✓	Heavy data augmentation In this Relu first time use ✓
3.)	ZF Net ✓	In this using more smaller filters instead of large one ✓
4.)	VGG ✓	In this using smaller filter and more layers ✓
5.)	NiN ✓	

11/

b) Consider a CNN with the following architecture:

INPUT → Conv → ReLU → MaxPool ,

where "Conv" denotes a convolutional layer with the filter (kernel) given below, stride 1 and no padding, "ReLU" denotes the rectified linear unit activation and "MaxPool" denotes a pooling layer with 2 × 2 max pooling, stride 2 and no padding.

Given is the following input and the following filter (kernel) function:

$3+0+0+1=4$
$0-2+0+0=-2$
$0+2+0+3=5$
$-1+0+0-2=-3$

4	-2
5	-3

Input: $W=3$

3	0	1
0	−1	0
−2	−3	2

Filter (kernel): $K=2$

1	−2
0	−1

Further assume that all biases are set to 0 for this exercise. Calculate the output of the CNN (including *all* intermediate stages). (max.: 7 points)

$$= \frac{W-K+2P}{S}+1 = \frac{3-2+2(0)}{1}+1 = \frac{3-2}{1}+1 = 1+1 = 1+1=2$$

Conv ✓

4	-2
5	-3

4P

ReLu

4	0
5	0

✓ 2P

MaxP

5	0

Scanned with CamScanner

6.) General properties of neural networks (multiple choice)

a) Indicate (by checking the appropriate column) whether the following statements are true or false: (For each correct answer: 1 point, for each incorrect answer: 1 point subtracted, min.: 0 points, max: 12 points)

Statement	true	false
1.) In multilayer perceptrons, at least three hidden layers are needed to approximate arbitrary continuous functions over a compact interval with arbitrary accuracy.		X
2.) The solution delivered by the perceptron learning algorithm does depend on the sequence of pattern presentation.	X	
3.) If the same training pattern occurs in several iterations of the perceptron learning algorithm, then the learning task cannot be solved by a perceptron.		X
4.) When applying stochastic gradient descent, picking a learning rate that is very small has no disadvantage and can only speed up learning.		X
5.) When applying stochastic gradient descent, if we reduce the learning rate during iterations (and run stochastic gradient descent long enough), it's possible that we may find a set of better parameters than with constant larger initial learning rate.	X	
6.) If we want stochastic gradient to converge to a (local) minimum rather than wander or "oscillate" around it, we should slowly increase the learning rate over time.		X
7.) If we plot the cost function (averaged over the last 1000 examples) and stochastic gradient descent does not seem to be reducing the cost, one possible problem may be that the learning rate is poorly tuned.		X
8.) If the number of hidden layers in a multi-layer perceptron is increased, the test error always decreases.		X
9.) Suppose a convolutional neural network is trained on ImageNet dataset (object recognition dataset). This trained model is then given a completely white image as an input. The output probabilities for this input would be equal for all classes.		X
10.) When pooling layer is added in a convolutional neural network, translation invariance is preserved.	X	
11.) When the amount of training data is too big to handle in RAM (random-access memory) simultaneously, full batch gradient descent is a more advantageous learning strategy than stochastic gradient descent.	X	
12.) A multi-layer perceptron should have the same number of units in the input layer and the output layer.		X

8/12

Calculations

ANN

4	-2
5	-3

Relu

4	0
5	0

Max pool

4	5

or

5	0

c) Consider a CNN with the following architecture:

INPUT → Conv1 → Pool → Conv2

and the following specifications:

Layer	Specification	Dimension (width × height × depth)
INPUT	32 × 32 gray scale image	32 × 32 × 1
Conv1	5 × 5 filter, no padding, stride 1, 6 feature maps	5×5×6
Pool	2 × 2 filter, no padding, stride 2, max pooling	2×2 ×1
Conv2	3 × 3 filter, no padding, stride 1, 10 feature maps	5×5×10
Conv3	1 × 1 filter, no padding, stride 1, 2 feature maps	5×5 ×2

f

For each layer, insert the dimension of the *output* of that layer in the form (width × height × depth) into the table above. Then, calculate the number of trainable parameters (synaptic weights *plus* biases) involved in each layer and insert the corresponding equation and the result into the table below: (max. 10 points)

Layer	Number of trainable parameters (calculation)	Trainable parameters (result)
Conv1	5*5*6 +1	151
Pool	no trainable parameters	—
Conv2	(5 ×5×10+1)2	502
Conv3	(5 ×5×2+1)	51

f

1 250
251
2
502

6)
25
6
150

00l

Ex.5: 18/2