Exam Neural Networks

Wojtek Kowalczyk
w.j.kowalczyk@liacs.leidenuniv.nl

05.06.2019

It is a closed book exam: you are not allowed to use any notes, books, calculators, smartphones, etc. 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 questions with a "single choice answer". It means that for each question you should select exactly one answer. For every correct choice you get some points; for an incorrect choice or no choice you 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":

a) bla bla
b) ble ble
c) bli ble
a) bla bla
b) ble ble
c) bli ble

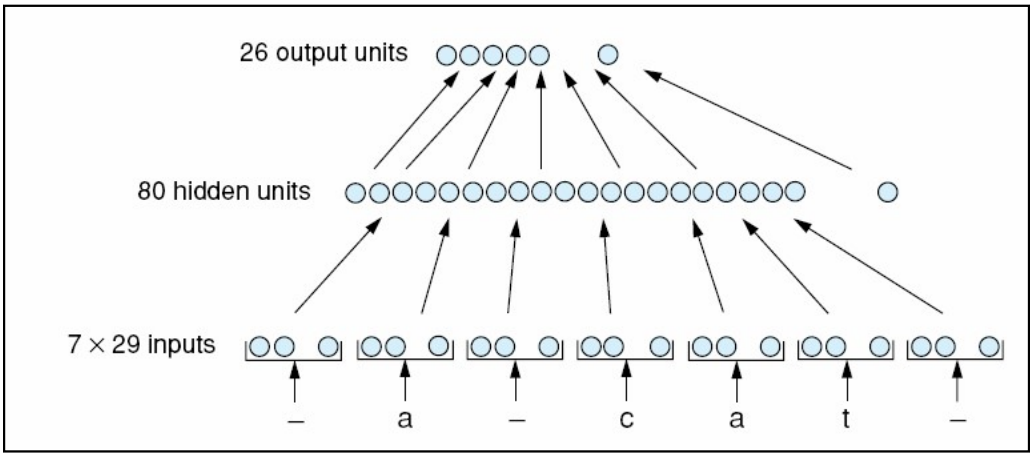
If you think that your marking is no longer readable, put your final choice(s) on the left margin (e.g., by writing "a" if you want to select "a"). Finally, you are free to add to your answers your comments (in a free space). Your comments may help us to adjust the exam grade (up or down – depending on the comment).

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

Name:

Student number:

Study type (ICT, Astronomy, …):

10 pts	**Q1: NetTalk**
Question	In 1987 Sejnowski and Rosenberg published a paper on their Text2Speech system called NetTalk. Their network has 7*29 input nodes, 80 hidden nodes and 26 output nodes. For simplicity, let us assume that no biases are used. How many multiplications are needed to compute the output of the network on the input that represents the phrase “a cat”? We assume here that multiplications are very expensive and don’t multiply two numbers when one of them is 0.  26 output units 80 hidden units 7 × 29 inputs – a – c a t –
Answer options	a) 7*29*80+80*26 b) 29*80+80*26 c) 4*80+80*26 d) 5*80+80*26 e) None of the above Both answers (c and e) are correct. We may assume that the space character is represented by a vector of seven zeros and then the input vector contains exactly 4 ones (at places that represent the characters ‘a’, ‘c’, ‘a’, ‘t’). Consequently, only 4*80 multiplications are needed to compute the activation of the hidden layer, followed by another 80*26 multiplications needed to compute the activation of the output layer, so the answer (c) can be viewed as the correct one. Alternatively, we may assume that the space character is treated as a normal character (actually, the original NetTalk paper makes this assumption) and then we need 7*80 multiplications to find the activation of the hidden layer, plus 80*26 multiplications to find the activation of the output layer, so the answer (e) is correct. Finally, let us note that the input vectors contain only 0’s and 1’s, so there is no need at all to multiply these values by weights leading from the input layer to the hidden layer: 0*x=0 and 1*x=x.

9 pts	Q2: 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 1000 apples is given. It turns out that in this set 800 bananas are yellow and 600 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)=400/1000=0.4 (as we know that P(Green|Apple)=0.6). Therefore: P(Apple|Yellow)=P(Yellow|Apple)*P(Apple)/NF = 0.4*0.75/NF=3/NF and P(Banana|Yellow)=P(Yellow|Banana)*P(Banana)/NF=0.8*0.25/NF=2/NF, where NF is a normalization factor that has to satisfy 3/NF+2/NF=1, so NF=5 so P(Apple|Yellow)=3/5=0.6, so P(Banana|Yellow)=0.4.

