Deep representation learning for human motion prediction and classification

Judith Bütepage Michael J. Black Danica Kragic Hedvig Kjellström

A. Appendix

In this section, we give a detailed account of the implementation and training procedure of the models.

A.1 Detailed model structure

The input and output data for all models is of dimension $3 \times N_{joints} \times \Delta t = 3 \times 24 \times 100$. While these dimensions are retained for the input layer, the output layer is flattened to 7200 dimensions. The weights of all layers are initialized sparsely and with a Gaussian distribution (mean = 0, variance = 1). We apply a sigmoid non-linearity to all layers except the bottleneck layer, the output layer of the temporal encoders and the output layer of the classifiers. We do not apply any pooling. Instead of eliminating information by pooling or by applying a rectified linear non-linearity, it has proven to be crucial to let all information flow freely from the input to output layer.

We denote the dimension of a fully-connected layer by a single number of output dimensions D and the connection between layer D and E by D–E. Layers can be parallel in a network, i.e. they are all connected to parts of the same input and output layer but not to each other. When the neighbouring layers have the same structure Q, we denote this by $Qi_{j=1:M}$, where M is the number of such layers. Furthermore, we denote the number of output neurons N and the filter size of a convolutional layer by [N,Δt^w].

All layers of the symmetric temporal encoder are fully-connected. The structure of the network is depicted in Figure 1 a).

A number of layers in the convolutional encoder are convolutional with a filter of size $3 \times N_{joints} \times \Delta t^w$. The structure of the convolutional encoder is depicted in Figure 1 b).

Finally, the hierarchical temporal encoder contains 24 parallel layers in the first level of the hierarchy representing a single joint dimension in the data layer. The second layer combines these nodes into separate limbs. In the following, the arms and legs are summarized in a node each and form the body together with the trunk in the subsequent layer. This network is depicted in Figure 1 c).

The structure of the classifier for input data with dimension N and M classes is N–50–20–M, where the last layer is a softmax layer.

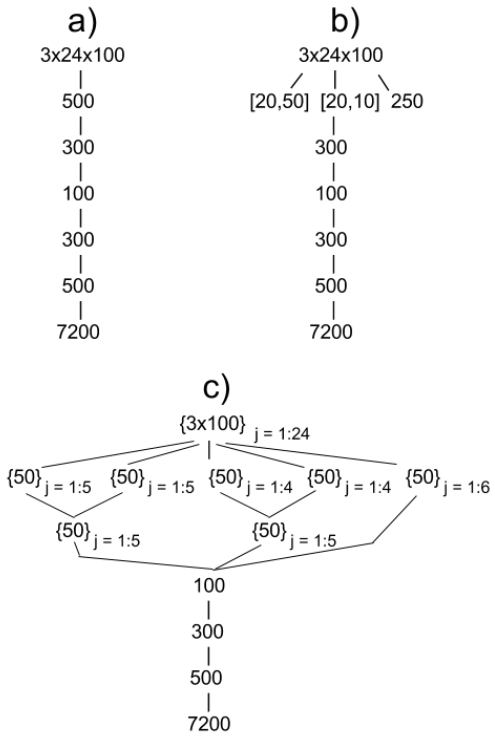

Figure 1: The structure of the symmetrical temporal encoder (a)), the convolutional temporal encoder(b)) and the hierarchical temporal encoder (c))

A.2 Training details

We implement and train our models with help of the Caffe deep learning framework [1]. For this, we shuffle the training and testing data randomly, making sure that the temporal structure between data points of the same recording disappears. We train mini-batches of 300 to 500 data points with a learning rate of 0.01, a weight decay of 0.0005 and momentum of 0.9. We apply an increasing amount of dropout on the input data layer, ranging from 0.1 to 0.3.

A.3 Missing data - Additional results

Following the description in Section 4.5, we evaluate the prediction error for different actions when data of a single

limb is missing. In Table 1 we present the prediction error for the actions *walking, smoking* and *discussion* for a missing left leg and a missing right arm respectively.

Table 1: Motion prediction error, single actions

Method	Short Term			Long Term	
	80ms	160ms	320ms	560ms	1000ms
Walking - missing left leg					
S-TE	0.48	0.51	0.52	0.55	0.55
C-TE	0.41	0.45	0.45	0.5	0.55
H-TE	0.38	0.41	0.42	0.46	0.53
Walking - missing right arm					
S-TE	0.43	0.43	0.44	0.49	0.52
C-TE	0.37	0.39	0.41	0.44	0.5
H-TE	0.38	0.39	0.38	0.4	0.44
Smoking - missing left leg					
S-TE	0.41	0.42	0.43	0.43	0.55
C-TE	0.38	0.38	0.4	0.43	0.44
H-TE	0.38	0.38	0.39	0.4	0.42
Smoking - missing right arm					
S-TE	0.38	0.38	0.4	0.42	0.5
C-TE	0.34	0.34	0.37	0.42	0.44
H-TE	0.35	0.37	0.43	0.43	0.41
Discussion - missing left leg					
S-TE	0.35	0.35	0.38	0.44	0.46
C-TE	0.26	0.26	0.28	0.36	0.45
H-TE	0.28	0.29	0.29	0.34	0.34
Discussion - missing right arm					
S-TE	0.34	0.34	0.34	0.35	0.4
C-TE	0.24	0.25	0.29	0.32	0.32
H-TE	0.26	0.26	0.28	0.32	0.33

References

[1] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 1