DIT 821: Software Engineering for AI Systems **(20 points)**

Exam 2021-10-27

Examiner: Ivica Crnkovic

Number of points and grades:
16-20 points – Pass with Distinction (VG).
10-15 points – Pass (G)
Less than 10 points – Fail (U)

Activities at the lectures in form of participation in breakout rooms and presence at the lecture can give 5%, or 1 point.

Only pens and blank papers are allowed during the exam. Please enumerate the pages and write the answers clearly. Unreadable and/or confusing answers will be graded with zero.

Good luck!

Problem 1	Linear regression; learning rate	3 points
Problem 2	Training and validation	3 points
Problem 3	Confusion matrix	4 points
Problem 4	Deep learning	4 points
Problem 5	ML system development workflow	3 points
Problem 6	General questions	3 points

Problem 1 - Linear regression; learning rate (3 points)

Gradient descent is a method to calculate the weight factors that will give a minimum of the cost function. The method calculates new values of the cost function by using a new value of weight factors from the previous value, in an iteration process, as expressed by the formula below.

$$w_j \leftarrow w_j - \alpha \frac{\partial}{\partial w_j} J(\mathbf{w}) \tag{0.1}$$

In the formula there is a number *learning rate* α that determines the change of the weight factors w_j in each iteration. It is up to the developer to choose a specific value of α.

a) Explain shortly (in a few sentences) how the value of α influences the decent gradient iterations. (0.5 points)

b) One of the issue with gradient descent and value of α is the computation (execution) time. Explain how the computation time depends on α and illustrate it by drawing a graph on Figure 1 a). From the computation time perspective, is it better to choose small small α or large α? (1.0 point)

c) Using a gradient descent may lead that the exact minimum of the cost functions is not found, but a certain approximation. Selection of α may have influence on the cost function value. Try to draw a graph on Figure 1 b) how the cost function might change depending on α selection and explain shortly (a few sentences) your reasoning. (1.0 point)

d) The second factor that influences the precision of the cost function minimum is a number of iterations in the gradient descent loop. The number of iterations can be specified in advance, or by some criteria. Define and explain shortly which criterion can be used to stop the iterations. (0.5 points)

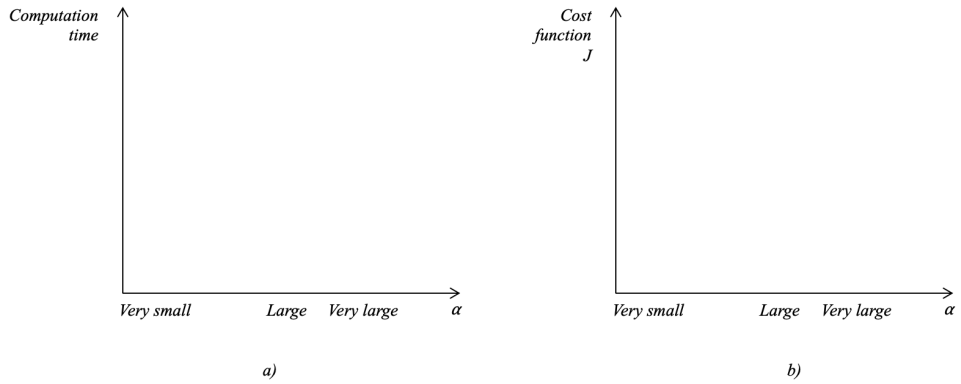

Figure 1: Influence of learning rate α on a) computation time, b) cost function

Solution

a) α defines a step in which the gradient changes. A small value gives a small step and require more iterations to reach the minimum. The larger values lead to larger steps, but can cause that in a step the minimum is jumped over and the calculated cost is larger than the minimum.

b) see the figure a)

c) see the figure b)

d) One criterion can be the value of J - (for example if $J < J_min$) - but this is not good as we do not know the real minimum. The other is the value of the gradient that must be close to zero when we are close to the minimum. A third criterion can be that the difference of the cost function between two iterations is very small, or zero.

