Resit Neural Networks

Wojtek Kowalczyk
wojtek@liacs.nl

29.06.2018

It is a closed book exam: you are not allowed to use any notes, books, calculators, smartphones, etc. For each question you will get some points; additionally you will get 10 points for free. The number of points attached to each question reflects the (subjective) level of question's difficulty. In total you may get 100 points. The final grade for the exam is the total number of points you receive divided by 10.

The exam consists of a number of "multiple questions - single choice answer" questions. It means that for each question you should select exactly one answer. For every correct choice you get some points; for an incorrect answer or no answer you will get 0 points.

Mark your choices by crossing the selected option. In case you want to "undo" your choice put a circle around the cross. For example, on the left side the option **b** is selected; on the right side nothing is select – the selection of **b** is "undone":

If you think that your marking is no longer readable, put your final choice(s) on the left margin (e.g., by writing "a" if you want to select "a"). Finally, you are free to add to your answers your comments (in a free space). We don't know yet if and how we will process such comments, but it may help us to adjust the exam grade (up or down – depending on the comment).

Before starting answering the questions, fill in the following entries:

Name:

Student number:

Study type (ICT, Astronomy, …):

5 pts	**ALVINN**
Question	In 1989 D.A. Pomerleau presented a system ALVINN that could learn to drive a car. What is the closest estimate of the number of tunable parameters used by this system?
Answer options	a) 100 **b) 1000** **c) 10000** d) 100000 e) More than 100000 The network had 30*32 inputs, 4 hidden nodes and 30 output nodes, so it had about 4000 (30*32*4+4*30) tunable parameters. As 4000 is close both to 1000 and 10000, both answers are considered to be correct.

3 pts	**Network types**
Question	What network type is most suitable for removing noise from images?
Answer options	a) A multi-layer perceptron b) A convolutional network c) A recurrent network **d) An autoencoder** e) An LSTM network Denoising autoencoder is just an autoencoder that is trained on noisy data.

12 pts	**Bayes' Rule**
Question	Suppose that you have to develop, with help of the Bayes' Rule, a simple system that decides if an input image of a fruit is a banana or an apple. The system should use only one binary feature of input images: the color of the fruit, which may be either yellow or green, and no other colors (or fruits) are possible. As a training set a random sample of 1000 bananas and 2000 apples is given. It turns out that in this set 800 bananas are yellow and 1200 apples are green. Additionally, let us assume that in reality apples are 3 times more frequent than bananas (so the training set is biased) - this fact should also be taken into account when building the system. What probability estimate of P(Banana|Yellow) should be produced by your system?
Answer options	a) 0.2 b) 0.3 **c) 0.4** d) 0.6 e) Something else Apples are 3 times more frequent than bananas, so P(Apple)=0.75, P(Banana)=0.25. Moreover, P(Yellow|Banana)=800/1000=0.8 and P(Yellow|Apple)=800/2000=0.4. Therefore P(Apple|Yellow)=P(Yellow|Apple)*P(Apple)/NF = 0.4*0.75/NF=0.3/NF and P(Banana|Yellow)=P(Yellow|Banana)*P(Banana)/NF=0.8*0.25/NF=0.2/NF, where NF is a normalization factor that has to satisfy 0.3/NF+0.2/NF=1, so NF=0.5 and **P(Banana|Yellow)=0.2/0.5=0.4.**

5 pts	**Discriminant functions**
Question	Let us suppose that $f_A(x)$ and $f_B(x)$ are discriminant functions for sets A and B, as required by the definition of *multi-class separability*. Which of the following pairs of functions does not necessarily discriminate sets A and B?
Answer options	a) *sigmoid*$(f_A(x))$ and *sigmoid*$(f_B(x))$ b) *exp*$(f_A(x))$ and *exp*$(f_B(x))$ **c) $(f_A(x))^2$ and $(f_B(x))^2$** d) $(f_A(x))^3$ and $(f_B(x))^3$ The quadratic function is not monotonic!

5 pts	**Linear separability and multi-class linear separability**
Question	Let us suppose that three sets: A, B, and C, are *pairwise linearly separable*, i.e., A and B, A and C, and B and C are linearly separable. Which of the following statements is always true:
Answer options	a) A, B and C are also separable according to the multi-class linear separability definition **b) A, B and C are not necessarily separable according to the multi-class linear separability definition** c) If A, B and C are not convex then they are not separable according to the multi-class linear separability definition d) If A, B and C are convex then they are separable according to the multi-class linear separability definition Consider A={(x,y): x>1}, B={(x,y): -1<x<1}, C={(x,y): x<-1}. Obviously, A is linearly separable from B, B from C, A from C, but it is impossible to provide three linear functions fA, fB, fC that would separate all three sets as required by the definition of the multi-class linear separability.

5 pts	**Fighting Overfitting**
Question	During the course you've learned several methods of reducing overfitting. Which of the following methods does NOT reduce overfitting:
Answer options	a) L2 regularization **b) Nesterov momentum** c) Dropout d) Data augmentation Nesterov momentum speeds us the convergence and has nothing to do with overfitting

10 pts	**Backpropagation and its complexity**
Question	Let us consider a multi-layer network with K inputs, L nodes in the hidden layer and M nodes in the output layer. For simplicity, we assume that the nodes do not use biases. Additionally, let us assume that all the nodes in the hidden and output layer use the ReLU function.

