Week 12

Master Thesis 2020

Tobias Engelhardt Rasmussen (s153057)

DTU Compute

December 3, 2020

Outline

Since last

- Implementation
  - Video functions
  - CNN for THETIS (videos)
  - cuda in hpc clusters
- Compressing the trained network

Outline

Data pre-processing

- Problems with a few videos that where removed (3)
- Resolution was decreased by a factor of 4 from (640×480) to (160×120) decreasing the number of pixel values by a factor of 16 from 307,200 to 19,200
- All videos standardized to have same length (number of frames) around the middle of the shot. Length 1.5 s gives 28 frames per video (some videos have different frame rates)
- The B/W depth video (1 input channel) was added to the RGB video, resulting in video shape of:

$$(\mathit{channels}, \mathit{frames}, \mathit{height}, \mathit{width}) = (4, 28, 120, 160)$$

CNN for THETIS

Using the so-called "slow fusion" method which is using a 3D filter able of exploiting both spatial features, but also movement in time. Seems intuitively superior to the other methods, since the goal is to classify one specific task instead of an activity in general.

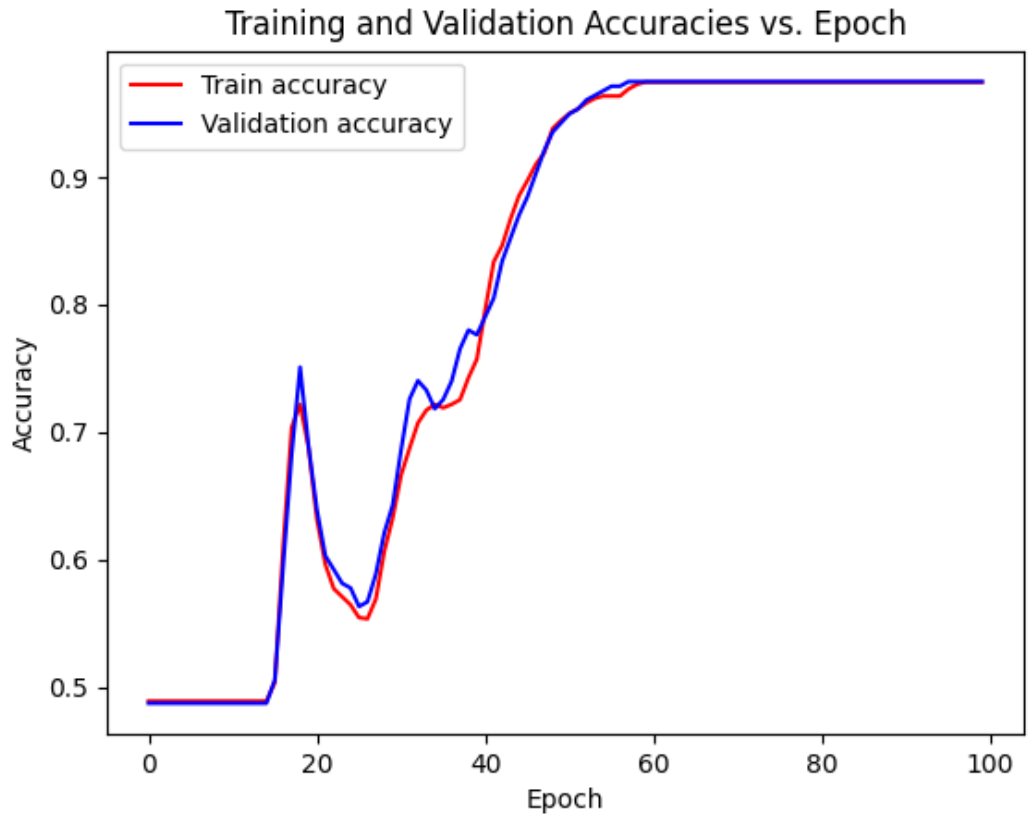
Architecture:

Layer	3D Conv	3D Max Pool	3D Conv	3D Max pool	Linear	Linear	Linear
Input ch	4	6	6	16	2800	128	64
Output ch	6	6	16	16	128	64	2
Kernel	(5, 11, 11)	(2, 4, 4)	(5, 11, 11)	(2,4,4)	-	-	-
Padding	yes	no	no	no	-	-	-

Results for original (full) network

Total data set of **327** videos of forehands and backhands, trained holding back **15%** $= 50$ for testing. Validated using **5-fold cross validation**.

	Training	Validation	Testing
Accuracy	97.47 %	97.49 %	92 %



Decomposed network

Works since we can do the same analytically as for images. The 3D convolution:

$$\mathcal{Y}_{f',h',w',t} = \sum_{i=1}^{D_f} \sum_{j=1}^{D_h} \sum_{l=1}^{D_w} \sum_{s=1}^{S} \mathcal{K}_{i,j,l,s,t} \; \mathcal{X}_{f_i,h_j,w_l,s}$$

Where:

$$f_i = (f'-1)\Delta + i - P_f \quad h_j = (h'-1)\Delta + j - P_h \quad w_l = (w'-1)\Delta + l - P_w$$

Are indices in the input tensor given the padding P, stride Δ, and indices in the output tensor (f', h', w').

