Ubicomp Digital 2020 - Convolution Layer with Recurrent neural network

Wei-Cheng Lai, Hendrik Schröter

July 15, 2020

Abstract

The challenge Ubicomp Digital 2020 from STABILO used the multivariate time series data. The data were collected from 100 volunteers, and contained 15 features in each sample. Multivariate time series classification was be used in this paper, the neural network was trained to classify 52 classes, including lower and upper cases of Arabic letters. The structure of neural network which we proposed in this paper is CNN-LSTM network. It combines convolutional neural network and also Long short-term memory layer (LSTM). We reached an accuracy of 68% on our writer exclusive test set.

1 Introduction

Handwriting is an important skill, and it's being used actually daily in our life [1]. The research indicated that contemporary handwriting has influence on the developing capabilities of children, atypical development [2, 3]. With the benefit of the digitization, handwriting is no more just off-line. Nowadays, there are many products, like tablets, which people can write on. They help us improve the life quality and make it easier to do meetings or tutorials online. Handwriting is a fine motor skills, for which various kinds of movement recording devices were employed [4]. Many methods for handwriting classification focuses on character recognition on pixel level. Therefore, the data was usually based on images [5]. In this paper, we recognize letters not based on images but based on sensor data recorded in a pen. This allows to write on a paper, while transmitting the digitized writings to a connected device such as a smartphone or computer. This technique can also eventually be used for auto-correction. Unlike traditional image data for handwriting recognition [6], the data for the challenge were collected from sensors, including two accelerometers, a gyroscope, a magnetometer and a force sensor. Therefore, we couldn't not use the way of image classification, but the way to solve the multivariate time series classification.

2 Data

The dataset provided by STABILO for the Ubicomp Digital 2020 challenge contains data from 100 writers. It was collected using a so called Digipen, a pen equipped with multiple sensors such as two 3-axis accelerometers (front and rear), a 3-axis gyroscope, a 3-axis magnetometer and force sensor sampled at 100 Hz. Each data sample needs a corresponding calibration to correct the bias and scale of each feature. The data includes upper case letters and lower case letters of the Arabic letters. Therefore, 52 classes are used for the classification
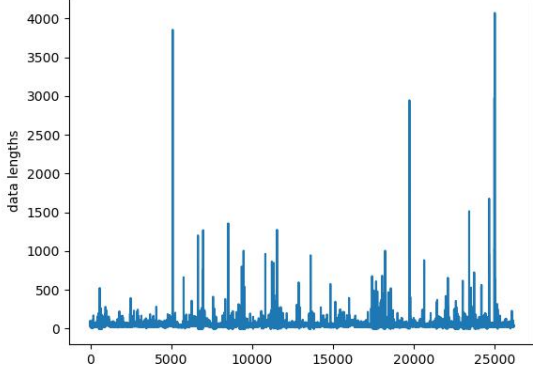
1

Figure 1: Histogram of raw data length

similar to other features, At the end, we didn't use the traditional zero-mean normalization, but applied a logarithm scaling. We found that logarithmic transformation of the feature space, which results in smaller value ranges being resolved more finely than large ones.

The logarithm formula:

$$S_{\log} = \text{sign}(S) \cdot \log(|S| + 1) , \qquad (1)$$

where S is the original feature vector.

3.2 Model of deep learning network

Based on the good performance of convolutional neural network on sentence-level classification tasks [7], we found that it would be helpful in letter classification task. Because of the information store in cells of long short-term memory (LSTM) [8, 9], we could also benefit from using it on time series data. Our final Model is based on a combination of convolutional neural network (CNN) and recurrent neural network (RNN), which was e.g. also used for language modeling [10]. There are 8 convolutional layers and a LSTM layer after the convolutional part in our proposed model. After the third convolutional layer, we did an exponential transformation, which transforms back to the normal value ranges. Between the convolution layers, we also applied the activation function (Rectified Linear Units), Batch normalization [11] and Max pooling layer. At the end, we took the values of hidden state of LSTM layer, which includes only the value of the last time state, and placed them into the fully connected layer, in order to predict the probabilities of 52 classes. In our proposed network, we used relative small learning rate $3 \cdot 10^{-5}$ and also weight decay $1 \cdot 10^{-3}$ for Adam optimizer [12]. Architecture of the model can be viewed at Table 1.

task. All 100 volunteers were right-handed. There was no guideline for the way of holding the pen. The subjects held the pen in their natural way. The data lengths from the raw data can be viewed at Figure 1.

