Protocol 23: HRNN

21.06.2019

Hierarchical Recurrent Neural Network (HRNN)

1. Introduction

Continued with the training of hierarchical recurrent neural network [1] (Figure 1) from last week (cf. protocol22), some adjustments have been made to improve the performance on MSR Action3D Dataset [2]. The main adjustments are the accumulation of the output across the time dimension from fully-connected layer, and the maximum likelihood loss function [3].

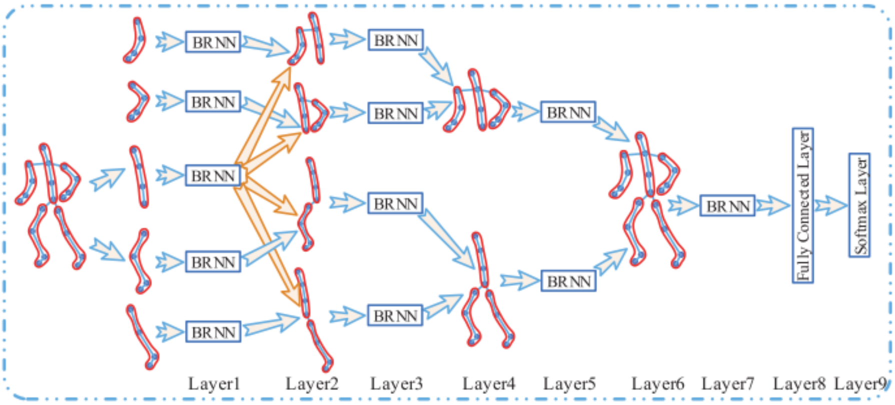

Figure 1: Hierarchical recurrent neural network. The human skeleton is divided into five parts, which are fed into five bidirectional recurrent neural networks (BRNNs) [1].

2. Methodology

The HRNN has 9 layers (Figure 1). The coordinates of the five body parts are respectively fed into the five RNN in the first layer. As the number of layers increases, the representations extracted from the subnets are hierarchically fused based on the correlation between the parts. All the recurrent layers are bidirectionally RNNs (BRNNs), and only the last recurrent layer (before fully connected layer) consists of LSTM neurons. The network outputs the predicted class from the softmax layer. Unlike most classification tasks, the maximum likelihood loss function [?] is used rather than the typical cross entropy loss:

$$\mathcal{L}(\Omega) = -\sum_{m=0}^{M-1} ln \sum_{k=0}^{C-1} \delta(k-r) p(C_k|\Omega_m)$$

where Ω: training set, C: classes, M: sequences, r: ground truth label. The paper [1] introduced weight noise to overcome the issue of overfitting.

3. Experiment 1: MSR Action3D Dataset [2]

- **Dataset:** MSR Action3D Dataset has 20 actions performed by 10 subjects, 567 samples with 22077 frames. All sequences are around 40 frames and were captured in 15 frames per second, and each frame in a sequence contains 20 3D skeleton joints, as shown in Figure 2.

  The dataset is divided into three action sets AS1, AS2 and AS3, each contains eight actions (classes). These three sets are further split into training and test sets based on the subject ID (odd number for training).

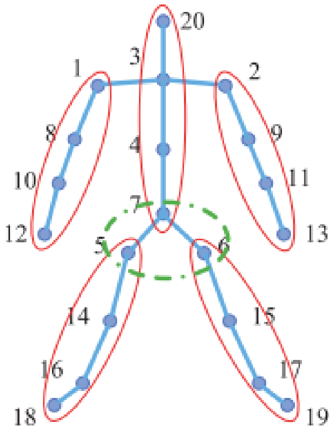

Figure 2: The 20 skeletal joints in MSR Action3D Dataset [1]. The joints are divided into five parts according to the paper.

- **Result:** The overall accuracy of the three action sets are 86%, 84%, and 89%, respectively.
- **Observations:**
  - Batch normalization does not seem to improve the result.
  - The number of sequences has an effect on the performance. Any number from 30 to 50 frames has better result than shorter or longer sequences.

3. Experiment 2: Lameness detection

- **Dataset:** The original dataset contains 501 samples, each of which contains the coordinates of 25 skeletal joints (Figure 4) from more than 100 video frames and a locomotion score as the label. As in the previous protocols, the locomotion scores fall into four classes. The dataset is augmented by mirroring and stretching, such that the amount of data is tripled.

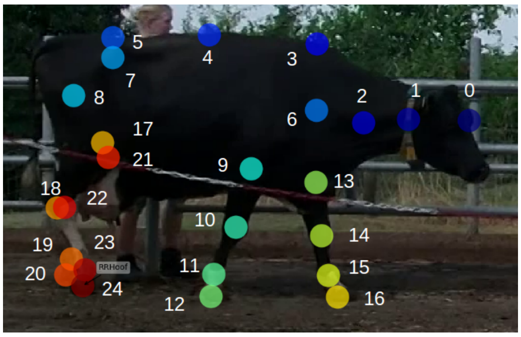

Figure 3: The 25 skeletal joints of a cow, and are divided into five parts.

- **Strategies:**
  - The value of learning rate was fixed in the beginning (0.001) and reduced for every ten epochs. The number of epochs is around 100.
  - Sequence length: The length was fixed as 40 frames for each video.
  - Weight noise was added during the training process.
  - Batch size: 16.
- **Result:** The overall accuracy of is 47.8%. The confusion matrix is displayed in Table 1, and the evaluation metrics for each class are listed in Table 2.

Table 1: Confusion matrix of lameness detection.

	Predicted Class 1	Predicted Class 2	Predicted Class 3	Predicted Class 4
Actual Class 1	104	37	2	4
Actual Class 2	14	37	44	12
Actual Class 3	4	15	9	6
Actual Class 4	4	6	18	2

Table 2: Precision, recall and f1-score

Class	Precision	Recall	F1-score
1	0.825	0.707	0.762
2	0.389	0.346	0.366
3	0.123	0.265	0.168
4	0.083	0.067	0.074

- **Observations:**
  - The network seems to underestimate the locomotion score for lame cows (Class 3 and 4).
  - The test accuracy began to drop as the training accuracy passed around 80%.

4. Discussion

With the maximum likelihood loss function, the hierarchical recurrent network is able to learn better than with cross entropy. In addition, the inclusion the output of fully-connected layer from all the time points seems to reveal more information than the last time point. For lameness detection, the prediction seems to be more difficult, as the data are noisy and look much more similar. The subtleties of skeletal joints can be degraded by poor data quality. Besides, some joints may vary from subject to subject, but tell nothing about the level of lameness. As for the training process, there are two obvious issues and will be dealt with in the next step:

- Class distribution

  The accuracy of lameness detection seems to be correlated with the imbalanced classes. This will be dealt with by oversampling/undersampling to make sure each class has almost the same amount.

- Overfitting

  The issue of overfitting is obvious, especially for lameness detection: the training accuracy can be over 90% while the test accuracy stagnates around 50%. In order to solve the issue, the number of parameters will be reduced. The manner of adding weight noise may be modified as well.

Additionally, another important step for the flollowing few weeks is transforming the task from classification to regression.

Reference

[1] Y. Du, W. Wang, and L. Wang, Hierarchical recurrent neural network for skeleton based action recognition, in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 11101118, 2015.

[2] W. Li, Z. Zhang, and Z. Liu, Action recognition based on a bag of 3d points, in 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, pp. 914, IEEE, 2010.

[3] A. Graves, Supervised sequence labelling with recurrent neural networks. 2012, URL http://books. google. com/books.