	How many multiplications are needed to compute the output of the network for a single input?
Answer options	a) K +L+M b) K*L*M c) K*L+M **d) L*(K+M)** e) None of the above There are K*L connections between the input and the hidden layer and L*M connections between the hidden and the output layer.

4 pts	**SGD with Nesterov momentum**
Question	Suppose that the gradient descent algorithm with Nesterov momentum is used for finding a minimum of a function $f(x)$. Let α denotes the learning rate, β denotes the momentum rate, $g(x)$ denotes the gradient of the function being optimized (i.e., the vector of partial derivatives of f at x), and $d(x)$ denotes the direction of the last step, i.e., $d(x_k) = x_k - x_{k-1}$, where $x_1, x_2, \ldots, x_k$ are consecutive points visited by the algorithm. What should be the next point visited by the algorithm:
Answer options	*a)* $x_{k+1} = x_k + \alpha g(x_k) + \beta d(x_k)$ *b)* $x_{k+1} = x_k - \alpha g(x_k) + \beta d(x_k)$ ***c)*** $\boldsymbol{x_{k+1} = x_k - \alpha g(x_k + \beta d(x_k)) + \beta d(x_k)}$ *d)* $x_{k+1} = x_k + \alpha g(x_k + \beta d(x_k)) + \beta d(x_k)$ *e)* $x_{k+1} = x_k + \alpha g(x_k + \beta d(x_k))$ *f)* $x_{k+1} = x_k - \alpha g(x_k + \beta d(x_k))$ *g) none of the above* By the definition of Nesterov momentum

4*5 pts	**Convolutional Networks**
Question	Let us consider a simple convolutional network for classifying images of size 10x10: • the input layer **I** has the size 10x10, • the first convolutional layer **C** consists of 5 feature maps, generated by 5 convolutional filters of size 3x3, shifted by 1 pixel at a time (stride=1), using the "valid" scheme, i.e., filters move within the image boundaries, • the pooling layer **P** consists 5 maps that are computed by a 2x2 non-overlapping receptive field that computes the max() function, • the pooling layer **P** is fully connected to a hidden layer **H** with 100 nodes, • the hidden layer **H** is fully connected to an output softmax layer **O** with 10 nodes. For simplicity we assume that no biases are used. Answer the following questions, writing down formulas that lead to these values, like 10*5*2 instead of 100. **a)** What is the number of nodes in each layer of the network: **C**: 8*8*5 (there are 5 maps, each map has size 8*8)

	P: 4*4*5 (each pooling filter reduces width and height of filters by factor 2) **H**: 100 (this value is given) **b)** What is the number of weights between **I** and **C**: 8*8*5*9 (each node in C gets input from 3*3 input nodes) **C** and **P**: 4*4*5*2*2 (each node in P gets input from 2*2 nodes of C) **P** and **H**: 4*4*5*100 **H** and **O**: 100*10 **c)** what is the number of trainable weights between **I** and **C**: 5*3*3 (each filter needs 9 trainable parameters) **C** and **P**: 0 (max function is parameterless) **P** and **H**: 4*4*5*100 (all connections are trainable) **H** and **O**: 100*10 (all connections are trainable) **d)** How many trainable weights would we have if all the nodes of the network were fully connected (**I** to **C**, **C** to **P**, **P** to **H**, and **H** to **O**)? From (a) we know the number of nodes in C and P; we also know that I=10*10, H=100, O=10, so the total number of weights is: I*C + C*P + P*H + H*O.

8 pts	**Vanilla Recurrent Neural Networks**
Question	During the course we discussed a simple recurrent network that was used for language modelling. The network used 8000 input nodes to represent 8000 words (using "one-hot" encoding), 8000 output nodes to represent probability distribution over "the next word in the sentence", and 100 hidden nodes. No bias parameters were used. What is the total number of trainable weights used by this network?
Answer options	a) 2*100*8000 b) 3*100*8000 c) 2*100*8000 + 100 **d) 2*100*8000 + 100*100** e) None of the above There are 8000*100 connections between input and hidden, 100*8000 between hidden and output and 100*100 between hidden and "next hidden".

3 pts	**LSTM Networks**
Question	What is the biggest advantage of LSTM networks over Vanilla Recurrent Networks? Select only one answer which is in your opinion the best one.
Answer options	a) Faster convergence? b) Ability to remember the most recent states? c) Ability to remember remote states? **d) Ability of learning which remote and recent information is relevant for the given task and using this information to generate output** e) None of the above

5 pts	**RBM networks for recommender systems**
Question	During the course we discussed an application of an RBM network to the Netflix Challenge, where the task was to predict a rating a user could give to a movie. How the output of the network was translated into a rating between 1 and 5?
Answer options	a) By selecting the node with the highest activation and returning the corresponding rating b) By averaging all the outputs **c) By taking the weighted average of the output predictions** d) The network was trained to solve a regression problem, so it had one output node that was directly returning the rating e) None of the above

5 pts	**Batch Normalization**
Question	What is batch normalization? Select only one answer which is in your opinion the best one.
Answer options	a) The process of normalizing each batch of data by dividing it by its mean and standard deviation. b) The process of scaling and shifting each batch of data before the batch is used for training. c) The process of finding an optimal transformation of each batch that is executed before the network is trained. **d) The process of finding an optimal transformation of each batch, layer after layer, that is optimized during the training process.** e) None of the above.