IE643: Deep Learning : Theory and Practice **Jul-Nov 2019**

End-term Project Report : QuaterNet: A Quaternion-based Recurrent Model for Human Motion

Student Name: Swapnil Jayant Kumar *Roll No: 160100022*

Abstract

This report contains details of our project, which aims to address the prediction (short term) and generation (long term) of 3D human pose sequences. We take QuaterNet as a base architecture. QuaterNet is a recurrent network that models human motion by using Quaternions to represent rotations rather than Euler angles or exponential maps. Another improvement is in the error calculation approach. Errors are calculated using a new loss function which performs forward kinematics on a skeleton to penalize absolute position errors instead of angle errors. In this project, the main focus is to tweak the architecture of QuaterNet model in order to increase its computational efficiency and reduce training time, without significant loss in accuracy. The model architecutre was modified to one inspired by seq2seq models in used in NLP. Teacher forcing was used to increase accuracy. At the end we were able to reduce the model training time by 53.6%.

1 Introduction

Modeling human motion has been a highly sought after research topic in recent times. Different approaches such as Markov models, restricted Boltzmann machines, random forests, Convolutional networks and Recurrent Neural Networks (RNN) have been tried. Among them, neural-based methods have proven to be the most successful. We focus specifically on the problem of generating sequences of human poses on past pose data, using Recurrent networks. Human pose prediction has a wide range of applications in Autonomous Visual Surveillance, Autonomous Driving Vehicle, Human-Robot interactions, and entertainment industry. The prediction problem can be further classified into short and long-term prediction tasks. Human motion is a stochastic sequential process with a high-level of intrinsic uncertainty. This poses a huge challenge for predictions task, especially in the long term. For example, Short-term (prediction) tasks can be assessed quantitatively by comparing the prediction to a reference recording through a distance metric but, long-term (generation) tasks are harder to assess quantitatively due to the stochastic nature of human motion.

This project proposes two major improvements in the approach to enhance the performance and robustness of existing state-of-art models. First, we propose representing rotations using quaternions in our model. Other parameterizations such as Euler angles and exponential maps suffer from discontinuities and singularities (Gimbal lock), which can lead to exploding gradients and difficulty in training the model. As quaternions don't exhibit the above issues, a quaternion based parameterization helps resolve them. Second, a new differentiable loss function is used which conducts forward kinematics on a parameterized skeleton and evaluates position based loss. This approach combines the ease of working with angles to represent motions in the network and a better position-based loss, which is not prone to error accumulation along the kinematic chain. Human3.6m dataset has been used to test, evaluate and compare our model.

We provide a survey of existing literature in Section 2. Our proposal for the project is described in Section 3. We give details on experiments in Section 5. A description of future work is given in Section 6. We conclude with a short summary and pointers to forthcoming work in Section 7.

2 Related Work

2.1 Cited Papers

Our project draws inspiration from the work of Julieta Martinez [2]. In their paper [2], the authors proposed a simple GRU (Gated Recurrent Unit) based Recurrent network for the prediction of human poses. This model is much simpler and robust than the existing ones such as LSTM-3LR [6] (3 layers of Long Short-Term Memory cells), ERD (Encoder-Recurrent-Decoder) and SRNNs (structural RNNs). Figure 1 shows the comparison of LSTM-3LR and Julieta Martinez's architecture (RED- Recurrent Encoder-Decoder). RED is a sequence-to-sequence architecture. During training, the ground truth is fed to an encoder network, and then the error is computed on output of a decoder network that takes the previous prediction and the hidden state as input.