3 Methods

3.1 Data processing

We separated the whole dataset with 80% for training set, and 20% for test set, which were separately 20979 samples and 5200 samples. The writers didn't overlap in training set and test set. This ensured that we preserve the writers' uniqueness in our dataset. Each sample had a different length. Before we entered the data into model, we needed to process them to the same length because of the input length of the convolutional neural network. We used the Fourier method to transcribe each signal to the same length. We also got rid of the features of 3-axis magnetometer, because we found that they don't have great influence on classification. The unit of 3-axis gyroscope features would also be transformed from angle to radian, in order to let the range of values become more
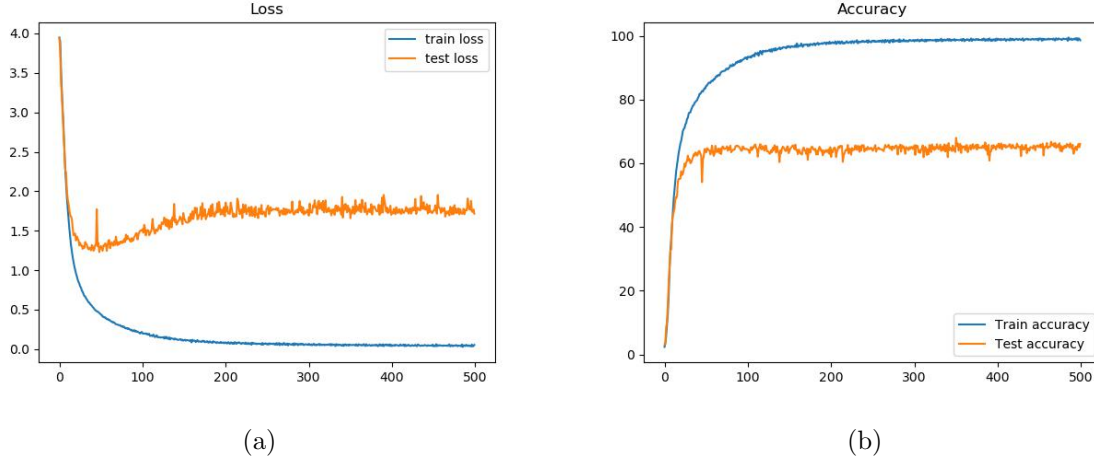
2

Figure 2: Overview of the result from the proposed model. (a) Cross Entropy Loss, overfitting after 100 epochs. (b) Accuracy, reaching 68% on the test set.

4 Results

We have trained on our CNN-LSTM-Net12 model for 500 epochs resulting in an accuracy of 68% on the test set. The Loss and Accuracy can be viewed at Figure 2a and 2b. Confusion Matrix can be viewed at Figure 3. We have also observed that the network has some difficulty to distinguish the lower and upper case of the same letters. This observation can be seen in Confusion Matrix. This problem especially holds for letters that are similar in the lower and upper cases. We believe that for a given word and sentence context, this is likely to improve, since upper case letters usually occur at the beginning of a sentence. In Figure 2, we can also see that the network is actually overfitting. The test accuracy though does not decrease significantly while proceeding with the training. Thus, for submission, we retrained the model using the whole dataset without any evaluation on a test set. The resulting train accuracy of 99.3% indicates that our proposed model could still model the whole training data distribution. Due to the overfitting characteristic our final submission is from

Layer Name	CNN-LSTM-Net12
conv1	1 x 11, 32, stride 2
conv2	1 x 11, 32, stride 1
conv3	1 x 11, 32, stride 1
exp	exponential
conv4	1 x 5, 64, stride 1
maxpool1	1 x 7, stride 3
conv5	1 x 5, 64, stride 1
maxpool2	1 x 7, stride 3
conv6	1 x 3, 128, stride 1
conv8	1 x 3, 128, stride 1
conv9	1 x 3, 128, stride 1
lstm1	hidden size 256, layer 1
fully connected	256 x 52 fully connection

