# Week 4

## Master Thesis 2020

Tobias Engelhardt Rasmussen (s153057)

DTU Compute

23. september 2020

# Outline

# Since last

- Project Plan done
- Draft for introduction
- Reading
- Decomposing 3s and 4s

# Outline

# Decomposition methods overview

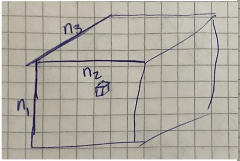
- Tucker
  - One rank for each dimension
  - One loading matrix for each dimension of size $n_d \times r_d$
  - One core tensor of size $r_1 \times r_2 \times \ldots$
  - Possible to choose how much variation we allow in each dimension
- Parafac / Candecomp / CP / Canonical Decomposition
  - Special case of the Tucker model with the same rank for each dimension (sum of outer products of vectors)
  - Core is a super-diagonal, hence less freedom than in Tucker
- Block Term Decomposition
  - A sum of tucker terms, which allow for different ranks for the same dimension in different terms
- Tensor-Train (TT) decomposition / Matrix Product State
  - Sequence of order-3 tensors
  - Works well for high dimensions since robust wrt the curse of dimensionality

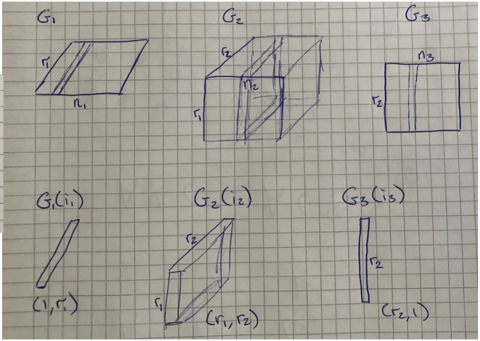
# TT - decomposition

We have ranks $r_0, r_1, \ldots, r_d$ where we need $r_0 = r_d = 1$

$$A(i_1, i_2, \ldots, i_d) = G_1(i_1)G(_2(i_2)\ldots G_d(i_d)$$

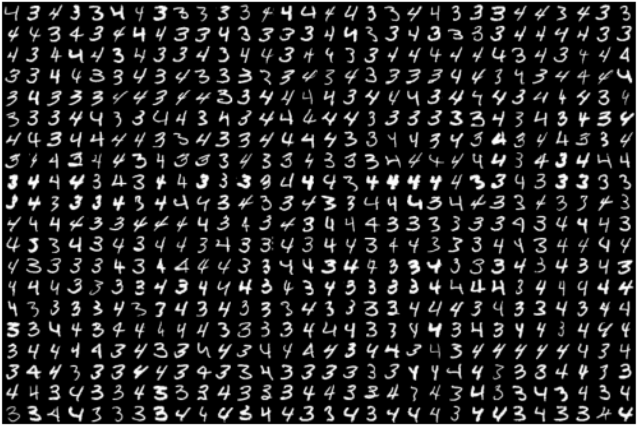
Where $G_k(i_k)$ is $r_{k_1} \times r_k$ matrix



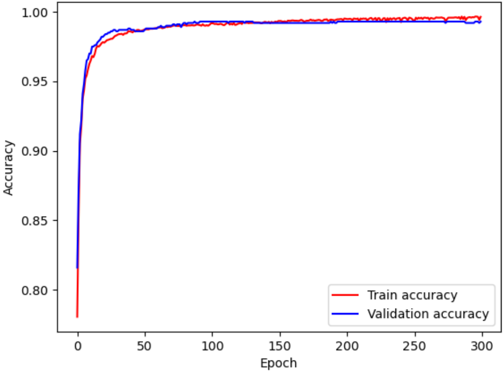
# Outline

# Relatively simple!



# Results with just 1 hidden neuron

- Just **1** hidden neuron
- Testing accuracy of $> 99\%$

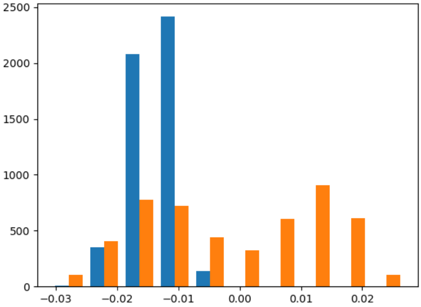


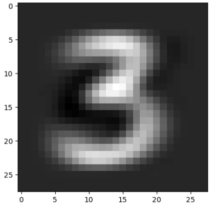
# Tucker - decomposition

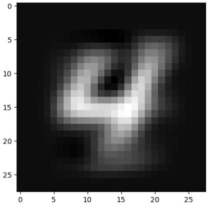
Using the Tucker(2,10,10) decomposition.


Looks like "archetypes"of 3 and 4.

# The first loading matrix



# The training set

Assume B, C and $\mathcal{G}$ are the same.

$$\mathcal{X} = \mathcal{G} \times_1 A \times_2 B \times_3 C$$

$$\Updownarrow$$

$$\mathcal{G} \times_1 A = \mathcal{X} \times_2 B^\dagger \times_3 C^\dagger$$

$$\Updownarrow$$

$$A\ \mathcal{G}_{(1)} = (\mathcal{X} \times_2 B^\dagger \times_3 C^\dagger)_{(1)}$$

$$\Updownarrow$$

$$A = (\mathcal{X} \times_2 B^\dagger \times_3 C^\dagger)_{(1)}\ \mathcal{G}_{(1)}^\dagger$$

Which can then be used to estimate the loadings of A for the training data

# Results using first loading matrix

- Just **1** hidden neuron
- Just **2** input neurons!!!
- A total of **8** weights and **2** biases
- Testing accuracy of $\approx 97\%$ ($\approx 95\%$ with only 1 hidden neuron)