# Matrix representation as sum of local low rank matricies

## Projectplan

Stepahn Nüßlein

18. November 2021

## Motivation

- Matrices for fully connected Layers get larger -> Computational expensive
- Matrix approximations with reduced computational cost
- Both approxiamtion of Matrix in some Norm important ($\|\|_F$,$\|\|_1$,$\|\|_\infty$ or similar) also Accuracy on Data set

# 1 Project Description

## Goals

- Find Matrix approximation with reduced cost
- Find algorithm to construct said approximation
- Get some theoretical predictions on the behavior

## Approach

These assumptions and properties are underlying the idea:

- Matrices in Fully connected layers often have full rank -> so no rank reduction easily possible (with some caveats, as it is not certain that this is the relevant ingredient) [1]
- The order of inputs/outputs is not important $A$ is equivalent to $\Pi_o A \Pi_i$ with $\Pi_i$ and $\Pi_o$ permutation matrices as the ordering of the neurons can be changed.
- To get the optimal trade of between accuracy and cost it should be possible to set them at a late stage in the algorithm

Approximate the given Matrix $A \in \mathbb{R}^{n \times m}$ using the representation

$$A = \sum_{i=1}^{k} a_i u_i v_i^\top \tag{1}$$

With $k$ not necessarily $\leq \min(n,m)$ and most of $u_i$ and $v_i$ sparse. Without loss of generality we can assume $a_i \geq 0$. The $a_i$ are ordered in decreasing order. Therefore the vectors $u_i$ and $v_i$ have some constraint on their norm. Also sparsity has to be defined in a more rigid way.

The expression can be viewed as a modification of the SVD. Alternatively the weighted sum can be expressed as a Matrix-vector product.

$$\mathrm{vec}(A) = \sum_{i=1}^{k} a_i \,\mathrm{vec}(u_i v_i^\top) = \left[\mathrm{vec}(u_1 v_1^\top), \cdots, \mathrm{vec}(u_k v_k^\top)\right] a = \left[v_1 \otimes u_1, \cdots, v_k \otimes u_k\right] a \tag{2}$$

It is possible to express different matrix structures with the proposed representation. Hirarchical matricies can be converted trivially. Sequentially semiseperable and semiseperable are technically possible. Also said structures with row and/or collumn permutations and sum of said are possible.

[Note: Additional further ideas: Maybe transformation in $P_1 T_{[A,B,C,D]} P_2$ where the $P$s are (sub)permutations and the $T$ is a state Space Model]

Unlike the hierarchical matrices the low rank structures are overlapping. This poses the risk that it might be ill defined and the prone to numerical errors. Compare it to (2). The vectors might be linearly dependent. Therefore it might be worthwhile to look into convergence and the properties of Matrix containing the basis vectors (Maybe something from Funktionalanalysis, look at constructions of Basis->how many entries, condition..., Any matrix can be expressed if we have $k = nm$ and the $v_i \otimes u_i$ are not linearly dependent, but this case is not desirable)

Risk: Computations not cheaper: if the $u_i$ and $v_i$ are not sparse enough and we need to much of them the computation will not get cheaper TODO: some basic approxiamtion on the computation cost

# of multiplications Regular matrix vector:

$$\approx nm$$

Representation:

$$\approx \sum_{i=1}^{k} \|u_i\|_0 + \|v_i\|_0$$

With $\|x\|_0$ the "normöf nonzero elements.

Possible ideas to get a said representation

**Optimization Problem** Convert the given properties into manifolds and objective functions and use optimization techniques. This might be easier to define, but will possibly result in a problem with a very high number of dimensions. Maybe it is possible to express it as some standard problem, then it would be possible to use standard solvers and use existing guarantees and runtime estimates.

**Iterative Methods** Use some guesses that are updated periodically. A SVD-Based iterative approach seems reasonable. -> Use SVD, set vectors with large $\sigma$ aside, change vectors to make them sparse according to some measure, get residuum and decompose it again using SVD and repeat the process. This might also a update step for old vectors. This approach maybe more efficient, as the number of dimensions is not as high, but deriving properties like convergence might be not possible in this thesis. Also K-SVD might be interesting.

## 2 Workpackages

- **Recherche:**
  - **Existing Decompositions:** Low Rank+Sparse, Multiscale Low rank,Hierarchical [2], [3]
  - **Sparsity:** Appropriate Sparsity Measures [4], [5]
  - **Random Matrix Theory:** Some ideas on structures in Random Matricies, maybee intersting to get an idea what computational cost we can expect
  - **Existing Algorithms:** Look into the structure of related algorithms
- **Evaluierung:** Analyse der bisherigen Recherce
  - **Auswahl:** choose appropriate assumtions
  - **Auswahl:** Evaluate if computational benefits can be expected

  - **Theoretical considerations:** Uniqueness, numerical stability, computations, can we guarantee that it is as least as good as the other matrix classes that can be represented with it.... and under which assumptions.

