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: March 25, 2025	Examination time: 8:15 – 12:30

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, Exam, Mar 25th, 2025, 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 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 exam questions begin on the next page.

UNIVERSITY
OF SKÖVDE

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. What is the key to enable stacked layers of neural networks to succeed, to avoid the vanishing gradients problem? Explain for one activation function how this is met. (4p)
3. Convolutional neural networks (CNN): What is the underlying assumption about the data that makes CNNs effective in compressing the data in a significantly smaller representation? (2p)

2. Deep Learning and application (10 points):

1. Deep learning model design:
   a) Give the draft model outline that is meant for a Keras-based implementation, of a deep network based on CNN models of two types of inputs, a 2-rank tensor and a 3-rank tensor, for each typical training example. You don't need the code for this example, but you should give key hyperparameters for such network setup. Hint: The deep network lecture example has a similar mix of its feature data. (4p)
2. Model training:
   a) Describe a way to *detect overfitting* using plotting of the *accuracy* and *loss* parameters during neural network training. (2p)
   b) Explain the difference between structured tabular data and structured graph data. (2p)
3. Tensorflow usage:
   a) Briefly explain two separate reasons how the Tensorflow Data API helps in building better data preparation pipelines than without this API. (2p)

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

1. Transformers:
   a) Describe the essential components of a Transformer and briefly explain the function of each component within it (how each component works). (2p)
   b) What are the roles of these components in the Transformer? (2p)
   c) How does a Transformer differ from neural network models like RNNs and CNNs? (2p)

2. LLM enhancement techniques, RAG (Retrieval-Augmented Generation) and RLHF (Reinforcement Learning from Human Feedback):
   a) Define RAG and RLHF and explain their core mechanisms. (2p)
   b) What are the benefits and limitations of RAG and RLHF, respectively? (2p)

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

1. One of the 5Vs of Big Data is Volume.
   a) Describe Volume with respect to the 5Vs of Big Data. (2p)
   b) Describe why Volume is a concern of Big Data and the known software tools/approaches for overcoming this concern. (2p)
   c) How does the Volume impact the Scalability of processing in big data? (2p)
2. Volume alone is not enough to make Big Data useful. Please discuss what must be done to help firms gain value from their volume. (3p)
3. Graph databases are one form of database that can be used in data storage and analytics. Please briefly describe graph databases and their connection to the Volume problem and why people might use them. (1p)

5. Functional programming and Spark (10 points):

1. Describe how a Spark cluster works. You can either focus exclusively on the types of processes or use different machines to help explain what is happening. (3p)
2. Compare a database's query engine (you can choose the type, but please specify) with Spark for data analytics. (3p)
3. Below are two sections of code. One is in PySpark; the other is in Scala/Spark. They do the same thing but are provided so you can pick the language you prefer. What does the following code do? How much do they parallelize? (4p)

```
<Python>
rdd1 = sc.textFile("untitled.txt").map(lambda x:x.split(" "))
.filter(lambda x:int(x[3])>93)
.sortBy(lambda x:float(x[1]))
m1 = rdd1.map(lambda x:float(x[1])).sum()/rdd1.count()
print(m1)

<Scala>
val rdd1 = sc.textFile("untitled.txt").map(x=>x.split(" "))
.filter(x=>x(3).toInt>93)
.sortBy(x=>x(1).toFloat)
val m1 = rdd1.map(x=>x(1.toFloat).sum()/rdd1.count()
print(m1)
```