Table 1: Model Architecture
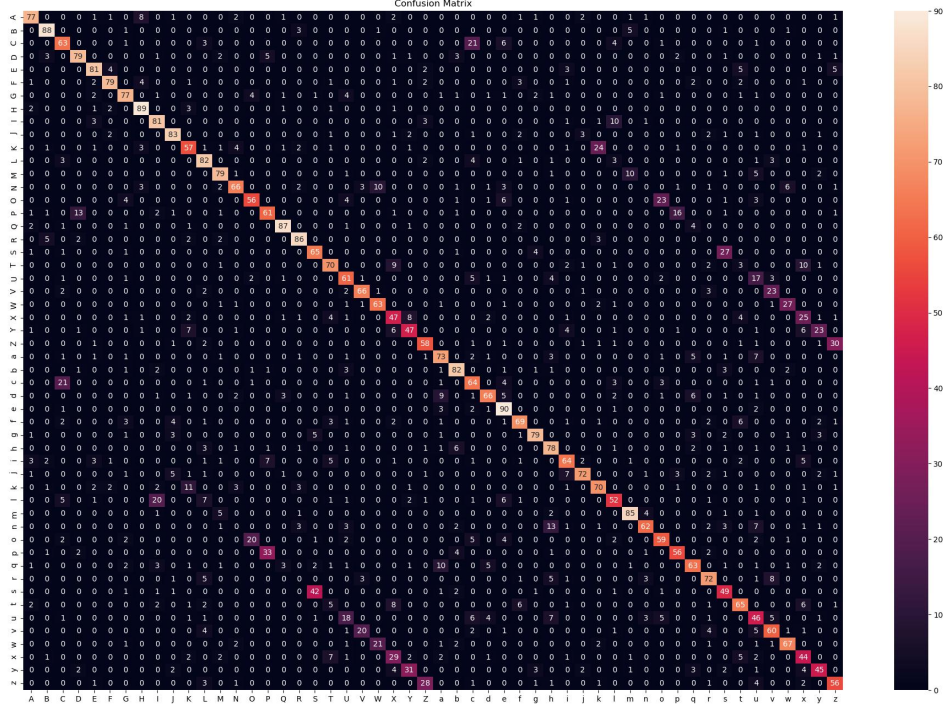
Figure 3: Confusion Matrix for the test set. Here, you can see the ambiguity sometimes between lower and upper case letters.

epoch 150.

5 Conclusion

Our proposed model could classify correctly most of the letters from the dataset. Nevertheless, if the upper and lower case letters are too similar, our model wouldn't be able to recognize the correct letter. For example, letters like c, C or o, O, it's sometimes even not easy for human beings to distinguish the difference, as human beings look normally the whole sentence but not just a single letter. This problem can actually be solved in the future, if we have more data and may eventually have the sentence-based dataset for stage 3. There were actually two improvements on our model during the training process. First, we changed the unit of gyroscope features from angles to radian. Seconds, we added a recurrent neural network (LSTM) after the convolutional neural network. Each of them made the improvement 3-6% for accuracy on test set. It was also interesting that Resnet18 [13] didn't work as well as the sequential convolutional neural network. It was overfitting and couldn't fit well into test set.

4

References

[1] Weintraub N Graham S. A review of handwriting research: Progress and prospects from 1980 to 1994. `https://doi.org/10.1007/BF01761831`, 1996.

[2] Abbott R. D. & Berninger V. W. Structural equation modeling of relationships among developmental skills and writing skills in primary- and intermediate-grade writers. Journal of Educational Psychology, `https://doi.org/10.1037/0022-0663.85.3.478`, 1993.

[3] Jean Alston. The handwriting of seven to nine year olds. `https://doi.org/10.1111/j.1467-8578.1985.tb00609.x`, 1985.

[4] Hans-Leo TEULING & Frans J. MAARSE. Digital recording and processing of handwriting movements. `https://doi.org/10.1016/0167-9457(84)90011-3`, 1984.

[5] N. Arica and F. T. Yarman-Vural. An overview of character recognition focused on off-line handwriting. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 31(2):216–233, 2001.

[6] J. J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994.

[7] Yoon Kim. Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882, 2014.

[8] Jürgen Schmidhuber Sepp Hochreiter. Long short-term memory. `https://www.bioinf.jku.at/publications/older/2604.pdf`, 1997.

[9] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent convolutional neural networks for text classification. In *AAAI*, 2015.

[10] Paul Mandal and Rakeshkumar Mahto. *Deep CNN-LSTM with Word Embeddings for News Headline Sarcasm Detection*, pages 495–498. 05 2019.

[11] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[12] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.