Human Motion Prediction

Rohit Kaushik
Department of Computer Science,
University of Zurich, Switzerland
rokaushik@student.ethz.ch

Pratyush Singh
Department of Computer Science,
University of Zurich, Switzerland
psingh@student.ethz.ch

Melvin Ott
Department of Computer Science,
ETH Zurich, Switzerland
ottme@student.ethz.ch

ABSTRACT

Modelling human motion is a challenging task in computer vision and graphics since human body movement can change drastically from one environment to another. It's an important task for developing and deploying autonomous agents. In order to tackle this issue, we take inspiration from the Structured Prediction Layer (SPL) which decomposes the pose into individual joints and can be augmented with different neural network architectures. We introduce the SPL dropout layer and describe its effect on the prediction scores. Using the per joint loss instead of the standard mean squared error as well as a residual connection for modelling velocities help us stay afloat on the top of leaderboard in the Machine Perception course at ETH Zurich.

1 INTRODUCTION

Human motion prediction is an attractive field of research because of its applications in various fields like robot-locomotion, autonomous driving systems and bold decision making. Despite the progress in Deep neural network architectures, it remains a problem to predict the future sequences precisely since human motion is highly non-linear, dynamic and stochastic in nature.

With the development in Recurrent Neural Networks (RNNs), especially Long short-term memory networks (LSTMs), it is possible to model the human motion sequences more accurately and predict the short-term sequences with a certain limited uncertainty. Our model is built on the RNN layers with LSTM Block cell and augmented with a Structured Prediction Layer[2] which models the joint dependencies and decomposes the pose into individual joints. This is achieved via a hierarchy of small-sized neural networks that are connected analogously to the kinematic chains of the human skeleton. The proposed layer is agnostic to the underlying network and can be used in combination with most already existing architectures.

2 RELATED WORK

In the past, researchers have attempted to solve this problem using RNNs which significantly improved the accuracy of the predictions. Early work took great care in choosing a model expressing the interdependence between joints [5]. Martinez et al. [7] uses a simple residual encoder-decoder and multi-action architecture by using one-hot vectors to incorporate the action class information. Their incorporation of residual connection helps to model prior knowledge about the statistics of human motions. Ghosh et al. [3] propose a dropout autoencoder LSTM (DAE-LSTM) that combines a 3-layer long short-term memory (LSTM-3LR) with a dropout autoencoder to model temporal and spatial structures. Pavllo et al. [8] proposes QuaterNet which represents rotations with quaternions and their loss function performs forward kinematics on a skeleton to penalize absolute position errors instead of angle errors. Quaternion based parametrization improves short-term predictions. We base our work on the research by Aksan et al. [2] where the researchers introduce a structured prediction layer to model individual joints and their results are shown to achieve state-of-art performance in some of the human activity categories.

3 METHOD

In this section we discuss the network architecture, the motivation behind the architecture and our contribution to the architecture.

3.1 Dataset

We are given sequences of prerecorded human motion data (a subset of the Human3.6M Dataset [4]) and the task is to predict 24 frames in the future based on 120 previous frames (1 frame $\approx$ 17 milliseconds). The data is provided as TFRecords consisting of *file_id* and *poses* representing the human skeleton with 15 joints each having a 3x3 rotation matrix.

3.2 RNN with Structured Prediction Layer

We initially tried the sequence to sequence approach proposed by Marinez et al. [7], but we didn't achieve competitive prediction scores. The Structured Prediction Layer introduced by Aksan et al. [2] which can take the context embedding and produce a decomposed prediction for each of the joint, performed best on our dataset. Hence in this project we take inspiration from Structured Prediction Layer model and build on top of it by introducing new layers such as BLSTM and dropout within the SPL or joints predictions. Our code is based on the work done by Aksan et al. [2]. Table 1 describes our model in detail.

Layers
Input
Input Dropout
BLSTM
SPL-Dropout

Table 1: Layers used in our model

We pre-process the input by normalizing it to zero mean and unit standard deviation. Since this normalization modifies the output as well and it no longer remains a valid rotation matrix, we unnormalize the predictions before computing the loss.

Our major contribution to the model architecture is the use of BLSTM instead of LSTM and addition of Dropout between joint prediction in SPL layer which we call SPL-Dropout. We also perform extensive hyper parameter tuning and discuss the results in the later section, Ablation Study.