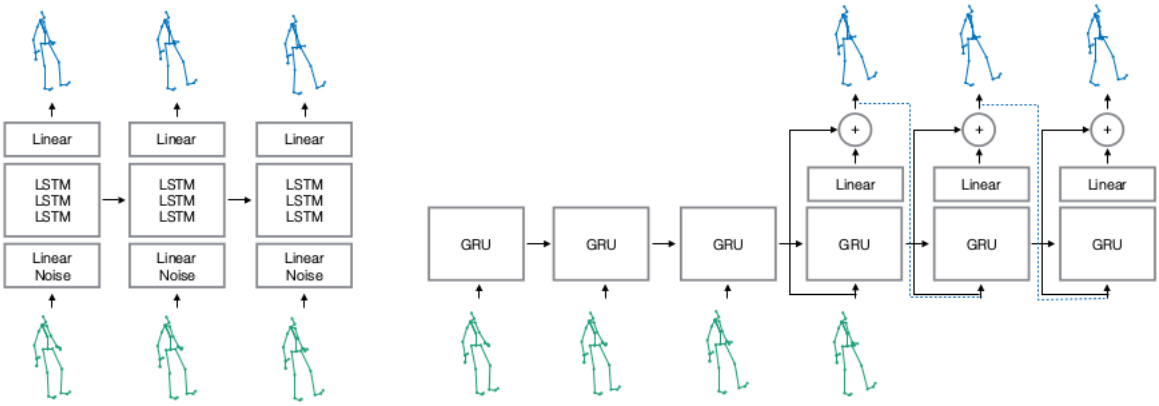

Figure 1: Left: LSTM-3LR architecture, introduced by Fragkiadaki et al [6]. Right: Julieta Martinez's [2] sequence-to-sequence architecture - RED.

Most of the earlier models suffer from the problem of first frame discontinuity in the predictions. RED [2] tackles this issue by predicting (modeling) velocities instead of actual angles of joints. This is done by adding a residual connection in the decoder, which effectively forces the RNN to internally model angle velocities. To make the model robust, other approaches add noise to the ground-truth value before training. But to work properly, they rely heavily on the complex hyper-parameter tuning. RED [2] achieves this in a much more simpler way by using an autoregressive structure (decoder's output in the previous time-step if fed as an input for next prediction). RED [2] is much simpler to train due to its shallow architecture.

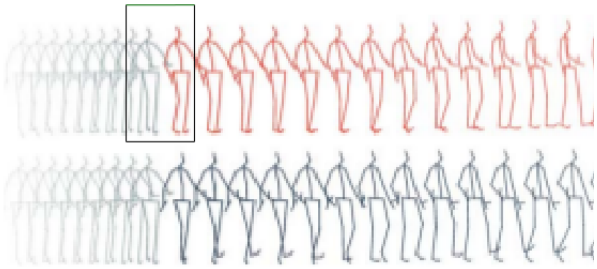

Figure 2: Sequence 1: Prediction by LSTM-3LR [6]. Sequence 2: Prediction by RED [2]. First frame discontinuity depicted by the box, this problem isn't there in RED's [2] predictions

Our project also draws inspiration from Chase J. Gaudet's paper on Deep Quaternion Networks [3]. In this paper, the author explores the advantages of a Quaternion-based CNN over a conventional real CNN. He also touches upon the topics of Differentiability, Gradient-decent, Weight initialization, Batch-Normalization, when inputs and weights are represented using quaternions. Catalin Ionescu's paper [4] gives a brief description of the Human 3.6M dataset that is going to be used in this project. The author touches upon the topics of data-capturing setup, the structure of the dataset, other details about the number of activities, actors, backgrounds. Yu Kong's paper [5] gives an extensive survey of work done on Human Action Recognition and Prediction.

2.2 Base Paper

Our project extensively draws inspiration from a closely related work by Dario Pavllo [1]. In this paper the authors took useful insights from [2] [6] [3], they also incorporated two other modifications for improving the model's performance. First was the use of quaternions to represent joint rotations to avoid issues of discontinuities and Gimbal lock. Second, they proposed the use of a better position based loss function. Position based loss does not suffer from error accumulation along the kinematic chain. Using this approach the authors have addressed both the short and long term subproblems in human pose prediction. Figure 2 and Figure 3 shows the Recurrent Network used for short and long-term prediction respectively.

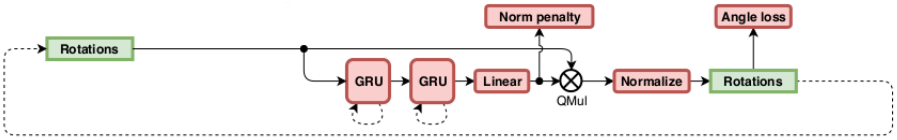

Figure 3: This figure shows the architecture used for short-term prediction [1]. "QMul" stands for quaternion multiplication, it forces the model to output velocities. If bypassed, the model emits absolute rotations.

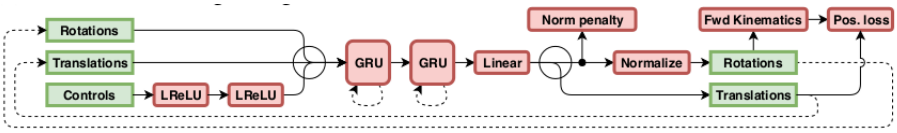

Figure 4: This figure shows the architecture used for long-term prediction [1]. The model includes additional inputs (referred to as Translations and Controls in the figure), which are used to provide artistic control.

