Human Motion Prediction

Maria Rozou
rozoum@student.ethz.ch

Rahel Straessle
strrahel@student.ethz.ch

ABSTRACT

This paper provides a sample of a LaTeX document which conforms, somewhat loosely, to the formatting guidelines for ACM SIG Proceedings.

1 INTRODUCTION

Motion prediction is one of the tasks that is automatically done by humans but very hard to accomplish by machines. While even insects already have a physical model of the world, either learnt or predefined, we have to teach it to machines. Applications: human-computer interaction, motion synthesis, motion prediction for virtual and augmented reality. ... Using a single LSTM cell we build the most basic RNN network for human motion prediction and took this as our baseline for improvements.

2 RELATED WORK

Recent work focused on RNN based architectures to model human motion, with the goal of learning time-dependent representations that perform tasks such as short term motion prediction.

Julieta Martinez 2017: Trained encoder and decoder together with shared weights, single GRU. Used velocities through a residual architecture. Error reduction: fed predictions of the net back [3]

Fragkiadaki 2014: introduced an encoder-decoder (ERD) network, which is a type of recurrent neural network (RNN) model, that combines representation learning with learning temporal dynamics [2].

Sutskever 2014 (Sequence to sequence learning with neural networks): Deep LSTM's significantly outperform shallow LSTMs. They used therefore 4 layers. They also reversed the order of word inputs and would then perform better on long sentences. Stochastic gradient, 7.5 epochs, same length of input vector in batches: speed up of training. Best result with ensemble of LSTM that differ in their random initialization and in the random order or minibatches. [4]

Buetepage: [1]

3 METHODOLOGY

Input from assistant (Manuel Kaufmann):

- We won't be able to beat the hard baseline with a simple RNN cell (which I guessed after several runs with many different paramenters and I couldn't even beat your score)
- We should implement Seq2seq model
- Normalization/Standardization important
- One of the tasks of this exercise is to figure out if one-hot encoding of activity improves the performance

What I have done so far: I'm slow in python, sorry for that...

- Pushed your code to master
- Created new branch where I did all the changes.
- Wrote function in util.py to get mean, std, and dIlya Sutskever 2014:
  imensions where std is smaller than 10e-4
- Wrote function in util.py to standardize data
- If preprocess is on, we standardize input data and ignore the dimensions where std is almost zero, we also ignore those dimensions in the target. Test/eval still to do...
- One-hot encoding of activity labels. Has to be done after standardize data. Wrote the function to add one-hot vector. Need to get rid of those rows for validation error.
- Added model class for many layer LMTS with dropout option, but haven't tested model yet.
- Trying to figure out how to implement easy sequence to sequence model as another model class.

Additional comments:

- Our validation error goes down, but the performance on the test set does not improve at all. Why???
- Standardization of data to zero mean and variance of 1: if std is close to zero, what shall we do? Martinez ignored those dimensions for training so that is what I did as well... Where do we need to do the de-normalization of the data?
- Seq2seq model?
- Human motion is dynamic. I would love to use velocity and acceleration in the model. But this is probably too hard to do.

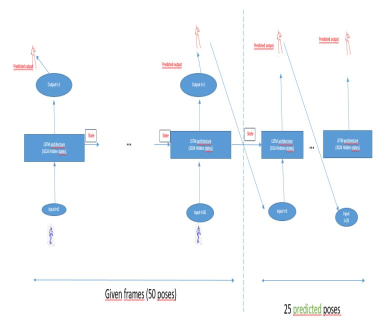
3.1 Preprocesssing and representation

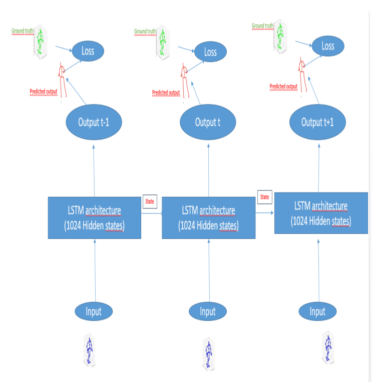
In the project of Human Motion, the architecture we used to solve the problem is based on RNN architecture using LSTM networks. We created an unrolled LSTM network that in each time step is being trained to predict the next pose. This can be seen also in training architecture, where the red poses are the predicted poses fromt he LSTM architecture and the greenones are the ground truths, meaning the real-true positions from the training data. The loss we used wa sthe L2 loss to minimize. We tried also some other distance functions, like the cosine loss, but this didnt help a lot and the vlaidation error was the same or a bit worse compared to the L2 norm, therefore we prefered the L2 loss. Also we used SGD, stochastic gradient descent and the ADAM optimizer to solve the optimization problem in the training process. We clipped also the norms to a factor of 5, so that we limit the magnitude of gradients and the Adam optimizer descends indeed to a minimum.

This is an abstract footnote

Table 1: Value of the Parameters

Parameter	Value
Learning rate	0.0015 fixed
Batch Size	10
Optimizer	Adam
Epochs	9
Max Length size	600
Clipping grads	5





3.2 Network structure

(1) Standardization
(2) One-hot encoding
(3) One to multilayer LSTM
(4) With /without dropout
(5) Seq2Seq model

4 EXPERIMENTS

5 CONCLUSION

ACKNOWLEDGMENTS

The authors would like to thank Prof. Dr. Otmar Hilliges for providing the theoretical background needed for this work and his assistants for the skeleton code that allowed us to focus on the most interesting aspects of the task.

The work is carried out in the frame of the lecture "Machine Perception" at ETH Zurich in the spring semester 2018.

REFERENCES

[1] Judith Bütepage, Michael J. Black, Danica Kragic, and Hedvig Kjellström. 2017. Deep representation learning for human motion prediction and classification. *CoRR* abs/1702.07486 (2017). arXiv:1702.07486 http://arxiv.org/abs/1702.07486

[2] Katerina Fragkiadaki, Sergey Levine, and Jitendra Malik. 2015. Recurrent Network Models for Kinematic Tracking. *CoRR* abs/1508.00271 (2015). http://arxiv.org/abs/1508.00271

[3] Julieta Martinez, Michael J. Black, and Javier Romero. 2017. On human motion prediction using recurrent neural networks. In *CVPR*.

[4] I. Sutskever, O. Vinyals, and Q. V. Le. [n. d.]. Sequence to Sequence Learning with Neural Networks. *ArXiv e-prints* ([n. d.]). arXiv:1409