UNIVERSITY
OF SKÖVDE

School of Information Technology

WRITTEN EXAMINATION

Course: Big Data Programming

Examination: Final written exam

Course code: IT739A

Credits for written examination: 5.5hp

Date: May 9, 2025

Examination time: 14:15 – 18:30

Examination responsible: Richard Senington

Teachers concerned: Juhee Bae, Richard Senington, Gunnar Mathiason

Aid at the exam/appendices

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

Examination results should be made public within 18 working days

Good luck!

Total number of pages

UNIVERSITY
OF SKÖVDE

IT739A, Re-Exam, May 9th, 2025, at 14:15-18:30

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 your 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 a maximum 10 points each.
1) You need to pass 5 points per each of the 5 questions to pass the exam (25 points).
2) Accumulated points above that threshold linearly determines the exam grading (the 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 need 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.*

The re-exam questions begin on the next page.

UNIVERSITY
OF SKÖVDE

Re-Exam questions:

1. Keras and Deep Learning (10 points):

1. Describe how the Loss function and the Optimizer interact in a neural network in a typical supervised learning setup. (4p)
2. Why do you use "dropout" when training a neural network? How does it work to achieve this? (1p)
3. Convolutional networks:
    a. Motivate why convolutional networks typically use fewer trainable weights compare to fully connected network. Also, under which assumptions about your data can you reach the benefit of using convolutional layers (tip: this has to do with re-occurring patterns)? (3p)
    b. Explain and differentiate between the terms feature map and response map that are used to describe the data transformation in a convolutional operation. (2p)

2. Deep Learning and application (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. 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)
3. 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? (3p)

3. Large Language Models (LLM) (10 points):

1. Prompt engineering:
    a) Define "prompt engineering" and explain why it is critical for maximizing the performance of large language models. (1p)
    b) Provide two distinct prompt engineering techniques, including for each: a short description, an example, and a practical use case. (3p)
2. List and describe three evaluation metrics used to assess large language models. For each metric, specify: (a) what is measured, (b) the name of the metric, (c) its key characteristics, and (d) the types of tasks it is typically used for. (3p)

UNIVERSITY
OF SKÖVDE

3. Identify two major challenges when deploying large language models in real-world applications. For each challenge, explain the risk involved and propose at least one mitigation strategy. (3p)

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

1. Describe each of the 5Vs of Big data. (2p)
2. One of the 5Vs of Big Data is Value. Please describe the Value aspect of big data in more detail and how it is connected to each of the other Vs. (3p)
3. Explain how virtualisation and/or containers are used to support Big Data analytics. (3p)
4. Explain the limitations of SQL type databases for Big Data analytics. (2p)

5. Functional programming and Spark (10 points):

1. Describe how a Spark cluster works. You can either focus exclusively on the types of process, or use different machines to help explain what is happening. (3p)
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. (2p)
4. Below are 2 sections of code. One is in PySpark the other in Scala/Spark. They do the same thing, but are provided so you can pick the language you prefer. They both refer to a csv file with the following 3 columns; name, country, region, population. What does the following code do? Please include the overall purpose of the code. Please include where, how and to what extend parallelism is achieved. (4p)

```
<Python>
rdd1 = sc.textFile("city_data.csv").map(lambda x:x.split(","))
.filter(lambda x: x[2]=="Americas")
.groupBy(lambda x: x[1])
.map(lambda x: (x[0],x[1].map(lambda y:y[3]).max()   ))
rdd1.foreach(print)

<Scala>
val rdd1 = sc.textFile("city_data.csv").map(x=>x.split(","))
.filter(x=> x(2)=="Americas")
.groupBy(x=> x(1))
.map((a,b)=> (a,b.map(y=>y[3]).max()   ))
rdd1.foreach(println)
```