The architecture for short-term prediction has a two-layer gated recurrent unit (GRU) structure. The two GRU layers comprise 1,000 hidden units each, and their initial states h0 are learned from the data. It is an autoregressive model, i.e. at each time step, the model takes as input the previous recurrent state as well as features describing the previous pose to predict the next pose. Similar to RED [2], the authors favor predicting relative rotation deltas (analogous to angular velocities) than absolute rotations. This is implemented in quaternions by using a quaternion product (QMul block in Figure 3). If the "QMul" block is bypassed the model will predict absolute rotations.

The architecture for long-term prediction is similar to that of the short-term with some additional inputs (referred to as Translations and Controls in figure 4), which are used to provide artistic control. Due to the stochastic nature of human motion, it becomes difficult to generate pose sequences for the long term. To address this challenge the authors propose a method known as curriculum schedule (progressively exposing the network to its predictions in previous time step).

The network takes as input the rotations of all joints (encoded as unit quaternions), plus optional inputs (in the long-term task), and is trained to predict the future states of the skeleton across k time steps, given n frames of initialization; k and n depend on the task. The authors have stated the results of experiments on datasets and comparison with those by existence models, which indicate the usefulness of the approach.

3 Methods and Approach

Previous works [2] [7], show that encoder-decoder seq2seq models are much easier to train as compared to a multi-layered GRU. Encoder-Decoder models are known to give a highly competitive performance. Hence in this project, we intend to implement the ideas of quaternion parameterization and position based loss, in an Encoder-Decoder type model. By this architecture we intend much easier, stable, quickly converging training and results comparable to Quaternet [1].

But, the Encoder-Decoder model tends to perform badly in long-term generation tasks, as also stated in [2]. Hence we intend to use the curriculum schedule as used by Quaternet [1] for the long term tasks. The following sub-sections give detailed information about our project.

3.1 Datasets

To facilitate the performance comparison between Quaternet [1] and our model, we will be using datasets the same as Quatenet [1]. Human 3.6M [4] for short-term predictions and custom dataset used by the Quaternet authors for long-term generation.

The Shorts Term dataset [10] consists of 7 subjects (namely S1, S5, S6, S6, S8, S9, S11). Each of these subjects performs 30 different actions. Hence the data has information of 210 activity sequences. Each of these activity sequences contain joint angles of 32 joint skeleton for 1383 time-stamps (frames). Hence overall the data contains 290430 instances of joint angles (210*1383). The data is stored in a hierarchical way in the model with the help of dictionary data structure.

3.2 Network Architectures

For short-term prediction task, we will use the encoder-decoder implementation of the Quaternet [1]. Encoder section will take the poses as input and encoder the data through time in the final hidden state. Then decoder takes the final hidden state of the encoder as its initial hidden state and compute the first sequence. For later predictions, the decoder takes hidden state and output of the previous time-step to compute the next pose.

For long-term generation task, we try to generate locomotion sequences from a given trajectory. Before generating the sequence of poses we need to determine some parameters along the trajectory like facing direction of the character, local speed, frequency of footsteps. These are learned by a separate network called pace-network (Recurrent network with one GRU layer). Similar to the Quaternet some extra-input and outputs (controls and Translation) are added for artistic control.

Figure 5 shows the network architecture to be used for short-term prediction. In the case of long-term prediction, the output will have a translation component along with the pose. Also, the input will have controls concatenated to the input pose. To implement curriculum schedule for the long-term generation task, the decoder will see the ground-truth value with a probability p. This probability p is initially kept 1, at the start of the training. With each epoch, the probability p decreases with a decay factor B.