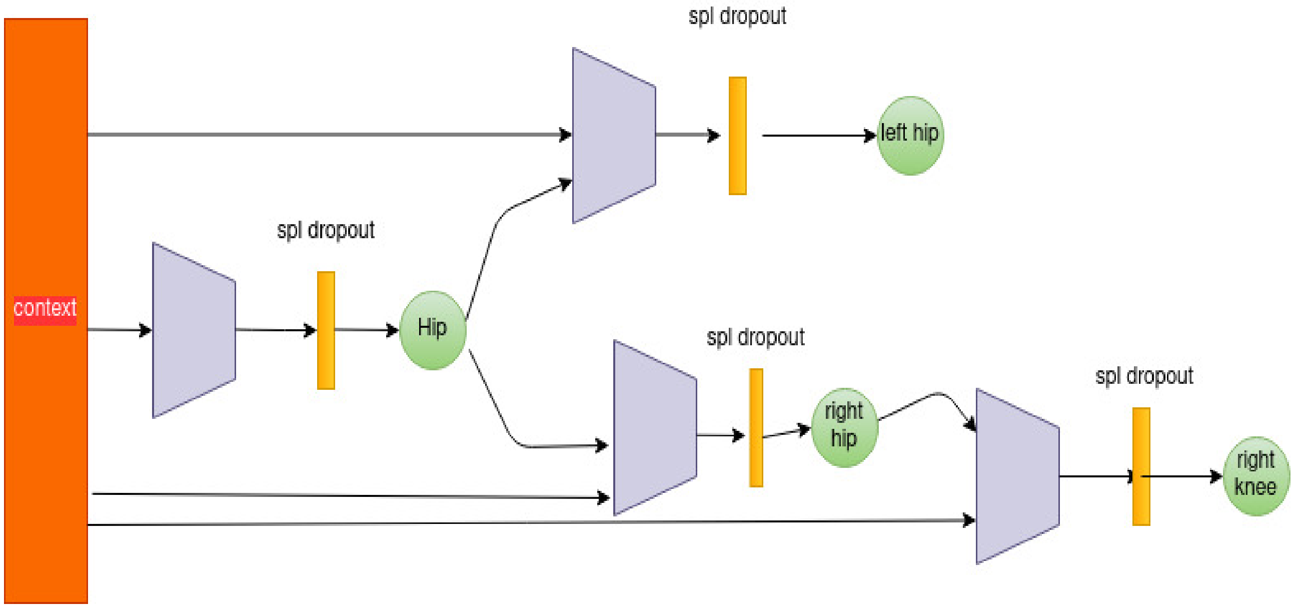

Figure 1: SPL Dropout Layer

We propose additional dropout between predictions of each joints that is extra dropouts between the structured layer predictions. The motivation behind this was that the validation loss significantly increased after 5-10 predicted frames. We call this layer as SPL-Dropout. We use a very small dropout of 0.01 between the join prediction.

3.3 Loss function

We started out with the standard mean squared error, but we noticed our prediction to be more accurate when we used the per joint loss introduced by Aksan et al. [2]. The main idea behind the loss is to evaluate the loss function (again mean squared error with our models) first at every joint and then sum up the results in a second step. The loss can be formulated as follows:

$$\mathcal{L}(X, \hat{X}) = \sum_{t=1}^{T} \sum_{k=1}^{K} \left\| \boldsymbol{x}_t^{(k)} - \hat{\boldsymbol{x}}_t^{(k)} \right\|_2^2 \tag{1}$$

where we denote $\hat{\boldsymbol{x}}_t^{(k)}$ as the predicted poses and $\boldsymbol{x}_t^{(k)}$ as the ground-truth.

4 EXPERIMENTS AND RESULTS

4.1 Evaluation Metric

We use the same evaluation metric as Aksan et al. [2]: the mean joint angle difference in the rotation matrix representation. Let $\hat{\boldsymbol{R}}$ be the predicted rotation matrix and $\boldsymbol{R}$ be the ground truth. We can then calculate $\tilde{\boldsymbol{R}}$ as: $\tilde{\boldsymbol{R}} = \hat{\boldsymbol{R}}\boldsymbol{R}^T$. The mean joint angle difference can then be formulated as follows:

$$L_{\text{angle}}(t) = \frac{1}{|\mathcal{X}_{\text{test}}|} \sum_{\boldsymbol{x}_t \in \mathcal{X}_{\text{test}}} \frac{1}{K} \sum_{k} \left\| \log\left(\tilde{\boldsymbol{R}}_t^{(k)}\right) \right\|_2 \tag{2}$$

where $\tilde{\boldsymbol{R}}_t^{(k)}$ is the rotation matrix of joint k at time t.

