Dear friends, here is a sample list of projects. You are free either to choose a project from the list or (better) to create your own one of similar style.

A project should consists of some computer experiments and theoretical study. Finally, each project team gives a talk on our seminar. The day of the talk should be chosen one month before the talk. A preliminary version of slides for the talk should be sent to the staff one week before the talk.

Typically, a project team consists of single person. Still, teams of 2 or 3 persons are also allowed. In this case, the work should be divided between the team members in such a way that we all could evaluate the contributions of the members.

1. Singular decomposition for face recognition

The lecture will discuss the use of SVD for image compression.

The project proposes to write a program that allows for the person recognition based the photo using singular value decomposition. Then apply it to the task of face recognition (for example, see [1]). In this project, the photo can be converted into a matrix using any standard application or package. What happens when individual factors in the singular value decomposition are changed?

The major goal is to modify the method and achieve the best recognition.

Extra. Kernel methods offer the advantage of capturing and restoring non-linear relationships between variables. By incorporating additional features without significantly increasing computational complexity, kernels enhance the performance of machine learning algorithms.
In this project, we propose the implementation of Kernel SVD for face recognition, employing Gaussian and polynomial kernels. For kernel parameters selection, the dataset should be divided into validation and testing parts. It is also proposed to test the approach on a dataset for comparing grouped images of faces using kernel principle angles [6,7].

References:

1. Shevtsov G. S. "Linear algebra: Theory and Applied aspects": textbook, 2010 — 525 sec.

2. J.Demmel "Applied Numerical Linear Algebra"

3. http://link.springer.com/chapter/10.1007%2F978-1-4020-6264-3_26

4. TianY, TanT, WangY, FangY: Do singular values contain adequate information for face recognition? Pattern Recogn. 2003, 36: 649–655.

Additional:

5. Chin, Tat-Jun & Suter, D.. (2006). Incremental kernel SVD for face recognition with image sets. 461- 466. 10.1109/FGR.2006.67. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.137.8242&rep=rep1&type=pdf

6. L. Wolf and A. Shashua. Learning over sets using kernel principal angles. JMLR, 4:913–931, 2003

7. Kernels with python: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.kernel_metrics.html#sklearn.metrics.pairwise.kernel_metrics

2. Incremental SVD for Latent Semantic Index

The proposed approach aims to identify pertinent documents for input queries by utilizing the SVD decomposition of the term-document matrix. It is important to acknowledge that the amount of data continues to expand, and recalculating the SVD directly each time is not efficient. Therefore, in order to address the scalability issue, two incremental SVD techniques, as outlined in the article [1], namely folding-in and recomputing SVD, need to be implemented. As part of the experimentation, a comparison between the speed of direct SVD recalculation and recalculation using incremental SVD for large input matrices should be conducted. Additionally, the obtained word and document embedding space should be visualized. To represent high-dimensional vectors on a two-dimensional plane, a dimensionality reduction method should be employed, and Principal Component Analysis (PCA) [2] is recommended for this purpose.

The utilization of matrix factorization holds potential in a diverse range of machine learning applications. So, it is suggested to explore the correlation between Latent Semantic Indexing (LSI) and collaborative filtering in recommendation systems, specifically in the context of the user-item rating matrix [3]. Consequently, it is essential to find an appropriate dataset, conduct experiments, explore the incorporation of new items, and analyze the subsequent recalculation of recommendations.

References:

1. Berry, Michael & Susan, & Dumais, T. (2000). Using Linear Algebra For Intelligent Information Retrieval. SIAM Review. 37. 573-595. 10.1137/1037127. https://pdfs.semanticscholar.org/0265/769b0fbf86bb0e700573c80e388bb54c3f7a.pdf

2. https://www.sartorius.com/en/knowledge/science-snippets/what-is-principal-component-analysis-pca-and-how-it-is-used-507186

3. https://towardsdatascience.com/intro-to-recommender-system-collaborative-filtering-64a238194a26

3. Linear Algebra and the Search Problem

Develop a program that implements the official open version of the PageRank algorithm employed by the renowned search engine. The algorithm is covered in the lecture, therefore it is expected that the dataset will be substantial and atypical, and utilizing the ranking algorithm will yield significant insights. The feasibility of incorporating more advanced algorithms will be assessed additionally.

References:

1. *K Bryan, T Leise The $25,000,000,000 eigenvector: The linear algebra behind Google - Siam Review, 2006 - SIAM*

4. Quadratic optimization

The problem of quadratic optimization (or quadratic programming) is a generalization of the least squares method. It requires finding a solution to a system of linear equations that minimizes some quadratic function (the generalized version of the “squared error”) under constraints defined in the form of linear inequalities. Several effective methods for solving this problem are known. We suggest to study these methods, implement them (or use already existing packages) and compare their effectiveness for the case of economic statistics - the reconstruction of tables of intersectoral transactions in the scale of national economy.

References:

1. For quadratic optimization algorithms, see Pshenichny BN, Danilin Yu.M. “Numerical methods in extreme problems”, chap. III, paragraph 1, or

2. Hadley J; “Nonlinear and dynamic programming”, chap. 7

3. For the more contemporary examples of general optimization methods, see: Y.E. Nesterov. “Convex optimization methods” (in Russian)

4. An example of economic challenges can be found here: http://www.wiod.org/publications/papers/wiod2.pdf, paragraphs 2.7 and 2.8

5. Multidimensional arrays, TT decomposition and big data analysis

Illustrate the use of the “tensor-train decompositions” or their analogues in problems of big data analysis.