3 pts	Q3: Multi-class linear separability
Question	Which definition of the *multi-class linear separability* concept (restricted here to 3 classes) is correct: Sets A, B, C are linearly separable if:
Answer options	a) Any two of them are linearly separable. b) Any one of them is linearly separable from the union of the remaining two sets. c) There are 3 linear functions f_A, f_B, f_C such that: for all $x \in A$ $f_A(x)$ is bigger than $f_B(x)$ and $f_C(x)$, and for all $x \in B$ $f_B(x)$ is bigger than $f_A(x)$ and $f_C(x)$, and for all $x \in C$ $f_C(x)$ is bigger than $f_A(x)$ and $f_B(x)$. d) None of the above definitions is correct.

5 pts	**Q4: Cover’s Theorem**
Question	Let us consider a collection of 500 images of size 16x16, where each pixel is a randomly generated number between 0 and 1. Suppose, that these images are split into two classes A and B, at random. What is the probability that classes A and B are linearly separable?
Answer options	a) About 0.0 b) About 0.5 **c) About 1.0** d) None of them The ratio N/(d+1) = 500/257 < 2, so by Cover’s theorem A and B are almost certainly linearly separable.

5 pts	**Q5: Recognizing prime numbers**
Question	A prime number is an integer that is bigger than 1 and has only two divisors: 1 and itself, e.g, 2, 3, 5, 7, 11, 13, 17, … are prime numbers. Any integer between 1 and 2^100 can be represented by a vector of length 100 that contains only 0’s and 1’s. Is it (theoretically) possible to construct a network with 100 input nodes, one hidden layer (possibly with many nodes) and one output node that perfectly recognizes prime numbers? (We assume here that input vectors always consist only from 0’s and 1’s.)
Answer options	**a) Yes** b) No c) I don’t know It is sufficient to “generalize” the XOR network: for any input vector that represents a prime number p we create a hidden node with connections that have weights set to -100 if the connection goes from a node which has value 0 and 1 if it goes from a node which has value 1. Additionally, the bias connection to this hidden node is set to -1*(number of 1’s that occur in the binary representation of p) + 0.5. Consequently, such a node will become active if and only if the input vector represents p. Therefore the output node that computes the generalized OR function completes the construction of our network.

5 pts	**Q6: Overfitting and Regularization**
Question	One way of fighting overfitting is L^2-regularization: punishing the network for having too big (squared) weights. The regularization parameter λ controls the weight of this “punishment”. How do we find the optimal value of λ?
Answer options	a) We train the network on the training set and the algorithm automatically finds the optimal value of λ i.e,, λ is a yet another network parameter that is optimized by the training algorithm. b) We train the network on the training set, using several values of λ, and then select the one that minimizes the error on the training set. **c) We train the network on the training set, using several values of λ, and then select the one that minimizes the error on the test set.**

7 pts	**Q7: Gradient descent**
Question	Let us suppose that to find a minimum of the function $f(x) = x^2$ the gradient descent algorithm, with the learning rate set to 3/4, was run for a number of iterations, starting at $x_0 = 1$ and stopping as soon as an x was found such that $f(x) < 10^{-6}$. After how many iterations the algorithm stopped? Select the answer which is closest to your estimate:
Answer options	d) 5 **e) 10** f) 25 g) 50 h) 100 $(x^2)'=2x$, so $x_{n+1} = x_n - 3/4*2*x_n = -0.5*x_n = (-0.5)^n*x_0$. Thus, for n=10 we have $x_n=1/1024$, so $f(x_n)=1/(1024*1024) < 10^{-6}$.

2 pts	**Q8: Binary classification problems**
Question	Which setup (activation function of the output layer and the loss function) is recommended for solving binary classification problems with MLP?
Answer options	a) Linear and Sum of Squared Errors b) Linear and Cross-entropy c) Sigmoid and Sum of Squared Errors **d) Sigmoid and Cross-entropy** e) Softmax and Sum of Squared Errors

2 pts	**Q9: Multi-class classification problems**
Question	Which setup (activation function of the output layer and the loss function) is recommended for solving multiclass classification problems with MLP?
Answer options	a) Sigmoid and Sum of Squared Errors b) Sigmoid and Cross-entropy c) Softmax and Sum of Squared Errors **d) Softmax and Cross-entropy** e) Linear and Cross-entropy