4.2 Training set-up

We implemented our models in TensorFlow [1]. We used the Adam [6] optimizer with a learning rate of 0.001. The learning rate decay rate was part of our ablation study, but a learning rate decay of 0.91 yielded the best scores in combination of our hyper parameters. We trained the models for 700 epochs, with an early stopping tolerance of 60 epochs and a batch size of 64. For training we used a Nvidia GTX 1070 (local training) and a GTX 1080TI (on the ETHZ leonhard cluster).

4.3 Results

Table 2 summarizes our modifications and their improvements. Our models are built upon the RNN-SPL architecture by Aksan et al. [2]. Using the basic RNN-SPL without any major modifications we achieved an online score of 5.30. The online score is calculated as the joint angle difference between all joints summed over all 24 target timesteps on a test set. Because normalization is standard practice in most machine learning tasks, we applied it to our problem and noticed a small improvement of our score. There are various LSTM Implementations in TensorFlow; instead of the standard LSTM we used the block LSTM and improved our score again For the models above we used the standard mean squared loss. Aksan et al. [2] propose a per joint loss, where the loss is first evaluated per joint and then summed up. This gave us a major prediction score boost and we were near the easy baseline.

For more accurate short-term predictions Martinez et al. [7] proposed to add a residual connection to model velocities, so we incorporated this idea into our model as well, which led to a major performance bump (2.26). As described in Section 3.2 we propose to add dropout to the SPL layer which further improved our score,

Model	Public Score (joint angle)
Basic RNN-SPL	5.30
normalization	4.83
block LSTM	4.82
per joint loss	3.68
residual velocity	2.26
LSTM cell size: 1024	2.25
spl dropout: 10%	2.11
LSTM cell size: 2048	**2.07**

Table 2: Scores of our models on the submission board (lower score equals better)

which made us pass the hard baseline. Our best submission uses the following hyper parameters: an input dropout rate of 4%, an input hidden layer size of 256, a cell size of 2048 for the block LSTM, a learning decay rate of 0.91 and of course a residual connection to model velocities. Note that we did not use SPL dropout for our best submission, but we passed the hard baseline with the model that uses SPL dropout.

5 CONCLUSIONS

We improved the RNN-SPL architecture by adding a dropout layer to SPL and by further hyper parameter tuning. Additionally, we performed an extensive ablation study to show each specific improvement of the architecture.

Although we achieved fairly good prediction scores, there is still massive room for improvement, especially when it comes to prediction far ahead in the future, as our predictions get worse as time progresses. To further improve prediction accuracy we could create networks for each human activity separately and tune them independently.

ACKNOWLEDGMENTS

We really appreciated the help of Mr. Manuel Kaufmann throughout the entire course of the project and we would like to thank Prof. Dr. Otmar Hilliges for providing us the opportunity to work on this challenging task.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. https://www.tensorflow.org/ Software available from tensorflow.org.

[2] Emre Aksan, Manuel Kaufmann, and Otmar Hilliges. 2019. Structured Prediction Helps 3D Human Motion Modelling. In *The IEEE International Conference on Computer Vision (ICCV)*. First two authors contributed equally.

[3] Partha Ghosh, Jie Song, Emre Aksan, and Otmar Hilliges. 2017. Learning Human Motion Models for Long-term Predictions. arXiv:cs.CV/1704.02827

[4] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. 2014. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2014).

[5] Ashesh Jain, Amir R. Zamir, Silvio Savarese, and Ashutosh Saxena. 2015. Structural-RNN: Deep Learning on Spatio-Temporal Graphs. arXiv:cs.CV/1511.05298

[6] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. arXiv:cs.LG/1412.6980

[7] Julieta Martinez, Michael J. Black, and Javier Romero. 2017. On human motion prediction using recurrent neural networks. arXiv:cs.CV/1705.02445

[8] Dario Pavllo, Christoph Feichtenhofer, Michael Auli, and David Grangier. 2019. Modeling Human Motion with Quaternion-Based Neural Networks. *International Journal of Computer Vision* 128, 4 (Oct 2019), 855–872. https://doi.org/10.1007/s11263-019-01245-6