Machine Perceptron Project 4 Report

Zijian Dong
zidong@student.ethz.ch

Hanxue Liang
haliang@student.ethz.ch

ABSTRACT

In this paper we describe our solution to 3D human motion prediction given an input 3D skeleton sequence based on a recurrent encoder-decoder framework. We introduce and analyze the structure of encoder-decoder framework and incorporate local geometric structure constraint–*framewise geodesic loss* as a more precise distance measurement rather than using Euclidean loss. To address the problem of prediction discontinuities, we use sampling-based loss and residual architecture. We also provide insights to the use of adversarial training mechanism. Our network model is built during the Machine Perception course and achieves decent prediction accuracy in the human motion prediction competition.

1 INTRODUCTION

Human motion prediction has great application potential in human-robot interaction and collaboration. Our focus in this paper is to learn models of human motion from motion capture data. More precisely, we forecast the most likely future 3D pose of a person given their past motion. Recently, a family of methods based on deep recurrent neural networks (RNNs) have shown good performance on this task. As the short term motion prediction can be regarded as a search for a function that maps an input sequence to an output sequence, we use sequence-to-sequence architecture, which is quite popular for solving this type of problem. However, most existing techniques not only suffer from first frame discontinuity problem, but also fail to produce plausible motion in the long term. The widespread using of Euclidean loss function also does not capture the geometric structure on each frame. To address the above problems, we incorporate local geometric structure constraint–framewise geodesic loss as a more precise distance measurement rather than using Euclidean loss. To address the problem of prediction discontinuities, we use sampling-based loss and residual architecture in our model. We also analyze the performance of using adversarial training mechanism in the model.

2 RELATED WORK

2.1 Deep RNNS for human motion prediction

With the application of deep learning approaches, especially Recurrent neural networks (RNN), the field of human motion prediction has experienced large progress in recent years. Fragkiadaki[3] proposes two architecture: a 3-layer long short-term memory(LSTM-3LR) network and an encoder-recurrent-decoder(ERD) model that use curriculum learning to jointly learn a representation of pose data and temporal dynamics. Martinez[6] uses a simple residual encoder-decoder and multi-action architecture by using one-hot vectors to incoporate the action class information. His incorporation of residual connection helps to model prior knowledge about the statistics of human motions and decrease the motion discontinuity between predicted and input sequences. They also use sampling based method to produce more plausible motion in the long term.

2.2 GANs

GANs have shown impressive performance in various generation tasks. Gui[4] designs AGED model where they incorporate adversarial training mechanism in human motion prediction. They regard as predictor the encoder-decoder motion prediction architecture and introduce tow global recurrent discriminator. One is unconditional fidelity discriminator and the other is conditional continuity discriminator. They also introduce a novel frame-wise geodesic loss as a geometrically meaningful distance measurement to regress the predicted sequences to ground truth sequences.

3 METHOD

In this section, we mainly discuss our network architecture and the motivation of designing this structure. The details of this network structure are shown in Figure 1

3.1 Sequence-to-sequence structure

Learning motion prediction, mapping the input sequences to output predicted sequences, can be solved by a sequence-to-sequence(seq2seq) structure based on an encoder-decoder network.[6]The encoder is used to learn the hidden representation from the input sequence and then the decoder takes the last motion frame and internal states to produce the output sequence.

Besides, since some parts in input sequences are more important than others in prediction, attention mechanisms are incorporated to make the network focus on more important parts[1]. We use the similar structure as [6]. Instead of using gated current unit(GRU)[2], the encoder and decoder are composed of LSTM. Furthermore, to eliminate over-fitting and improve the ability of generalization, dropout layers are added after LSTM layers. See Figure 1 for more details of the structure.

3.2 Sampling-based loss and residual architecture

To help the network recover from its own mistakes, the decoder is forced to utilize its own output samples as input to produce the following output sequences. According to [6], This method could produce plausible prediction in long-term motion prediction without depending on heavily hyper-parameter tuning.
Another large problem is frame discontinuity. When visualizing predicted motion prediction, a large gap between the first output frame and the last input ground truth can be observed. To deal with this issue, a residual connection[6] between input and the output of each RNN are used to model the motion angles instead of rotation angles. This method can be proven to generate more smooth predicted frames.

