CSCI 4962
Xiao Jin

HW 4

1 Declare

In this homework, the model is trained for several hours to with GPU to get a relative good results. I can not do it on Jupyter notebook. So I can only share the code and write the results into a pdf file.

2 Data and model structure

In this homework, we train the model on the Human3.6M dataset. The dataset has 5964 training samples and 1368 testing samples. Each sample is a 8 frame 224×224×3 video and the model need to do a regression task finally. We firstly cliped the data into the target frame for each data point and save them as numpy files. We choose the batch size as 2 and the model structure can be seen in figure 1:

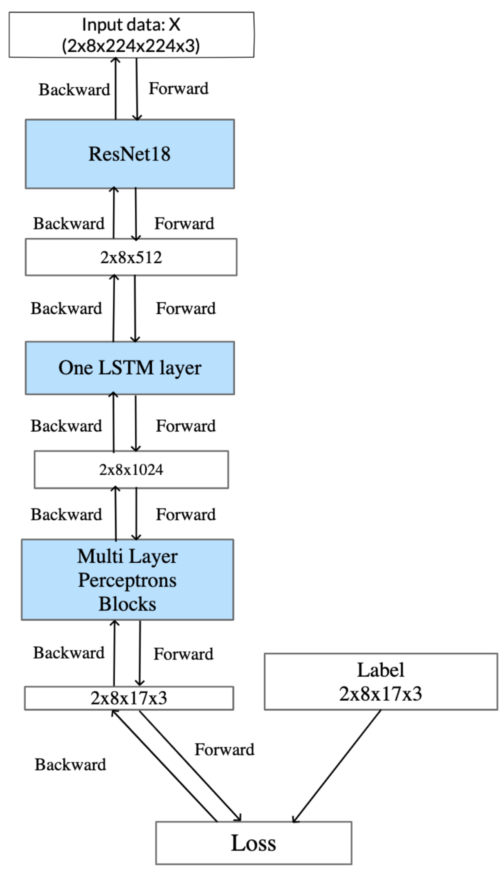


Figure 1: Neural Network Structure

3 Pseudo-code

Algorithm 1 Neural Network Training

Initialize Model parameter $\mathbf{\Theta}^0$, training data and labels $\mathbf{X}, \mathbf{Y}$, testing data and labels $\mathbf{X}_{test}, \mathbf{Y}_{test}$.
for $t = 0, 1, 2, \ldots, T$ **do**
    take batch data and labels $\mathbf{X}^t, \mathbf{Y}^t$
    compute loss: $\mathcal{L}(\mathbf{\Theta}^t; \mathbf{X}^t, \mathbf{Y}^t)$
    compute MPJPE and Loss
    compute gradient: $\nabla_{\mathbf{\Theta}} \mathcal{L}(\mathbf{\Theta}^t; \mathbf{X}^t, \mathbf{Y}^t)$
    optimize parameters : $\mathbf{\Theta}^{t+1} = \mathbf{optimizer}(\mathbf{\Theta}^t, \nabla_{\mathbf{\Theta}} \mathcal{L}(\mathbf{\Theta}^t; \mathbf{X}^t, \mathbf{Y}^t))$
    if An epoch of data have been trained **then**
        compute testing Loss $\mathcal{L}(\mathbf{\Theta}^t; \mathbf{X}_{test}, \mathbf{Y}_{test})$
        compute MPJPE and loss
    end if
end for

4 Loss function and MPJPE

We use the MSE as the loss function

$$\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{\|\hat{\mathbf{y}} - \mathbf{y}\|_F^2}{N \times 8 \times 17 \times 3} \tag{1}$$

where N is the batch size. We use the MPJPE as the function to evaluate the model performance.

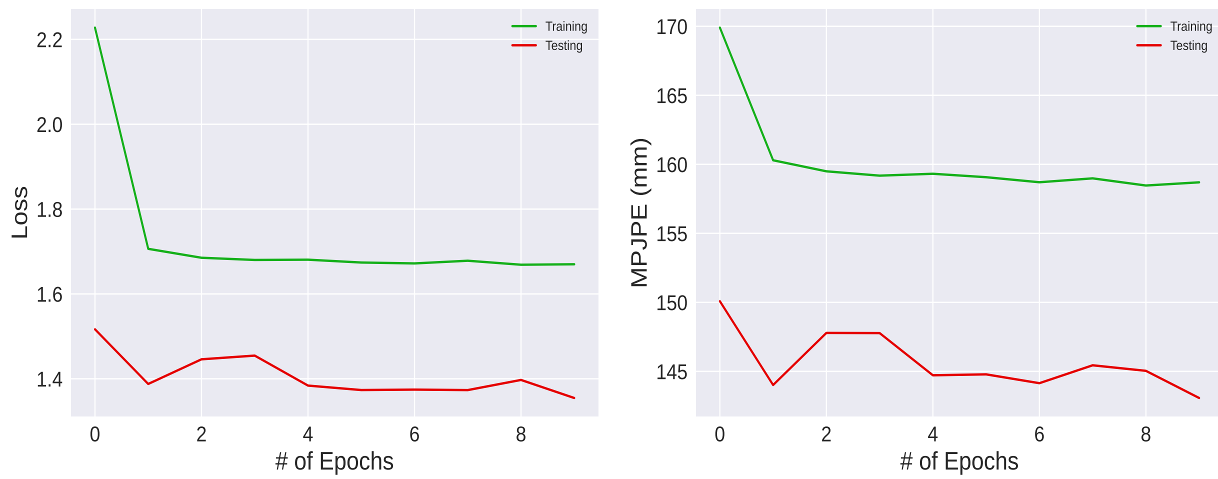
$$\text{MPJPE}(\hat{\mathbf{y}}, \mathbf{y}) = 1000 \times \frac{\sum_{n=1}^{N} \sum_{i=1}^{8} \sum_{j=1}^{17} \sqrt{\sum_{k=1}^{3} (\hat{\mathbf{y}}[n,i,j,k] - \mathbf{y}[n,i,j,k])^2}}{N \times 8 \times 17} (mm) \tag{2}$$

5 Environment setting

We choose the batch size as 2. The optimizer is Adam and we choose the learning rate (step size) as 0.001. We train the model on the RTX 2080 graphic card for 10 epochs. The MPJPE reaches near $150mm$ and the testing accuracy reaches near $150mm$.

6 Loss and MPJPE curves

The environment we implement our codes is tensorflow 2.0 with RTX 2080 Ti.
Figure 2a and 2b are loss curves and MPJPE curves respectively.



(a) loss curves

(b) MPJPE curves

Figure 2: Loss and MPJPE curves

7 Performance

Our model reaches the MPJPE less than 150% on testing data.

8 Brief Discussion

According to our simulation we found the following phenomenon in the convergence.

- The model always performs better on training data than testing data.
- The training loss continues decreasing under the Adam optimization method.
- From the curve, it can be inferred that over-fitting may not happen in our experiment.
- Since only one layer LSTM is used in the model, the performance keeps almost the same if we change it into a RNN layer