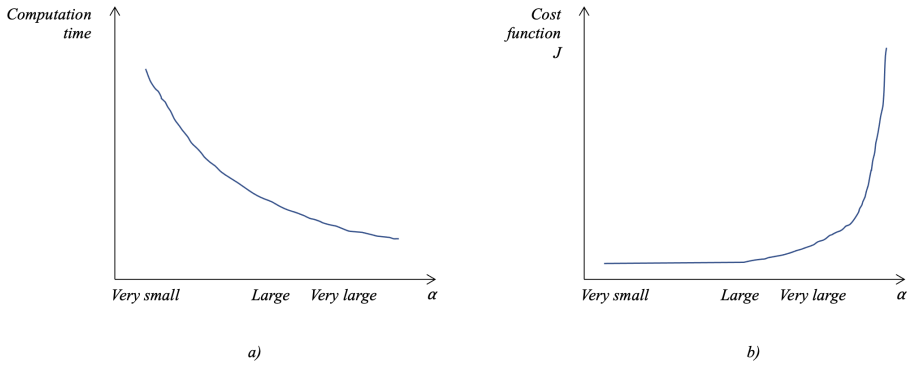

Figure 2: Influence of learning rate α on a) computation time, b) cost function

Problem 2 - Training and validation (4 points)

In a ML training process the goal is to find the best possible weight factors which give the minimal cost function. However, the chosen weight factors may not be so good for new examples. For this reason a minimal cost function of the validation set is of more interest. The cost function in linear regression is defined by a formula shown below. Several parameters in this formula should be setup by the developer by running experiments of training and validation datasets. (note: The w factors are calculated in the training, and used in validation. Also, a training dataset might be changed, while the validation set will be kept the same).

$$J(\mathbf{w}) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_w(\mathbf{x}^{(i)}) - y^{(i)})^2 + \lambda\sum_{j=1}^{n} w_j^2\right] \tag{0.2}$$

Your assignment: In in Figure 2 draw the graphs of the cost function and shortly describe the behavior of the cost function when changing:

a) size of training dataset m (1 point);

b) regularization parameter λ (1 point);

c) number of features and polynomial number (that corresponds to the dimension of weight factors vector $\mathbf{w}$ (1 point).

d) Describe shortly and explain why the order in which you would you run the experiments (1 point).

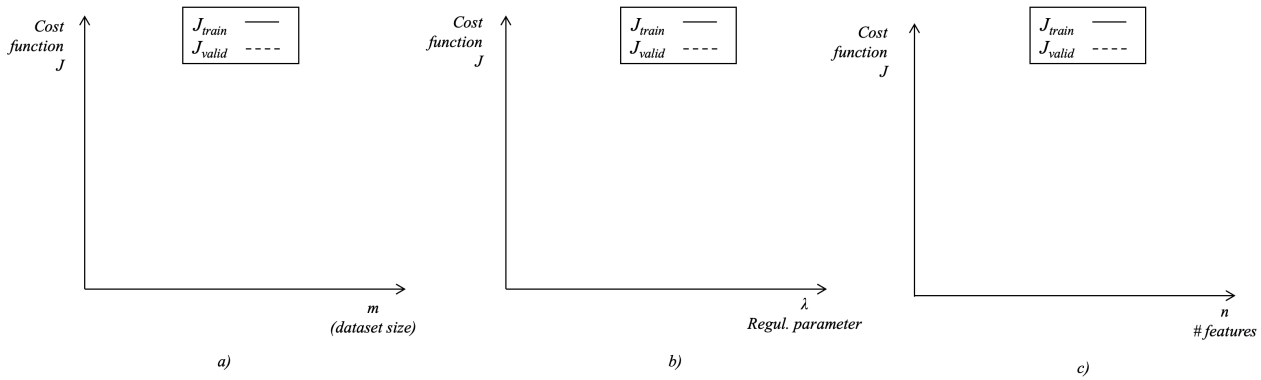

Figure 3: Cost functions for variations of a) m dataset size, b) λ regularization parameter, and c) n - number of features

Solution

a) Cost function in relation to size of dataset m. Training dataset: With a very small number of examples in the dataset will give a small value of J. By adding number of examples the value of J will increase, faster in the beginning and slower when the dataset is larger. Finally, the increase of J will be very small.