- **Implementierung:** Implement Algorithm
  - **Implement Algorithm(s):** Based on previous exploration choose algorithm to construct the representation. If uncertain implement two different approaches and try them on examples.
  - **Generate Tests:** Generate Testsdata and pipeline to test them, including meta-analysis (speed of convergence...)
- **Analyse:** Entwurf und Durchführung praxisnaher Tests
  - **Setup Test**
  - **Evaluation I:** Evaluation on tailor-made examples: Sum of Low Rank + low Rank sub matrix (I think we need more than $\min(n, m)$ rank sub matrices until the standard SVD not be able to recover the structure?, but even then it might be interesting, as we can reduce the computations)
  - **Evaluation II:** Evaluation on low rank + Random with different parameters
  - **Evaluation III:** Evaluation on AI maricies
- **Auswertung und Diskussion:** Ergebnisse zusammentragen und vergleichen
  - **Revisiting the Theory:** Explanations for experienced behavior?
  - **Description of Performance:** Beschreibung der durchgeführten Tests und Visualisierung der Ergebnisse
  - **Diskussion:** Auswertung der Ergebnisse und kritische Betrachtung des Nutzens für das Gesamtsystem
- **Ausarbeitung:** Abschließende schriftliche Darstellung der durchgeführten Arbeiten

## 3 Time Table

| **Month** | | December | | | | | January | | | | February | | | | March | | | | | April | | | | May | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Week** | 47 | 48 | 49 | 50 | 51 | 52 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| **Recherchen** | 🟥 | 🟥 | 🟥 | 🟥 | | | | | | | | | | | | | | | | | | | | | | | |
| **Evaluierung** | | | | | 🟥 | 🟥 | 🟥 | | | | | | | | | | | | | | | | | | | | |
| **Implementierung** | | | | | | | | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | | | | | | | | | | | | | | |
| **Analyse** | | | | | | | | | | | | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | | | | | | | | | | |
| **Auswertung** | | | | | | | | | | | | | | | | | | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | | | | | |
| **Ausarbeitung** | | | | | | | | | | | | | | | | | | | | | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 |

# 4 Risk Analysis

Computation effort might not be lower than the effort for regular Matrix vector Product.
**Likelihood:** Risk is hard to determine upfront as it depends on how fast the $a_i$ decrease and on how sparse the $u_i$ and $v_i$ are.
**Mitigation:** Derive the cost and the allowed cost in an early stage of the thesis, consider Transformation in other structure.

Unable to find an appropriate algorithm
**Likelihood:** Quite likely as it is a very underdetermined problem with many parameters
**Mitigation:** use more computation-time, reduction of degrees of freedom by requiring some artificial condition (e.g. set the structure of the submatrices)

Performance will not be good
**Likelihood:** unlikely as other student work with different structures have shown that they perform quite well. But it is unclear if it will outperform the cropped SVD
**Mitigation:** difficult as it is very inherent to structure, also unlikely that it will be possible to give a definite answer that it is impossible, if the efforts fail.

## Literatur

[1] C. H. Martin und M. W. Mahoney, “Implicit Self-Regularization in Deep Neural Networks: Evidence from Random Matrix Theory and Implications for Learning,” *arXiv:1810.01075 [cs, stat]*, 2. Okt. 2018.

[2] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo und A. S. Willsky, “Sparse and low-rank matrix decompositions,” *IFAC Proceedings Volumes*, 15th IFAC Symposium on System Identification, Jg. 42, Nr. 10, S. 1493–1498, 1. Jan. 2009.

[3] F. Ong und M. Lustig, “Beyond Low Rank + Sparse: Multiscale Low Rank Matrix Decomposition,” *IEEE Journal of Selected Topics in Signal Processing*, Jg. 10, Nr. 4, S. 672–687, Juni 2016, Conference Name: IEEE Journal of Selected Topics in Signal Processing.

[4] M. Ulfarsson, V. Solo und G. Marjanovic, “Sparse and low rank decomposition using l0 penalty,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, ISSN: 2379-190X, Apr. 2015, S. 3312–3316.

[5] A. Parekh und I. W. Selesnick, “Improved sparse low-rank matrix estimation,” *Signal Processing*, Jg. 139, S. 62–69, 1. Okt. 2017.