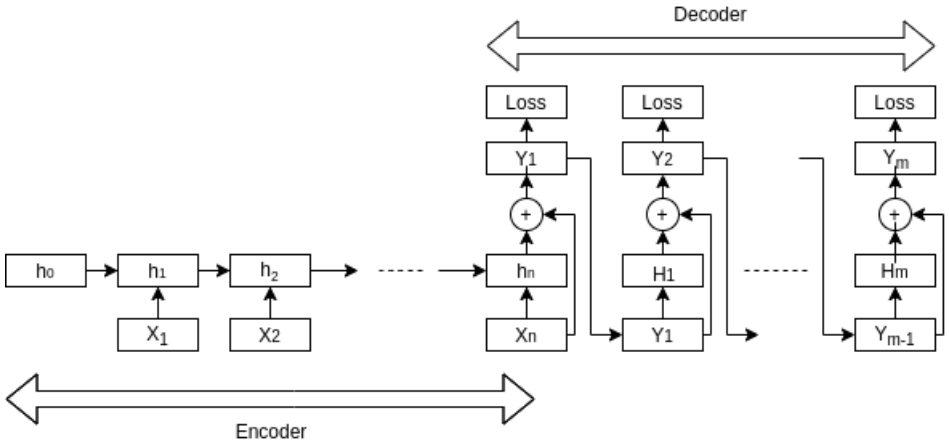

Figure 5: Propsed seq2seq Encoder-Decoder architecture.

3.3 Training the Network

For training the short-term network, we take 50 frames as prefix (input) and 10 frames as target (output). We take zero velocity as the baseline model for comparison. for long-term models, we have 30 frames as prefix and 60 frames as target. For efficient batching, we sample fixed-length episodes from the training set, sampling uniformly across valid starting points. We define an epoch to be a random sample of size equal to the number of sequences. We use an exponentially decaying learning rate with a decay-factor $=$ 0.999 per epoch. We also have extended the teacher-forcing (curriculum schedule) for the long-term generation to short-term model, we use the same approach as Quaternet [1].

We achieved a much easily converging and efficient training for the Encoder-Decoder network, it is primarily be trained on NVIDIA GEFORCE 940MX GPU. The loss of accuracy is negligible compared to the original architecture.

4 Experiments

4.1 Per Mid-sem

Extensive reading has been done for GRU (Gated Recurrent Unit) and Quaternions (with some basic operations). Problems of Euler angles representations such as non-uniqueness ($\alpha = \alpha + 2\pi n$), discontinuity, singularities have also been researched. The Quaternet project repository has been thoroughly read. Dataset representations, Skeleton implementation, Pace Network, Quaternet and Pose network implementation have been studied. Training and testing of the networks have been tried. It takes about 14 hours to train the Quaternet [1] network on NVIDIA GEFORCE 940MX GPU.

4.2 Post Mid-sem

Post-mid-sem, the focus was mostly on tweaking the underlining architecture of QauterNet. The model was trained on this tweaked architecture, results are shown is the result section. The training was done on the same device. The training was completed in about 6.5 hours with negligible loss in acccuracy.

5 Results

The following plots show training and validation loss comparisons for the 2 models.

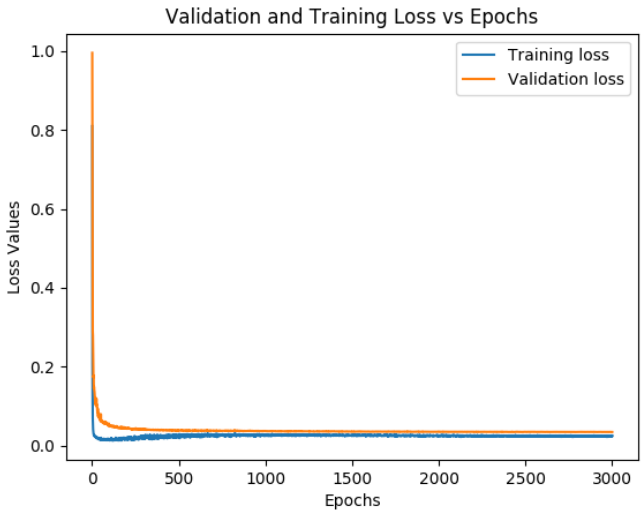

Figure 6: Training and Validation variation with number of epochs for original Quaternet.

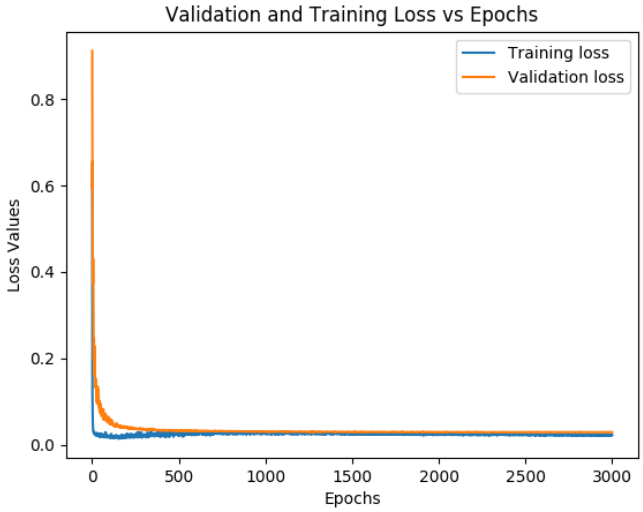

Figure 7: Training and Validation variation with number of epochs for our architecture.