Validation dataset: For a small training dataset the cost function J will be large - the prediction function will have a high bias. By increasing the training dataset the cost function will decrease - at the end (when training dataset is large) will be close to the cost function value of the training dataset.

b) Regularization parameter λ. Training set: If $\lambda = 0$ there is no regularization parameter, and the cost function has a minimum value for given w and training dataset m. With increase of λ the cost function increases, and by a large λ, the regularisation parameter becomes dominant.

Validation set: If the cost function in the training set is low due to a large number of features or polynomial, the prediction function will be over-fitted. That will make the cost function high for the validation dataset. By increase of λ the overfitting will decrease and J will decrease. By further increase of λ the regularization parameter becomes dominant and the cost will increase.

c) Number of features and polynomial number. Training dataset: With a low number of features/polynomial number the cost function will be high. By increasing a number of features/polynomial number the cost function will decrease. With a very high number of features, the prediction function fits better to the data, J decreases, and goes towards zero.

Validation set: With a small number of features J will be as high or higher than J for the training set. By increase of feature numbers J decreases, but with a higher number the over-fitting effect becomes dominant and J increases.

The results a), b), and c) are shown on Figure 3 below.

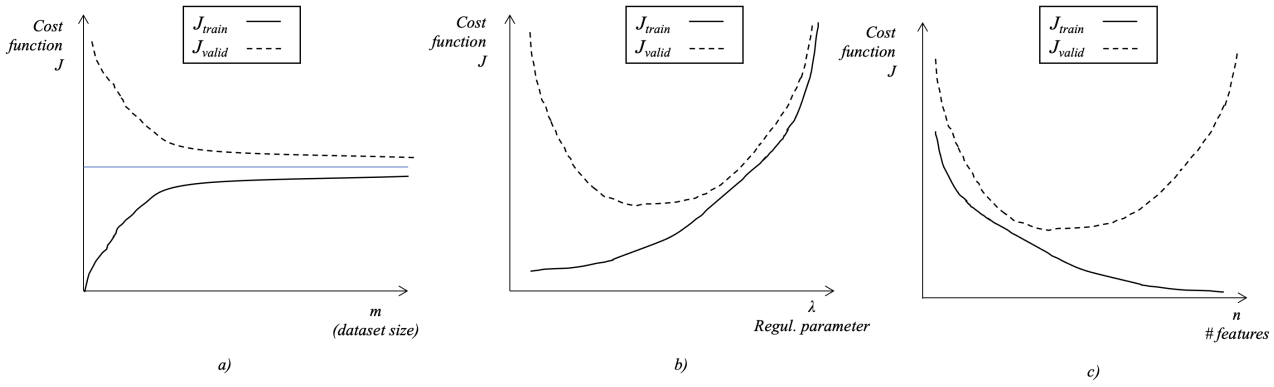

Figure 4: Cost functions for variations of a) m dataset size, b) λ regularization parameter, and c) n - number of features

Problem 3 - Confusion matrix (4 points)

Suppose the following: A new COVID-test system was experimently tested on known COVID-positive and known COVID-negative patients. The test system gave the following results: 100 COVID-positive patients were tested and the system gave the result 90 positive and 10 negative. The same system tested 100 covid-negative patients and provided the following results: 80 negative and 20 positive.

In an operation by testing 1000 people, the system found 50 COVID-positive and 950 COVID-negative cases. Based on these data do the following:

a) Write the confusion matrix, and specify TP, TN, FP, FN for the experiment with 100 COVID-positive and 100 COVID-negative cases. (1 point)

b) Specify the following formulas and calculate: the system accuracy (total number of correct predictions divided by the total number of a dataset), precision (the number of correct positive predictions divided

by the total number of predicted positive, and recall (the number of positive divided by all real positive examples in the dataset). (1 point)

c) Based on the quality metrics from the experiment, and from the results found in the operation, calculate the new confusion matrix, and from it the expectation of a real number of tested COVID-positive people, and COVID-negative people. (Round the numbers to integer values.) (2 points)