2 pts	**Q10: Regression problems**
Question	Which setup (activation function of the output layer and the loss function) is recommended for solving regression problems with MLP?
Answer options	a) Sigmoid and Sum of Squared Errors **b) Linear and Sum of Squared Errors** c) Softmax and Sum of Squared Errors d) Softmax and Cross-entropy e) Linear and Cross-entropy

15=3x5 pts	**Q11: Three variants of the Gradient Descent Algorithm**
Question	Three versions of the gradient descent algorithm: the “plain” gradient descent, gradient descent with momentum, and gradient descent with Nesterov momentum, were used to find a minimum of a function of two variables f(x,y). All algorithms, after several iterations, reached the same point (-1, 2) (i.e., x=-1, y=2), and the gradient of f(x,y) at this point was found to be (3, -1). Moreover, the last direction of the steepest descent was v = (2, 3). Finally, let us assume that the learning rate a=0.1 and the momentum term b=0.2. What is the next point (i.e., what are the new values of x and y) after applying the update rules by:
Answer options	1. Gradient descent algorithm: **a) x= -1.3; y= 2.1 [p_{new} = p_{old} - a*gradient]** b) x= -0.7; y= 1.9 c) x= -0.9; y=2.7 d) x= -1.1; y=1.3 e) some information is missing 2. Gradient descent with momentum: a) x= -1.3; y= 2.1 b) x= -0.7; y= 1.9 **c) x= -0.9; y=2.7 [p_{new} = p_{old} - a*gradient + b*last_direction]** d) x= -1.1; y=1.3 e) some information is missing 3. Gradient descent with Nesterov momentum: a) x= -1.3; y= 2.1 b) x= -0.7; y= 1.9 c) x= -0.9; y=2.7 d) x= -1.1; y=1.3 e) **some information is missing** **[we need the gradient at p+b*last_direction]**

3*3 pts	**Q12: LeNet5 (3 questions)**
Question	The first convolutional layer of LeNet5 consists of 6 feature maps, each of size 28x28. Each map is determined by a convolutional filter of size 5x5 which is applied to the input layer of size 32x32, with padding=0 and stride=1. Moreover, each filter uses a bias term.
Answer options	a) How many trainable parameters define this convolutional layer? a. 6*25 **b. 6*26** c. 6*28*28 d. 6*28*28+6 e. None from the above b) How many connections exist between the input and the first convolutional layer? a. 6*25*28 b. 6*26*28 c. 6*25*28*28 **d. 6*26*28*28** e. None from the above

	c) How many connections would exist if the input and the first convolutional layer were fully connected? a. 32*32*6*28*28 b. 32*32*6*28*28+1 c. 32*32*6*28*28+28*28 **d. 32*32*6*28*28+6*28*28** e. None from the above

5 pts	**Q13: Vanilla Recurrent Neural Networks**
Question	Let us consider a simple recurrent network that is used for language modelling. The network uses 5000 input nodes to represent 5000 words (using “one-hot” encoding), 5000 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*5000 b) 3*100*5000 c) 2*100*5000 + 100 **d) 2*100*5000 + 100*100** e) None of the above

3 pts	**Q14: LSTM Networks**
Question	What is the biggest advantage of LSTM networks over Vanilla Recurrent Networks?
Answer options	a) Faster convergence? b) Ability to remember the most recent states? c) Ability to remember the 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

3 pts	**Q15: Restricted Boltzmann Machine**
Question	What function is optimized during training an RBM:
Answer options	a) Mean Squared Error b) Logarithmic Loss c) Contrastive Divergence **d) LogLikelihood** e) Cross Entropy f) None of the above

5 pts	**Q16: 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. There were X users, Y movies, Z possible ratings, and the network was using T hidden nodes. How many trainable parameters had this network? For simplicity, we ignore the biases.
Answer options	a) X*Y*Z*T b) X*Y*Z **c) Y*Z*T (input: Y*Z, hidden: T)** d) X*Z*T e) None of the above

10 pts	**Q17: Batch Normalization**
Question	Batch normalization is a very powerful technique for training deep networks. How many additional trainable parameters does it introduce per layer that uses batch normalization:
Answer options	a) None (each batch is just shifted by the mean and divided by the standard deviation. These are not trainable parameters). **b) 2 [beta (shift) and gamma (scaling factor)]** c) 2*number_of_batches d) 2*dimensionality of data e) 2*number_of_batches*dimensionality of data f) None of the above answers is correct