Machine Perception Report - Moir4e

Roberto Pellerito Giacomo Manzoni Giovanni Cherubini

ABSTRACT

Human motion prediction aims at generating future frames of human motion based on an observed sequence of skeletons. Recent methods employ the latest hidden states of a recurrent neural network (RNN) or self-attention modules or Graph convolution Neural networks, to encode the historical skeletons, which can only address short-term prediction. The models were evaluated on AMASS dataset, where we were given sequences of 120 frames of pre-recorded human motion data and we predicted how the motion continues for 24 frames in the future. Naive approach implies feeding the entire sequence on an RNN and extracting the last 24 cells outputs. The latter can be improved using LSTM cells and some best pratices during training phase. This approach is outperformed by joint use of Transformers and Graph convolutional neural networks. We initially tackled the task with naive approach using LSTM cells, but after noticing scarse results both on validation set and visually, we followed the "current" state of the art, the work of Mao et al. [6], achieving satisfactory results.

1 INTRODUCTION

- Human motion prediction is relevant on applications where one needs to forecast the future, like in human-computer interaction, legged robotics and for autonomous navigation. Think for instance at an autonomous excavator navigating and interacting with object and people, in such an environment, forecasting human and object positions is a fundamental safety requirement. Predicting human motion is a common behavior that every driver has, therefore a fundamental block for self-driving cars.
- Human motion prediction for this project aims at short term motion predictions. For this reason to train and evaluate the models we used part of AMASS dataset, in LMDB format. It consists of given sequences of 144 frames 2.4 seconds of pre-recorded human motion data for training and 120 frames 2 s to test our inferred motion. The entire pipeline would ideally require to start with image frames, encode and extract human body shape, joint keypoints and then feed those on a model for sequential prediction. AMASS dataset directly contains the joint keypoints sequences.
- Because of the temporal nature of the signal of interest, the most common approach consists of using (RNNs) with for instance LSTM / GRU cells Fragkiadaki et al. [3] eventually in a sequence-to-sequence residual architecture, as introduced by Martinez et al. [7] We initially used the last method, with under state of art results, this method incorporates a nonlinear encoder and decoder before and after recurrent layers and a residual architecture. It was shown visually that predicted movements were usually slower in the inferred frames with respect to ground truth and predictions suffered from occasional discontinuities between the observed poses and predicted ones.

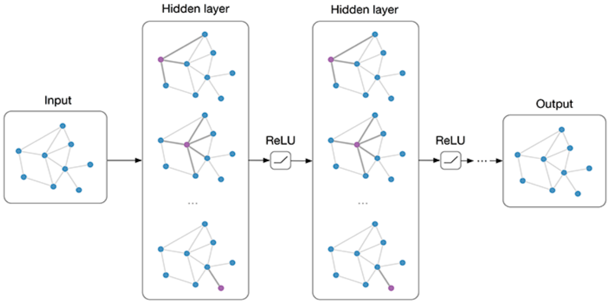


Figure 1: Graph Neural Network representation

- Extending the work of Mao et al.[6], these shortcomings were overtaken representing human motion trajectory with Discrete Cosine Transform (DCT). Indeed after discarding the high frequencies, DCT can provide a trajectory space representation that can captures smoothness of human motion. Upon that, we used a learnable graph convolutional networks to capture the spatial structure of the motion data. GCNs are networks that generalize the convolution operation to data with graph structure. Together with GCN, we used the self-attention and attention mechanism at the beginning and after GCN layer to capture similarities between current pose and the historical motion sub-sequences

2 METHOD

2.1 Dataset

We started using AMASS dataset as it is. It consists of given sequences of 144 frames of pre-recorded human motion data, for training and 120 frames to test our inferred motion. Each motion frame is associated with a skeletal representation of the human. In our case number of joints is equal to 15. The body-model used is called SMPL and each joint angle is represented by a 3-by-3 matrix all relative to their parent. This means that the given rotation specifies how the parent bone must be rotated to obtain the orientation of its child. Since each joint is associated with 3*3=9 parameters and we have 15 joints, we look to find 9*15 = 135 values.

2.2 Model details

After initializing hyperparameters (see below) and parameters, similarly to Mao et al.[6] we computed DCT matrices transformations to be used on our sequence after attention mechanism. The idea is to directly encode the temporal nature of human motion in our representation and work in trajectory space. For each forward pass we get the batch of poses then reshape it to obtain key and queries to be used on attention before feeding them to GCN. The shapes must be such that we have keys of dimension known sequence length - prediction sequence length and queries of dimension equal to the length of the sequence of past frames to look to predict the immediate future. The Values are the subsequence of original poses reshaped to match the size of the multiplied matrix Key*Queries. To vectorize our sequence we used a sequence of 2 convolutions activated with RELU or LeakyRELU, with or without batch normalization after each sequential layer (values: eps=1e-05, momentum=0.1). We used this convolutions to obtain vectorized sequences of 256 values.

After applying convolution and normalization, other results investigated the removal of normalization because by shifting the mean of our features towards zero, we ignored the negative ones by using RELU as activation function. Therefore, we decided to change the activation function to Tanh in order to keep the batch normalization and exploit its regularization effect.