Solution

a) *Write the confusion matrix, and specify TP, TN, FP, FN.*

Confusion matrix	Tested COVID-positive	Tested COVID-negative
Actual COVID-positive	TP = 90	FN = 10
Actual COVID-negative	FP = 20	TN = 80

b) *Specify the formulas and calculate: the system accuracy (total number of correct predictions divided by the total number of a dataset), precision (the number of correct positive predictions divided by the total number of predicted positive, and recall (the number of correct positive predictions divided by all real positive examples in the dataset).*

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{90 + 80}{200} = \frac{170}{200} = 0.85$$

$$Precision = \frac{TP}{TP + FP} = \frac{90}{90 + 20} = \frac{90}{110} = 0.81818$$

$$Recall = \frac{TP}{TP + FN} = \frac{90}{90 + 10} = \frac{90}{100} = 0.90$$

c) *Based on the quality metrics from the experiment, and from the results found in the operation calculate the new confusion matrix, and from it the expectation of a real number of COVID-positive people, and COVID-negative people.*

For the operation the system gave the result that is a sum of correctly found (TP) and wrongly found: $TP + FP = 50$

which also implies: $TN + FN = 950$.

From Precision $\frac{TP}{TP+FP} = 0.81818$ we get: $TP = 0.81818(TP + FP) = 0.81818 * 50 = 41$.

This gives: $TP = 41$, and $FP = 50 - TP = 50 - 41 = 9$

We have to find TN and FN.

from Recall $\frac{TP}{TP+FN} = \frac{90}{100} = \frac{9}{10}$ we have:

$TP = \frac{9}{10}(TP + FN)$ or $10TP = 9TP + 9FN$ or $TP = 9FN$ or

$FN = TP/9 = 41/9 \approx 4,6 \approx 5$

We know: $TN + FN = 950$, so $TN = 950 - FN = 950 - 5 = 945$.

Now we have all elements of the confusion matrix:

Confusion matrix	Tested COVID-positive	Tested COVID-negative
Actual COVID-positive	TP = 41	FN = 5
Actual COVID-negative	FP = 9	TN = 945

This will give the final answer:

Expected number of COVID-positive is $TP + FN = 41 + 5 = 46$.

Expected number of COVID-negative is $FP + TN = 9 + 945 = 954$.

Problem 4 - Deep learning (4 points)

Q1: Figure 5 CNN shows the Convolution Neural Network pipeline. Explain briefly (in a few sentences) each part of the pipeline (1 point)

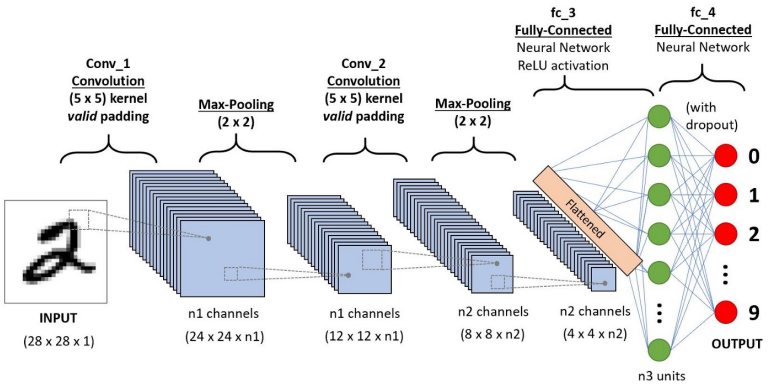

Figure 5: CNN model pipeline

Q2: Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture. The LSTM cell is shown on Figure 6. Describe shortly the principle of RNN and explain the inputs, outputs and flows shown on the figure. (1 point)

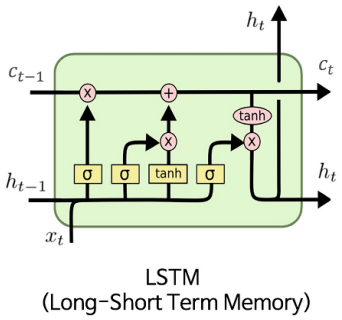

Figure 6: Long-short term memory

Q3: Figure 7 shows a neural network. For calculation of the hypothesis $h(x)$ the following equations are used:

$$a^{(1)} = x; \;\; a^{(2)} = sig(\theta^{(1)} a^{(1)}); \;\; a^{(3)} = sig(\theta^{(2)} a^{(2)}); \;\; h_\theta(x) = a^{(3)} \tag{0.3}$$