This is an abstract footnote

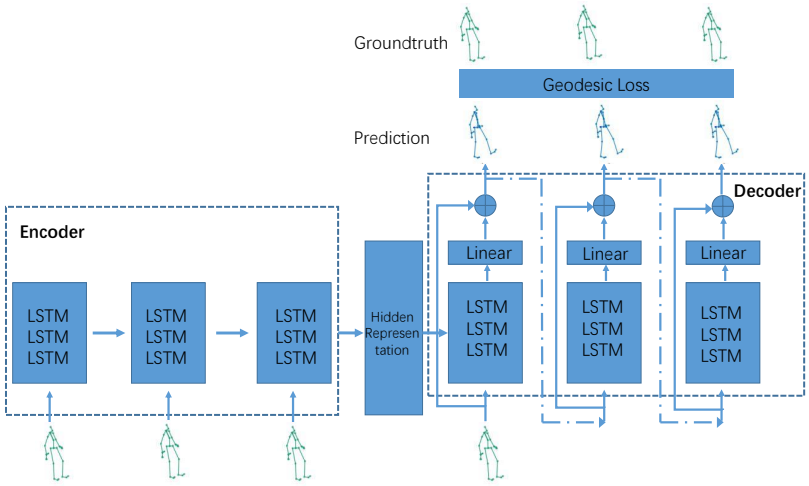

Figure 1: The training seq2seq network architecture.During training, the input sequence is fed into an encoder network and the decoder network could output its prediction. The encoder has a residual connection and could take samples of its output to predict following sequences. The Geodesic loss is calculated for backpropagation.

3.3 Geodesic loss

Given that the motion frame is presented as 3D rotations of all joint angles with respect to their parent bone, the aim of the loss function should capture the distance between two rotation. A traditional method to measure the distance between predicted frame and ground truth frame is the Euclidean distance[3]. However this type of distance fails to capture the geometric structure of 3D rotations[5][7]. In our project, we adopt another type of distance measurement between rotations and define the loss accordingly to regress our predictions to ground truths frame by frame.

Let k be a unit vector defining a rotation axis, and let v be any vector to rotate about k by angle θ (counterclockwise). Letting K denote the "cross-product matrix" for the unit vector k.

$$K = \begin{bmatrix} 0 & -k_z & k_y \\ k_z & 0 & -k_x \\ -k_y & k_x & 0 \end{bmatrix}$$

From Rodrigues' rotation formula, we could obtain the rotation matrix as follows:

$$R = I + (sin\theta)K + (1 - \cos\theta)K^2$$

The I is the 3×3 identity matrix, this matrix R is an element of the rotation group Special Orthogonal Group $SO(3)$ of orthogonal matrices with determinant 1. $SO(3)$ is a Lie Group with a Riemannian manifold structure. And K is an element of the Lie algebra $so(3)$ generating that Lie group.

We use geodesic distance to quantify the similarity between two rotations, which is the shorted path between them on the manifold. The geodesic distance is defined as follows: Given two rotation matrices $\tilde{R}$ and R, the rotation matrix of the difference angle between $\tilde{R}$ and R is $\tilde{R}R^T$. The angle can be calculated using the logarithm map in $SO(3)$ as

$$\log \tilde{R}R^T = A\frac{\arcsin(\|A\|_2)}{\|A\|_2}$$

where $A = (a_1, a_2, a_3)^T$ and is computed from

$$\frac{(\tilde{R}R^T - R\tilde{R}^T)}{2} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

The geodesic distance between $\tilde{R}$ and R is defined as

$$d_G(\tilde{R}, R) = \left\|\log(\tilde{R}R^T)\right\|_2$$

Based on the geodesic distance, we define the geodesic loss L_{geo} between predicted frame X and groundtruth X_{gt}. First, we convert each joint to their Euler format θ and $\tilde{\theta}$, calculate their corresponding rotation matrices R and $\tilde{R}$, then we would sum up the geodesic distance between the predicted frames and the groundtruth frames:

$$L_{geo} = \sum_{j=n+1}^{j=n+m} \sum_{k=1}^{k=K/3} d_G(\tilde{R}_j^k, R_j^k)$$

where $K/3$ is the number of joint, n is number of frames in seed motion and m is number of frames to predict.

4 EXPERIMENTS AND RESULTS

In order to test the prediction performance of different models, we try to make changes step by step and evaluate the mean joint angle difference on the validation dataset.

In all our experiments, we first use a single gated recurrent unit(GRU) with 512 cells optimized with SGD as the baseline. As seen from table 1,the model produces a mean joint angle difference of 12.7 on the validation dataset. Steady improvements are made according to the steps in table 1. Not all changes are recorded in this

table and we ignore the changes with negative impacts. As a result, we found that stacking recurrent layers and increasing LSTM cell numbers make the model slower to train, but make it achieve better prediction performance; We found that the best performance is achieved when choosing 3 LSTM layers with 1024 LSTM cells in each layer.