Vectorized Queries, key and Values sequences were used to calculate attention scores and weights.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

The idea is to exploit more specific part of the sequence of motion that are useful to predict future movements, computing attention weights.

Attention score, the GCN inputs, are then transformed with DCT matrix computed at the beginning, and finally fed to GCN layer.

Following the work of Mao et al. [6], this graph convolutional layer then takes as input a matrix product of input and output dimension of 2*(length of sequence to be predicted + length of sequence of past motion to look at : 24 + 40) x (number of pose parameters : 135). In details, this GCN architecture with skip connections uses: matrix A, which is the trainable weighted adjacency matrix for the layer p; matrix H from the previous layer p; and W, a trainable weights matrix of layer p.

To calculate the output of the layer p, we use the following equation:

$$H^{p+1} = \sigma(A^p H^p W^p) \tag{2}$$

Finally, we used Inverse Discrete Cosine Transform matrix calculated at the beginning to recover our predicted sequence.

2.3 Loss

As loss we used both MSE and as Mao et al. [6] average l1 loss without any particular improvements. We clipped the gradients of l2 norm to avoid exploding ∇, we used ADAM optimizer and learning rate decay.

$$l_a = \frac{1}{(N+T)*K} \sum_{n=1}^{N+T} \sum_{k=1}^{K} |\hat{x_{k,n}} - x_{k,n}| \tag{3}$$

Equation 3: *Average l1 loss*

3 EVALUATION

In this section we deal with ablation studies to our model we evaluated and compared. The evaluation metric used is the mean joint angle difference between all joints, summed over all 24 target time steps.

First approach was using Sequence 2 Sequence with LSTM cell. It resulted in pretty slow convergence rate (training time 9h on NVIDIA GeForce GTX 1080 Ti), however this is nothing unexpected form a recurrent architecture.

Table 1: Hyperparameters table

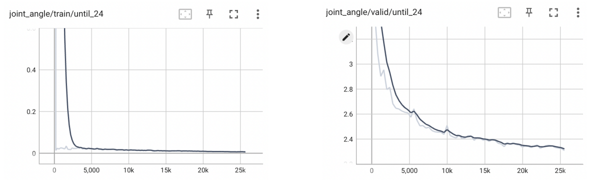
Hyperparameter	Value
Hidden features	256
GCN dropout	0.3
GCN number of stages	12
Learning rate decay rate	0.98
Learning rate decay step	330
Max norm for gradient clipping	1
Number of DCT dimension	64
Kernel size	40
Learning rate	0.0005
Number of epochs	1000
Batch size	128

Table 2: intermediate results table

Models	Score
LSTM	2.71859
GCN RELU no-norm lr 0.0005	1.88661
GCN RELU no-norm lr 0.0003	3.61631
GCN RELU norm	3.61646
GCN LeakyReLu norm	3.61649
GCN Tanh norm	3.61647

All results refer to the GCN+attention architecture.

The biggest boost in performance was achieved when switching to an attention model and GCN architecture. Convergence time of train error reduced from 8 hours for LSTM based architecture to around 2 hours and the score significantly improved (~x2). After carefully picking the most suitable learning rate, see table 1, we achieved a steady error decrease in both training set (fig: 2a) and test set (fig: 2b). These results match our expectations since the convergence is smooth and reaches the lowest value we obtained on the test set, presumably global minima. We tried decreasing the learning rate even further (second row of tab 2) but training got stuck in some kind of local minima resulting in worst generalization.



(a) Training error - lr 0.0005

(b) Test error - lr 0.0005

4 CONCLUSION

In conclusion, best results can be achieved with joint use of attention and GCN, that can deal with multimodal nature of human motion and unnatural artifacts that RNN based methods oftet produce. Future work can use adversarial training or generative modelling to obtain a more natural looking pose.

REFERENCES

[1] Emre Aksan, Manuel Kaufmann, Peng Cao, and Otmar Hilliges. 2021. A spatio-temporal transformer for 3d human motion prediction. In *2021 International Conference on 3D Vision (3DV)*. IEEE, 565–574.

[2] Emre Aksan, Manuel Kaufmann, and Otmar Hilliges. 2019. Structured prediction helps 3d human motion modelling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7144–7153.

[3] Ricson Cheng, Ziyan Wang, and Katerina Fragkiadaki. 2018. Geometry-Aware Recurrent Neural Networks for Active Visual Recognition. *CoRR* abs/1811.01292 (2018). arXiv:1811.01292 http://arxiv.org/abs/1811.01292

[4] Liang-Yan Gui, Yu-Xiong Wang, Xiaodan Liang, and José MF Moura. 2018. Adversarial geometry-aware human motion prediction. In *Proceedings of the european conference on computer vision (ECCV)*. 786–803.

[5] Wei Mao, Miaomiao Liu, and Mathieu Salzmann. 2020. History repeats itself: Human motion prediction via motion attention. In *European Conference on Computer Vision*. Springer, 474–489.

[6] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. 2019. Learning trajectory dependencies for human motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9489–9497.

[7] Julieta Martinez, Michael J Black, and Javier Romero. 2017. On human motion prediction using recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2891–2900.

[6] [7] [4] [2] [5] [1] [3]