Week 8 / 10

Master Thesis 2020

Tobias Engelhardt Rasmussen (s153057)

DTU Compute

November 19, 2020

Outline

Since Last

- Writing Methods section
- Implemented the use of VBMF for rank selection
- New data
  - Pre-processing
  - Big-data

Outline

Methods section

Outline:

- Decomposing the Input - Example with 3s and 4s (maybe 4s and 9s?)
- Decomposing pretrained network (not finished)
- (Using decomposition in training)

Example with 3s and 4s

Using just Tucker-1 decomposition as we are only interested in input-dimension, hence:

$$\mathcal{X} \approx \mathcal{G} \times_1 \boldsymbol{A} \times_2 \boldsymbol{I} \times_3 \boldsymbol{I}$$

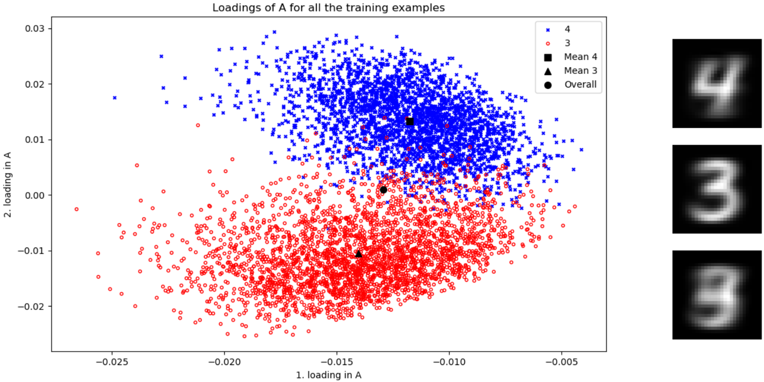


Figure: Scatter plot of the 2 loading values for each picture incl. means

Tucker decomposition of the linear layer

$$\texttt{Tucker-1}: \quad \boldsymbol{W} \approx \boldsymbol{G} \times_1 \boldsymbol{A} \times_2 \boldsymbol{I} = \boldsymbol{A}\boldsymbol{G} \quad \text{OR} \quad \boldsymbol{W} \approx \boldsymbol{G} \times_1 \boldsymbol{I} \times_2 \boldsymbol{B} = \boldsymbol{G}\boldsymbol{B}^\top$$
$$\texttt{Tucker-2}: \quad \boldsymbol{W} \approx \boldsymbol{G} \times_1 \boldsymbol{A} \times_2 \boldsymbol{B} = \boldsymbol{A}\boldsymbol{G}\boldsymbol{B}^\top$$

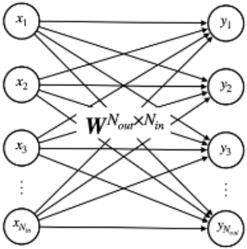
Method	# Multiplications	Faster when:
Uncrompressed	$N_{out}N_{in}$	-
Tucker-1	$R(N_{in} + N_{out})$	$R < \frac{N_{out}N_{in}}{N_{out}+N_{in}}$
Tucker-2	$N_{in}R_B + R_BR_A + R_AN_{out}$	$(N_{in} + R_A)(N_{out} + R_B) < 2N_{in}N_{out}$

be decomposed.[3] Inserting the decomposition of $\boldsymbol{W}$ in the linear transform (4) yields:

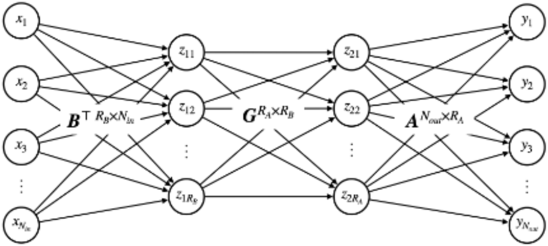
$$\texttt{Tucker-1}: \quad \begin{array}{c} \boldsymbol{y}^{N_{out}} \approx \boldsymbol{A}^{N_{out}\times R_A}\boldsymbol{G}^{R_A\times N_{in}}\boldsymbol{x}^{N_{in}} + \boldsymbol{b}^{N_{out}} \\ \text{OR} \\ \boldsymbol{y}^{N_{out}} \approx \boldsymbol{G}^{N_{out}\times R_B}\boldsymbol{B}^{\top\ R_B\times N_{in}}\boldsymbol{x}^{N_{in}} + \boldsymbol{b}^{N_{out}} \end{array}$$

$$\texttt{Tucker-2}: \quad \boldsymbol{y}^{N_{out}} \approx \boldsymbol{A}^{N_{out}\times R_A}\boldsymbol{G}^{R_A\times N_{in}}\boldsymbol{B}^{\top R_B\times N_{in}}\boldsymbol{x}^{N_{in}} + \boldsymbol{b}^{N_{out}}$$

Tucker decomposition of the linear layer



(a) Original linear layer



(b) Sequence of linear layers resulting after decomposing $\boldsymbol{W}$

Outline

VBMF

Using the analytical solution to Variational Bayesian Matrix Factorization (VBMF) to estimate the optimal ranks. Simply taken from an implementation based on a paper.

