UNIVERSITY
OF SKÖVDE

School of Humanities and Informatics

WRITTEN EXAMINATION

Course Big Data Programmering

Sub-course

Course code IT 739A

Credits for written examination 5,5

Date 2024-05-29

Examination time 08:15-12:00

Examination responsible Richard Senington

Teachers concerned Gunnar Mathiason, Richard Senington

Aid at the exam/appendices No

Other

Instructions

- ☐ Take a new sheet of paper for each teacher.
- ☒ Take a new sheet of paper when starting a new question.
- ☒ Write only on one side of the paper.
- ☒ Write your name and personal ID No. on all pages you hand in.
- ☒ Use page numbering.
- ☒ Don´t use a red pen.
- ☒ Mark answered questions with a cross on the cover sheet.

Grade points A-F

Examination results should be made public within 18 working days

Good luck!

Total number of pages 3

IT739A, Exam, May 29th, 2024, at 8:15-12:30

Instructions that must be followed:

- The max number of points listed with each question indicates how thoroughly you are expected to answer the question.
- Number your answer clearly with the same numbering as the question and its sub questions.
- Be clear in our writing. Sometimes fewer but more concise formulations are better than writing a lot of text.

The grading of your answers considers:

- The **correctness** of your answer/explanation
- The **clarity and logical cohesion** of your answer/explanation
- The **conciseness** of your answer/explanation

Grading levels:

- There are 5 questions (with sub questions), which give maximum 10 points each. 1) You need to pass 5 points per each of these 5 questions to pass the exam (giving 25 points). 2) Accumulated points above that threshold linearly determines the exam grading (exam maximum is 50 points).
- Exam grading range is A to F, which also determines the final course grading: A: 45-50 points, B: 40-44 points, C:35-39 points, D: 30-34 points, E: 25-29 points

Examination goals:

This exam assesses the following Course Curriculum goals

- *critically reflect on the needs for programming tools for Big Data,*
- *prove understanding of the development of tools for Big Data programming,*
- *show an in-depth understanding of paradigms that are frequently used for Big Data programming,*

For information: The course Assignment assesses the following Course Curriculum goals

- *demonstrate abilities to make use of chosen programming tools for Big Data programming and data analysis and*
- *execute a small data analysis project using Big Data programming.*

Exam questions:

1. Keras and Deep Learning (10 points):

1. Deep Learning: What is the role of an *optimizer* when propagating updates to weights in a neural network? (1p)

2. Why do you use "*dropout*" when training a neural network? How does it work to achieve this? (1p)
3. Explain the term "stride" that is used for the Convolutional operation? What is the result of setting a low or high value for the stride. (2+1p)
4. Explain why you use a pooling layer in a Convolutional network? (1p)
5. Explain why it is critical to choose a suitable loss function when selecting hyperparameters for a neural network design, and how does that choice influence the learning capabilities of that network? (4p)

2. Tensorflow and application (10 points):

1. Explain the concept of a *tensor data structure*. Also, what is the difference between *rank* and *dimensionality* for data that is formulated as a tensor? (3p)
2. Explain the main feature that makes a TPU faster than a GPU, and what mathematic operation that is typical in machine learning that will be much faster because of it? (4p)
3. The *AllReduce* operation is used in Tensorflow as a key function to propagate model updates (gradients) of Tensorflow's distributed model learning. Describe why AllReduce is needed in distributed machine learning and what is one disadvantage of it? (2+1p)

3. Programming and deployment for Big Data (10 points):

1. When evaluating the performance of a trained model, you need to reserve some hold-out data to use as a *test set*? Why do you need to do that? Also, explain what is meant with *leakage* of information when you try to improve a model's accuracy by tweaking the data or the data preprocessing. (2+2p)
2. Describe a way to *detect overfitting* using plotting of the *accuracy* and *loss* parameters during neural network training. (3p)
3. Some data are usually translated using *one-hot encoding* before exposed to the model input during training. Explain what type of data should be translated this way, and why is that needed for such type of data? What could be one side effect of not doing this translation? (3p)

4. Architecture, databases and the issues of big data (10 points):

1. Value is one of the 5Vs of big data. Discuss Value and compare it to the other 4 Vs. Please include in your answer how each of the other 4Vs can support Value or make it difficult. (4p)
2. With respect to BigData, compare and contrast the three major types of database; Traditional SQL, Graph and Document. (3p)

3. Discuss the similarities and differences between virtual machines and containers, and the advantages and disadvantages when using these for big data. (3p)

5. Functional programming and Spark (10 points):

1. Explain the differences and similarities between Scala and Spark. (1p)
2. Explain the difference between using Spark through Python and using Spark through Scala. (1p)
3. Explain how map-reduce works and why it can improve execution time on large datasets. (3p)
4. In this question, please write a spark/scala program or function to perform the following task. Please note that exact code will not be required but you do need to express the concepts in an appropriate way for the Spark environment.

   You are given 2 csv files with columns a, b, c. Column a is a float, column b is a float, column c is a unique integer.
   You have access to a database with the columns d, e, f. Column d is a float, column e is a float, column f is a unique integer.

   Please load the 2 csv files in spark and combine them into 1.
   Then join the dataset you have created with the database where column c should be equal to column f.
   You should now have a dataset of 5 columns, a, b, d, e, c. Column c is the unique id.
   For each row of this dataset compute the average of columns a, b, d, e.
   Please find the highest average and return the unique id.
   Print the unique id.

   (5p)