The equations are expressed in vector and matrix form.

a) Specify the form (shape) of each variable ($a^{(1)}, \theta^{(1)}, a^{(2)}, \theta^{(2)}, a^{(3)}$). (0.5 points)

b) For given numbers at the figure specify $\theta^{(1)}$ and $\theta^{(2)}$. (0.5 points)

c) Suppose you have a case $x_1 = 0; \; x_2 = 0$. Provide approximate values of $h^{(1)}(x)$ and $h^{(2)}(x)$. Show how you have calculated the result. For the values of sigmoid ($sig(t)$) use the graph on Figure 7 (Note: some of the weight factors are negative). (1 point)

Solution
Q1: CNN pipeline:

- Convolutional Operation: apply different kernels for extracting features maps form the image
- ReLu layer: for adding non-linearity
- Pooling to compress the features map by extracting the very important features

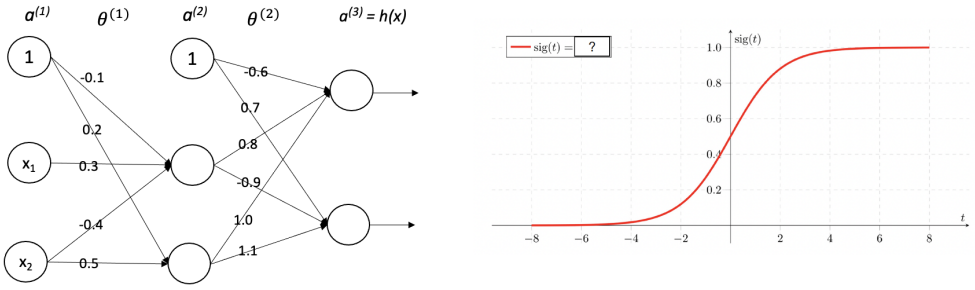

Figure 7: Neural network

- Flattening: preparing the pooled feature map to be injected into the ANN model
- ANN model: the model used for performing the tasks of classification and regression.

Q2: LSTM

we need the LSTM as solution for co-dependency problem. Example: The tree color is "green" here in this example the distance between the word ("green") that the model wants to predict and the necessary information (tree and color) is short. BUT what if we have a longer statement like (I live in Goteborg in Western Sweden..... where I live, the weather is generally "cold" most of the year) here THE GAP BETWEEN THE PREDICTION "COLD" AND THE NECESSARY CONTEXT INFORMATION "Sweden" IS LARGE. that's why we need the LSTM to solve this problem

LSTM contains gates that can allow or block information from passing by. Gates consist of a sigmoid neural net layer along with a pointwise multiplication operation. Sigmoid output ranges from 0 to 1: 0 = Don't allow any data to flow 1 = Allow everything to flow

$$f_t = \sigma(W_f{*}[h_t{-}1, X_t]{+}b_f), i_t = \sigma(W_i{*}[h_t{-}1, X_t]{+}b_i), C_t emp = tanh(W_c{*}[h_t{-}1, X_t]{+}b_c), C_t = f_t{*}C_t{-}1{+}i_t{*}C_t emp, o_t = \sigma(V \tag{0.4}$$

Q3: NN

a) Specify the form (shape) of each variable
$a^{(1)}$ - vector dimension 3; $\theta^{(1)}$ - matrix dimesion 3x2,
$a^{(2)}$ - vector dimension 3; $\theta^{(2)}$ - matrix dimes ion 3x2
$a^{(3)}$ - vector dimension 2

b) For given numbers at the figure specify $\theta^{(1)}$ and $\theta^{(2)}$. (0.5 points)

$$\theta^{(1)} = \begin{bmatrix} -0.1 & 0.2 \\ 0.3 & 0 \\ -0.4 & 0.5 \end{bmatrix}; \quad \theta^{(2)} = \begin{bmatrix} -0.6 & 0.7 \\ 0.8 & -0.9 \\ 1.0 & 1.1 \end{bmatrix}$$

