Univ.-Prof. Dr.-Ing. Matthias Boehm
Technische Universität Berlin
Faculty IV - Electrical Engineering and Computer Science
Berlin Institute for the Foundations of Learning and Data (BIFOLD)
Big Data Engineering (DAMS Lab) Group

July 31, 2025

Exam Architecture of Machine Learning Systems (SoSe 2025)

Important notes: The working time is **90min**, and lecture materials or any kind of mobile devices are not allowed. Please, make sure to put your name and matriculation number on the top right of the first page of the task description, and each additional piece of your own paper. You may give the answers in English or German, written directly into the task description.

Task 1 Parameter Servers (16 points)

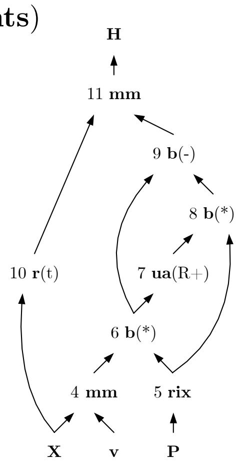
(a) Describe the overall system architecture of *data-parallel parameter servers*, explain its components and interaction among these components. (**10 points**)

(b) Describe synchronous (BSP) and asynchronous (ASP) *update strategies* in data-parallel parameter servers and name their advantages and disadvantages. (**6 points**)

	Synchronous Updates	**Asynchronous Updates**
Description		
Advantages **Disadvantages**		

Task 2 Compilation Techniques (21 points)

(a) Given the directed acyclic graph (DAG) below, perform *shape inference* and determine the dimensions (number of rows and columns) of the intermediates produced by operations (4) through (11). The input matrices have the following dimensions: **X** (30,000 × 700), **v** (700 × 3), and **P** (30,000 × 4). (**4 points**)



mm .. **matrix multiplication**

rix .. **right indexing** (1-based indexing, lower/upper bounds inclusive)

r(t) .. **transpose**

b(*) .. **binary elementwise multiplication**

b(-) .. **binary elementwise subtraction**

ua(R+) .. **unary aggregation, row-wise summation**

Original Expression:

```
Q = P[, 1:3] * (X @ v);
H = t(X) @
    (Q - P[, 1:3] * rowSums(Q));
```

(b) Sketch an algorithm for *common subexpression elimination (CSE)* by example of the following script and show both, the original tree and resulting DAG of operators. In the example, `@` refers to matrix multiplication and `S2[i,]` to matrix indexing. (**6 points**)

```
R1 = diag(S1) @ A + S2[i,] @ A
R2 = diag(S1) @ A + S2[i,] @ b
```

(c) Given the expression `sum(A*7+B)` where A and B are large matrices, simplify this expression through rewrites, and explain why the simplified expression likely improves runtime performance. (**4 points**)

(d) Assume an example chain of matrix multiplications (**A B C D E F**), describe the problem of *matrix multiplication chain optimization*, and a dynamic programming algorithm for solving it efficiently. (**7 points**)

Task 3 Data-parallel Execution (10 points)

(a) Given the distributed dataframe D of three partitions below, describe the data-parallel (MapReduce-like) computation of $Q : \gamma_{A,\min(B),\max(B)}(D)$ (group-by A, return min(B) and max(B)) including shuffling and the *example intermediates and results*. (**7 points**)

A	**B**

X	3
X	4
X	1
Y	7

X	2
Y	3
X	1
X	2

Y	5
X	3
Z	7
X	4

(b) Describe the structure of a *block-partitioned matrix* as one of the widely-used distributed matrix representations. (**3 points**)

Task 4 LLM Training and Inference (6 points)

(a) Explain the process and purpose of *tokenization* in large language models. (**3 points**)

(b) Describe the concept of *Key-Value (KV) caching* in LLM inference. (**3 points**)

Task 5 Data Access Optimizations (13 points)

(a) Assume an n-by-m matrix $\mathbf{X}$ with sparsity $\frac{\text{nnz}(\mathbf{X})}{n \cdot m}$ (fraction of number of non-zeros to cells). In the table below, indicate via a ✓ which matrix block representation is the *most space-efficient* one for each of the five different shape/sparsity scenarios (assuming 4 Byte integer and floating point data types for indexes and values). (**5 points**)

Shape, Sparsity	Dense	Compressed Sparse Rows (CSR)	Coordinate (COO)
1,000 × 1,000, 0.7			
1,000 × 1,000, 0.5			
1,000 × 1,000, 0.1			
20,000 × 50, 0.01			
200 × 5,000, 0.001			

(b) Describe min-max quantization of an `FP64` (64bit floating point) representation into `UINT8` (8bit integer). Why does such an encoding increase runtime performance? (**8 points**)

Task 6 Data Preparation (15 points)

(a) Given the input data below, apply *feature hashing and one-hot encoding* (with string-length as the hash function and k=2) to the categorical columns A and C, and *binning and one-hot encoding* with three equi-width bins to the numerical column B. (**10 points**)

A	**B**	**C**
Low	0	S
High	3.1	M
Med	7	L
Low	9	XL
Low	15	M
Low	7	M
Med	4.2	L
High	12	XL
High	13	L

(b) Given the following three sentences of single-character tokens, show their encoding in a *bag-of-words* representation? (**5 points**)

- A C A B B G D E F.
- A B A A G D.
- C A A B B B C.

Task 7 Model Selection (8 points)

(a) Describe the task of *hyper-parameter tuning* by example of *Bayesian Optimization* as a directed search strategy, and how it balances exploitation and exploration (**8 points**)

- **Hyper-parameter Tuning:**

- **Bayesian Optimization:**

Task 8 Model Debugging (5 points)

(a) Explain the concept of a *confusion matrix* and show the concrete confusion matrix for the following example. (**5 points**)

- Real labels $\mathbf{y} = \{1,1,1,4,2,3,1,2,3,3,2,2,1,4,4\}$.
- Predictions $\hat{\mathbf{y}} = \{1,1,4,4,2,3,1,3,2,3,2,2,1,4,1\}$.

Task 9 Model Deployment (6 points)

(a) Consider a deployed model M in a cloud serving environment and assume 1,000s of concurrent client requests. Name three strategies for improving model scoring throughput at the serving site, and the reason for the improvement. (**6 points**)