Furthermore, we use a learning rate of 0.0005 in our experiments to decrease the vibration of loss. The batch size is 64 and we choose the dropout rate as 0.5 and the attention length as 60. During training, we feed 120 input sequence to the encoder and predict 24 frames of motion from decoder. We implemented the experiment on NVIDIA 1080Ti in Leonhard Clusters.

5 DISCUSSIONS

5.1 Strategies for training Seq2seq architecture

When training seq2seq architecture, we find several techniques to accelerate training and increase training performance. First, Adam optimization method largely accelerates the convergence of the training model.

Table 1: Major changes made to our seq2seq structure and the resulting mean joint angle difference on public scoreboard

Changes	Mean Joint Angle Difference
GRU model with tuned parameters Epochs:50 Optimization method: SGD Cell number:512 Loss: mean squared loss	12.70
Change optimization method into Adam	5.15
Addition of sampling-based encoder residual architecture	4.55
LSTM model with orthogonal initialization Turned parameter for gates	4.12
Increase epochs to 200 Set learning rate weight decay	3.57
Increase into 2 LSTM layers	3.35
Increase into 3 LSTM layers	2.81
Addition of dropout layers attention mechanisms	2.66
Addition of geodesic loss	2.38
Increase cell number to 1024	2.34

In Table 1, Within 50 epochs, the model trained with Adam can achieve a mean joint angle difference of 5.15, but the model trained with SGD can only achieve the difference of 12.7.

Second, during training of LSTM layers, it is very important to choose a proper initialization method. For our task, when choosing orthogonal initialization method, the average angle difference can decrease from 4.62 to 4.17 with the same structure. It also requires less time to train. Third, overfitting and gradient explosion can sometimes happen when setting the number of layers into a large number. To deal with these issues, dropout layers and gradient clip are used. Another approach is to reduce the learning rate, but it may take more time to train the model.

5.2 Geodesic loss

For the design of loss, we have tried both the Euclidean loss and geodesic loss in the experiments, and the result is show in table 1. We could find that using geodesic loss help to improve the result from 2.66 to 2.34 Actually, using geodesic loss help us to obtain the best performance. So the design of geodesic loss does help to capture the geometric structure of 3D rotation and regress our predicted sequence to the groundtruth sequence more accurately. By contrast, simple use of Euclidean distance fail to fully capture the structural information between different 3D rotations.

5.3 Sampling-based loss and residual architecture

In our experiments, we also used Sampling-based loss and residual architecture. Sampling-based loss is that during training, decoder will produce a frame by taking as input its own samples from previous step. A residual architecture is adding a residual connection between the input and the output of each RNN cell. The application of sampling-based loss and residual architecture provides us a relatively better result, reducing the error from 5.15 to 4.55. By visualizing the predicted sequence and checking the error in long term, we noticed that the use of sampling based loss slightly help to produce plausible motion in the long run. As for residual connection, we note that residual connection does not help too much in improving the prediction accuracy. Perhaps it is because although residual connections can improve performance on very deep convolutional networks [14], while in our case it fails to capture the statistical distribution of 3D rotations from prior frames.

5.4 GANs

We also attempted to construct an adversarial training model to improve the performance. According to [4], we incorporate one fidelity discriminator and one continuity discriminator to our model and take our original model as a generator. The fidelity discriminator distinguishes whether the predicted sequences is human-like and the continuity discriminator examines the continuity between the generated sequences and the input sequences. However, this adversarial model achieves worse performance comparing to our original method. One possible reason for this poor performance is that the model is too complex to train,since the number of parameters is much greater than before.

6 CONCLUSION

In conclusion, we design and implement a seq2seq architecture based on encoder-decoder network composed of LSTM cells, which achieves better prediction accuracy and generalization for the task of motion prediction. To achieve lower mean joint angle difference of the prediction sequences, we incorporate sampling-based

loss, residual architecture and geodesic loss into our structure. In addition, dropout layers and attention mechanism are added to eliminate overfitting and make the model concentrate on more important parts.

We have several ideas and approaches on how to improve the performance of our model. One possible approach is to use the adversarial training strategy to ensure continuity and fidelity of the predicted sequences. Although we try to design these two discriminators and train them with our original model, we failed to obtain better performance due to the limited time. It is also very hard to make both the generator loss and discriminator loss converge. In the future, we hope to change the structure of the discriminators and choose a proper training strategy.We are also interested in representing poses as exponential map representations or quaternion representations.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[2] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* (2014).

[3] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. 2015. Recurrent network models for human dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*. 4346–4354.

[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] Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. 2016. Structural-RNN: Deep learning on spatio-temporal graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5308–5317.

[6] 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.

[7] Michael Mathieu, Camille Couprie, and Yann LeCun. 2015. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440* (2015).