c) Suppose you have a case $x_1 = 0;\ x_2 = 0$. Provide approximate values of $h^{(1)}(x)$ and $h^{(2)}(x)$. Show how you have calculated the result. For the values of sigmoid ($sig(t)$) use the graph on Figure 7 (Note: some of the weight factors are negative). (1 point)

$$a_0^{(1)} = 1;\ \ a_1^{(1)} = x_1 = 0;\ \ a_2^{(1)} = x_2 = 0$$
$$a_0^{(2)} = 1;\ \ a_1^{(2)} = sig(-0.1) = 0.4;\ \ sig(a_2^{(2)}) = sig(0.2) = 0.55;$$
$$a_1^{(3)} = sig(-0.6 * a_0^{(2)} + 0.8 * a_1^{(2)} + 1.0 * a_2^{(2)}) = sig(-0.6 * 1 + 0.8 * 0.4 + 1.0 * 0.55) = sig(0.27) = 0.55;$$
$$a_2^{(3)} = sig(0.7 * a_0^{(2)} + (-0.9) * a_1^{(2)} + 1.1 * a_2^{(2)}) = sig(0.7 * 1 + (-0.9) * 0.4 + 1.1 * 0.55) = sig(0.95) = 0.6$$
$$h_1(x) = a_1^{(3)} = 0.55;\ \ h_2(x) = a_2^{(3)} = 0.6$$

Problem 5 - ML system development workflow (3 points)

A company in Gothenburg recently decided to develop an object detection system to help with automation processes within their factory. They have assembled a new team of software engineers, data scientists and ML engineers to develop ML models and integrate it into their existing software systems.

a) List and briefly describe the machine learning workflow stages required for such project. (1 point)

b) Since the idea of ML-based system is new in the company, the management wants the engineers to be aware of how it differs from traditional software systems. Your task is to describe the differences between traditional software and ML-based software requirements. (0.5 points)

c) What is feature engineering? Mention and describe when to use any of the four feature engineering techniques of your choice. (0.5 points)

d) When annotating and labelling the image data, the stake-holders wants to determine the inter-annotator agreement. State and describe how you will measure it? (0.5 point)

e) Give two examples of machine learning model deployment patterns to consider. Discuss their pros and cons. (0.5 points)

Solution

a) The machine learning workflow includes the following stages: model requirements, data collection, data cleaning, data labelling, feature engineering, model training, model evaluation, model deployment, and model monitoring.

b) Differences between ML-based software and traditional software requirements.

- ML-based systems are data-driven and closely coupled with data of the particular application context.
- Quantitative measures such as accuracy metrics comprise a majority of the requirements, while traditional software often has its requirement as specifications, e.g., specification for interfaces.
- ML-based systems often involve a large number of preliminary experiments.

c) Feature engineering is the process of transforming raw input data to the most informative features for machine learning algorithms. Feature engineering techniques for numeric features include, normalization, standardization, log transformation, binarization and binning. Feature engineering techniques for categorical data include one-hot encoding and hash encoding. Feature engineering techniques for textual data include bag-of-words (BoW), and bag-of-n-grams. Principal component analysis (PCA) is an example of technique for dimensionality reduction.

d) Inter-annotator agreement (IAA) measures how well two or more annotators can make the same annotation decision for a specific category of object. IAA can be measured using Intersection over Union (IoU). IoU calculates the overlapping areas of intersections between two bounding boxes, divided by the total area of both bounding boxes.

e) Common deployment patterns mentioned in the course are i) Expose model via REST API, and ii) Embedded model.

Problem 6 - General Questions (3 points)

Answer to each question with a few sentences. Be precise and concise.

a) What is a difference between logistic regression and Linear Regression? (0.5 points)

b) What is *feature scaling* and how can it be performed? (0.5 points)

c) In the context of classifiers, what is One-vs-All? (0.5 points)

d) What are the problems of decision trees and how to address them? (0.5 points)

e) Describe shortly one of the techniques to choose the number of clusters in the K-means algorithm (0.5 points)

f) Explain one approach used to perform data labelling for ML system? (0.5 points)