Tucker-decomposition of the kernel

The tucker-2 decomposition of the 5D kernel $\mathcal{K}$ of the input and output channels gives:

$$\mathcal{K}_{i,j,l,s,t} = \sum_{r_4=1}^{R_4} \sum_{r_5=1}^{R_5} \mathcal{C}_{i,j,l,r_4,r_5} \boldsymbol{U}^{(4)}_{s,r_4} \boldsymbol{U}^{(5)}_{t,r_5} \tag{2.1}$$

Where $\mathcal{C}$ is the 5D core of size $D_f \times D_h \times D_w \times R_4 \times R_5$, the $\boldsymbol{U}$s are the loading matrices. Inserting (2.1) into the convolution gives:

$$\mathcal{Y}_{f',h',w',s,t} = \sum_{i=1}^{D_f} \sum_{j=1}^{D_h} \sum_{l=1}^{D_w} \sum_{s=1}^{S} \sum_{r_4=1}^{R_4} \sum_{r_5=1}^{R_5} \mathcal{C}_{i,j,l,r_4,r_5} \boldsymbol{U}^{(4)}_{s,r_4} \boldsymbol{U}^{(5)}_{t,r_5} \mathcal{X}_{f_i,h_j,w_l,s}$$

This rearranges into:

$$\mathcal{Y}_{f',h',w',s,t} = \sum_{i=1}^{D_f} \sum_{j=1}^{D_h} \sum_{l=1}^{D_w} \sum_{r_4=1}^{R_4} \sum_{r_5=1}^{R_5} \mathcal{C}_{i,j,l,r_4,r_5} \boldsymbol{U}^{(5)}_{t,r_5} \underbrace{\sum_{s=1}^{S} \boldsymbol{U}^{(4)}_{s,r_4} \mathcal{X}_{f_i,h_j,w_l,s}}_{\mathcal{Q}}$$

Tucker-decomposition of the kernel

$$\mathcal{Y}_{f',h',w',s,t} = \sum_{i=1}^{D_f} \sum_{j=1}^{D_h} \sum_{l=1}^{D_w} \sum_{r_4=1}^{R_4} \sum_{r_5=1}^{R_5} \mathcal{C}_{i,j,l,r_4,r_5} \boldsymbol{U}^{(5)}_{t,r_5} \mathcal{Q}_{f_i,h_i,w_i,r_4}$$

Again:

$$\mathcal{Y}_{f',h',w',s,t} = \sum_{r_5=1}^{R_5} \boldsymbol{U}^{(5)}_{t,r_5} \underbrace{\sum_{i=1}^{D_f} \sum_{j=1}^{D_h} \sum_{l=1}^{D_w} \sum_{r_4=1}^{R_4} \mathcal{C}_{i,j,l,r_4,r_5} \mathcal{Q}_{f_i,h_i,w_i,r_4}}_{\mathcal{Q}'}$$

$$\mathcal{Y}_{f',h',w',s,t} = \sum_{r_5=1}^{R_5} \boldsymbol{U}^{(5)}_{t,r_5} \mathcal{Q}'_{f',h',w',r_5}$$

Tucker-decomposition of the kernel

Now the three intermediate tensors:

$$Q_{f,h,w,r_4} = \sum_{s=1}^{S} \boldsymbol{U}^{(4)}_{s,r_4} \mathcal{X}_{f,h,w,s}$$

$$\mathcal{Q}'_{f',h',w',r_5} = \sum_{i=1}^{D_f} \sum_{j=1}^{D_h} \sum_{l=1}^{D_w} \sum_{r_4=1}^{R_4} \mathcal{C}_{i,j,l,r_4,r_5} \mathcal{Q}_{f_i,h_i,w_i,r_4}$$

$$\mathcal{Y}_{f',h',w',s,t} = \sum_{r_5=1}^{R_5} \boldsymbol{U}^{(5)}_{t,r_5} \mathcal{Q}'_{f',h',w',r_5}$$

Correspond to convolutions with kernels $(1 \times 1 \times 1)$, $(D_f \times D_h \times D_w)$, and $(1 \times 1 \times 1)$ respectively.

Results for compressed network

Number of parameters	3D conv 1	3D conv 2	Linear 1	Linear 2	Linear 3	**Total**
Original	14,520	58,080	358,400	10,752	168	441,920
Compressed	4,852	3,690	14,266	212	168	23,188
Ratio	0.334	0.064	0.039	0.020	1	0.052

Accuracy	Training	Validation	Testing
Original	97.47 %	97.49 %	92 %
Compressed	97.09 %	97.11 %	90 %