TIME SERIES CLASSIFICATION USING LSTM & CNN

Alokendu Mazumder

SPECTRUM Lab, Dept. of EE, IISc, Bangalore (PhD, 1^{st} year)

ABSTRACT

A hybrid neural network is proposed here to fetch features from a univariate time series using a stack of 1D CNN layers and LSTM simultaneously, which performs better classification than the baseline paper [1]. Another network with an attention [2] mechanism just followed by the LSTM is also proposed. Before feeding to dense layer for classification, the output feature vector of LSTM and CNN are merged by extracting maximum value from each dimension.

Index Terms— Time series, 1D CNN, LSTM

1. INTRODUCTION

Following are recent advancements in time series classification. Few researches to be highlighted are WEASEL [3] (Word ExtrAction for time Series cLassification) which is a feature based approach that extracts features to represent time series patterns in language. Multi scale CNN [4], fully connected CNN [5] are tailored for univariate time series classification also.

2. TECHNICAL DETAILS

In both proposed model, data is fed in parallel to stack of Conv. layers and LSTM.

2.1. 1D Convolution & LSTM

Three stacked Conv. blocks with filter sizes of 128, 256 and 128 followed by antirectifier [6] activation and batch normalization [7] after each are used here. Global average pooling [8] is applied after final Conv. block. Either LSTM or attention followed by LSTM is used with 0.8 dropout rate to extract features in parallel. Finally the output of 1D CNN block and LSTM are augmented by extracting the maximum value at each dimension from both.

2.2. Activation Function

Antirectifier activation is used here. Antirectifier over $x \in \mathrm{R}^d$ can be defined as:

$$Antirect(x) = concat[ReLU(x), ReLU(-x)] \in R^{2d} \quad (1)$$

Email: alokendum@iisc.ac.in, SR No: 04-03-00-10-12-21-1-20134

3. RESULTS

The model is tested over three UCR dataset [9] with 390, 381 and 600 training samples respectively. Each model is run for 2000 epochs with appropriate callbacks. Here, $\mathbf{P}_{lstm}$ & $\mathbf{P}_{alstm}$ are paper version of model, $\mathbf{M}_{lstm}$ & $\mathbf{M}_{alstm}$ are my models. In addition, the experiments of the paper were also performed over fine tuned variant of each model. My model outperformed some of them. CCE & MSE loss are used specific to each data set along with Nadam [10] and Adam [11] optimizer with variable step size.

Table 1. Performance comparison of proposed model with baseline paper

Dataset	$\mathbf{P}_{lstm}$	$\mathbf{M}_{lstm}$	$\mathbf{P}_{alstm}$	$\mathbf{M}_{alstm}$
Adiac	0.8593	**0.8865**	0.8670	**0.8696**
MedicalImages	0.8013	**0.8065**	0.7961	**0.8068**
MidPhxCorr	0.8217	**0.8378**	0.8400	0.7383

4. CONTRIBUTION

The codes are re-implemented by myself. ReLU [12] is used in paper, my model uses Antirectifier. In paper both feature vector output from LSTM and 1D CNN are just concatenated, while max-pooling along each dimension is performed here. I have also introduced MSE loss instead of CCE while training Adiac.

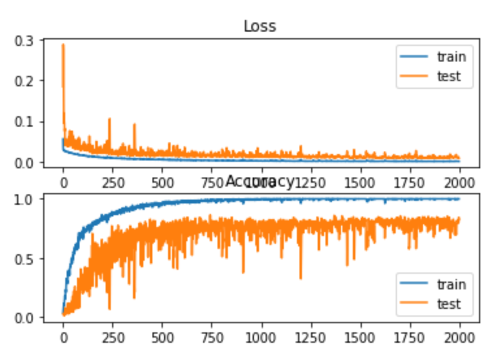

Fig. 1. Loss and Accuracy plot while training on Adiac

5. RESOURCES

The dataset used can be found **here**. This is the official github repository of the baseline paper **LSTM Fully Convolutional Networks for Time Series Classification**.

6. REFERENCES

[1] F. Karim, S. Majumdar, H. Darabi and S. Chen “LSTM Fully Convolutional Networks for Time Series Classification,” IEEE Access (Volume: 6), pages 1662 - 1669, 2017.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin ”Attention is all you need,” 31st Conference on Neural Information Processing Systems, Long Beach, CA, USA, 2017.

[3] P. Schäfer and U. Leser “Fast and accurate time series classification with WEASEL,” [Online] https://arxiv.org/ abs/1701.07681, 2017.

[4] Z. Cui, W. Chen, and Y. Chen “Multi-scale convolutional neural networks for time series classification,” [Online] https://arxiv.org/abs/1603.06995, 2016.

[5] Z. Wang, W. Yan, and T. Oates, “Time series classification from scratch with deep neural networks: A strong baseline,” in Proc. Int. Joint Conf. Neural Netw. (IJCNN), pp. 1578–1585, May 2017.

[6] F. Chollet ”Antirectifier: Keras example sript,” [Online] https://keras.io/examples/keras_recipes/antirectifier/, Jan 2016.

[7] S. Ioffe and C. Szegedy “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in Proc. Int. Conf. Mach. Learn, pages 448–456, 2015.

[8] M. Lin, Q. Chen, and S. Yan “Network in network,” [Online] https://arxiv.org/abs/1312.4400, 2013.

[9] Dau, Hoang Anh and Keogh, Eamonn and Kamgar, Kaveh and Yeh, Chin-Chia Michael and Zhu, Yan and Gharghabi, Shaghayegh and Ratanamahatana, Chotirat Ann and Yanping and Hu, Bing and Begum, Nurjahan and Bagnall, Anthony and Mueen, Abdullah and Batista, Gustavo, and Hexagon-ML ”The UCR Time Series Classification Archive,”, Oct 2018.

[10] T. Dozat ”Incorporating nestrov momentum into Adam,” Workshop Track - ICLR, 2016.

[11] D.P. Kingma and J. Ba ”Adam: A Method for Stochastic Optimization,” 3rd International Conference for Learning Representations, San Diego, 2015.

[12] K. Fukushima ”Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position,” Biological Cybernetics, 1980.