For MNIST (feats. in, feats. out, kernel size)

Layer	**Original**	**Decomposed**	**Rank(s)**
Conv1	$(1, 6, 5)$	$(1, 2, 5) \to (2, 6, 1)$	_,2
Conv2	$(6, 16, 5)$	$(6, 3, 1) \to (3, 6, 5) \to (6, 16, 1)$	3, 6
Lin1	$(400, 120)$	$(400, 20) \to (20, 22) \to (22, 120)$	20, 22
Lin2	$(120, 84)$	$(120, 9) \to (9, 84)$	9,_

Results

	Original	**Decomposed**	**Ratio**
Test acc.(%)	98.097	98.498	1.004
# Parameters	61,706	14,618	0.237
Time (ns)	1,758,916	3,574,018	2.0319

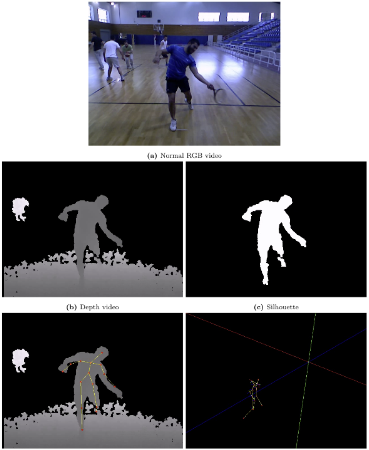
VBMF

- Under 1/4 of the parameters but takes twice the time...
  - Due to simplicity of problem
  - 1×1 convolution cache-inefficient
- Speed-up for bigger networks (tried VGG-16)
  - Hopefully will speed-up tennis dataset

Outline

New data

- 1980 tennis shots
- 12 types of shot
- 55 individuals
  - 31 beginners
  - 24 Experts
- 5 different features videos
- 480 × 640 resolution (with color makes 921,600 pixels per frame)



(a) Normal RGB video (b) Depth video (c) Silhouette (d) 2D Skeleton (e) 3D skeleton

Variance in the data

- Both left and right handed individuals
- Males and females
- 2 different locations
  - Sports arena with people playing basketball or doing yoga
  - Changing room with no noise
- Different length of videos
- Different speed of the shot (some fast / some very slow)
- Moving individuals and individuals standing still

Data pre-processing

Problems:

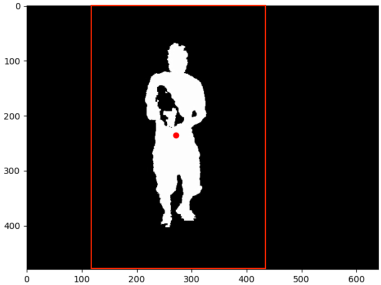
- Videos need to be the same length to have same shape (for learning)
- Videos are big and have a lot of unused background

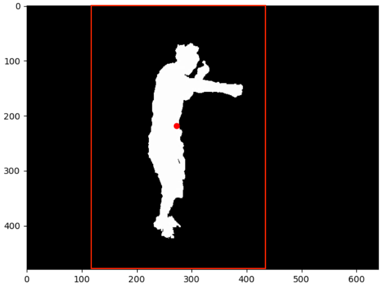
Ideas:

- Making the videos the same length by manually finding the time for the middle of the stroke
- Removing some background by using the mask (silhouette) to find the person and taking a width on both sides. This way all the players will also be centered during their shot.

Center of Mass

Using the center of mass in the silhouette to find the region of interest, which can then be subtracted from the other movies.





Should be able to approximately half the number of pixels

Inital Goal

- Taking out 2 distinct types of shots - forehand and backhand
- Using the two decomposition methods on this binary classification problem
  - Doing a CNN (now 3D convolutions) and use tucker-decomposition - does it generalise?
  - Decomposing the input to see if this holds enough information for an ANN
- Will use the big computers at DTU to do the decomposition and the training