It is clear from the above figures that, there is a negligible change in the loss function. Slight variations can be seen at the nose of the curves. But the second model was trained in 6.5 hours, whereas the original QuaterNet trains in about 14 hours on the mentioned device. The proof of this is given by the number of trainable parameters in both the models (figure 8).

.

QuaterNet :

```
swapnil@hp:~/Desktop/DeepLearningProject/QuaterNet$ python3.6 train_short_term.p
y
# parameters: 9526128
Training on 300 sequences, validating on 60 sequences.
Training for 3000 epochs
[1] loss: 0.80963 valid_loss 0.99525 lr 0.001000 tf_ratio 1.000000
```

Our Model :

```
swapnil@hp:~/Desktop/DeepLearningProject/QuaterNet2$ python3.6 train_short_
py
# parameters: 7038256
Training on 300 sequences, validating on 60 sequences.
Training for 3000 epochs
[1] loss: 0.64808 valid_loss 0.88342 lr 0.001000 tf_ratio 1.000000
[2] loss: 0.11843 valid_loss 0.58718 lr 0.000999 tf_ratio 0.995000
```

Figure 8: This figure shows screen-shot of linux terminal while training of the models.

It is clear from figure 8 that the number of parameters to be learned decreases from 9526128 to 7038256. This means about 26% reduction in parameters. Also, our architecture has a single hidden layer, it means computational efficiency increase in Gradient calculation using BPTT. Overall computation time reduced from 14 to 6.5 hours, 53.6% drop.

6 Future Work

Till now we have only focused on the short-term objective. In the future, we intend to extend the network for long-term generation tasks. Attention (more focus on key joints) can also be used to increase accuracy and efficiency. We wish to explore the possible application of this efficient version of QuaterNet in autonomous systems in cars. These models will give the automobile power to judge possible movements of pedestrians which can lead to accidents.

7 Conclusion

In this project, we introduced the problem of human pose prediction and it's application in various fields. We did a literature review of previous works in this problem, explaining their ideas, network architectures, and advantages. Quaternet (the primary reference for this project) was extensively reviewed and studied. In some initial experiments, it was observed that training the Quaternet [1] is computationally expensive. To tackle this problem a seq2seq encoder-decoder architecture of the Quaternet model was proposed. Training of this network was less expensive preserving the quality of results. In the future, we will be implementing this idea for the long term generation of human poses. We also intend to explore the possible use of Quaternet [1] on small scale less powerful computers with this new architecture.

References

[1] D. Pavllo, D. Grangier, and M. Auli. *Quaternet: A quaternion-based recurrent model for human motion.* British Machine Vision Conference, 2018.

[2] Julieta Martinez, Michael J. Black, and Javier Romero. *On human motion prediction using recurrent neural networks.* In Conference on Vision and Pattern Recognition (CVPR), 2017.

[3] Chase Gaudet and Anthony Maida. *Deep quaternion networks.* arXiv preprint, arXiv:1712.04604, 2017.

[4] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. *Human3.6M: Large scale datasets and predictive methods for 3d human sensing in natural environments.* TPAMI,36(7):13251339, July 2014.

[5] Y. Kong and Y. Fu. *Human action recognition and prediction: A survey.* arXiv:1806.11230, 2018.

[6] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik. *Recurrent network models for human dynamics.* In Conference on Vision and Pattern Recognition (CVPR), 2015.

[7] Hongsong Wang, Jiashi Feng. *VRED: A Position-Velocity Recurrent Encoder-Decoder for Human Motion Prediction.* arXiv:1906.06514, 2019

[8] Kratzer, Philipp et al. *Motion Prediction with Recurrent Neural Network Dynamical Models and Trajectory Optimization.* ArXiv abs/1906.12279 (2019)

[9] LSTM and GRU, Website Link

[10] Short-term Human3.6 Dataset, Download Link