References:

1. http://epubs.siam.org/doi/abs/10.1137/090752286

2. I. V. Oseledets and E. E. Tyrtyshnikov. “TT-Cross approximation for multidimensional arrays” INM RAS Preprint, 2009-05.

3. http://people.csail.mit.edu/moitra/docs/bookex.pdf

6. Solution of systems of algebraic equations and inequalities

Modern symbolic approaches for solving systems of algebraic equations and inequalities are associated with the Gröbner bases and Cylindrical Algebraic Decomposition. It is assumed that Groebner bases and their associated algorithms will be discussed on the lectures.

The primary objective of the project is to investigate and demonstrate the practical implementation of a method for resolving systems of algebraic inequalities using cylindrical decomposition. This entails expounding on the fundamental algorithms involved and providing illustrative examples throughout the project. For further details, please consult chapters 5 and 11 in reference [1], as well as [2].

References:

1. Basu, Saugata; Pollack, Richard; Roy, Marie-Françoise “Algorithms in real algebraic geometry.” Second edition. Algorithms and Computation in Mathematics, 10. Springer-Verlag, Berlin, 2006. x+662 pp. ISBN 978-3-540-33098-1; 3-540-33098-4 ; author's edition — https://perso.univ-rennes1.fr/marie-francoise.roy/bpr-ed2-posted3.pdf

2. Mats Jirstrand. “Cylindrical Algebraic Decomposition – an Introduction” — http://www.diva-portal.org/smash/get/diva2:315832/FULLTEXT02

7. Methods of the Krylov subspace for solving linear systems

The objective of this project entails the numerical solution of equations encompassing mathematical physics (for example, the Poisson equation), employing one of the Krylov subspace methods. The project presentation should strive to elucidate the physical significance of the solution, while also describing potential experiments that can be derived.

References:

1. J.Demmel "Applied Numerical Linear Algebra"

8. Tropical linear algebra and scheduling

Illustrate the use of operations on a tropical semiring, i.e. an "algebra" with two operations - "maximum" and "sum".

References:

1. P. Butkovic. Max-Linear Systems: Theory and Algorithms. Springer, 2010.

2. Francois Louis Baccelli, Guy Cohen, Geert Jan Olsder, Jean-Pierre Quadrat, "Synchronization and Linearity: An Algebra for Discrete Event Systems" John Wiley & Sons, 1993, pp. 514

9. Spectral clustering

Clustering is the task of dividing a set of objects into groups of "similar" elements. It is widely in data analysis having applications in various fields, ranging from statistics, computer science, biology, to social sciences or psychology.

As a project, it is proposed to study the spectral clustering method and try to apply this algorithm to real data. Implementation of the algorithm can be found in the python package *sklearn*. It is necessary to illustrate it on meaningful datasets and compare it with combinatorial clustering methods (k-NN and others).

References:

1. http://people.csail.mit.edu/dsontag/courses/ml14/notes/Luxburg07_tutorial_spectral_clustering.pdf

2. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.SpectralClustering.html

10. Low-Rank Adaptation of Large Language Models

Recently, there has been a significant surge in the widespread usage of large language models (LLMs), including GPT-3, Chat-GPT, LLaMa, and others. However, the training of these models necessitates substantial computational resources in the form of sizeable clusters, despite the relatively small size of the datasets employed.

This project seeks to investigate LoRA-based methods for fine-tuning LLMs, employing low-rank approximation techniques to the trainable parameter matrices.

The first objective of this project is to choose a specific text generation task and fine-tune the GPT-2 model [4] in two different variations: the base model and the model utilizing LoRA [1,2]. It is necessary to estimate the number of trainable parameters (memory costs), training time and quality (for example, using ROUGE or BLEU) depending on the rank in the LoRA approximation.

It is important to acknowledge that while LoRA can significantly reduce the number of trainable parameters, it does not guarantee an optimal approximation. To address this limitation, a modification called AdaLoRA has been proposed, which imitates truncatedSVD. Thus, the second project objective is to adapt the code from [2] for the SVD-LoRA model (as described in [3]). A comparison between LoRA and SVD-LoRA models should be conducted.

Prerequisites: (1) basic understanding of the concept of neural networks (language models) and their training; (2) basic knowledge of keras

References:

1) LoRA paper: https://arxiv.org/pdf/2106.09685v1.pdf

2) Starting keras code for LoRA training based on GPT-2 [you need to choose the dataset yourself]: https://colab.research.google.com/github/keras-team/keras-io/blob/master/examples/nlp/ipynb/parameter_efficient_finetuning_of_gpt2_with_lora.ipynb

3) AdaLoRA paper: https://arxiv.org/pdf/2303.10512.pdf

4) GPT-2 paper: https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf*

5) Pytorch code for AdaLora (you can use an analogy for keras):*
https://github.com/QingruZhang/AdaLoRA

Implementation of SVDLinear layer and regularization loss:
https://github.com/QingruZhang/AdaLoRA/blob/main/loralib/loralib/adalora.py

6) Code for PEFT(State-of-the-art Parameter-Efficient Fine-Tuning (PEFT) methods):*
https://github.com/huggingface/peft

7) LoRA for Computer Vision (image generation):*

https://stable-diffusion-art.com/lora/

https://huggingface.co/blog/lora

https://github.com/cloneofsimo/lora - code from the authors of the main paper