Time Series Classification using LSTM & CNN

Alokendu Mazumder
PhD (1st Year), SPECTRUM Lab, Dept. of EE
IISc Bangalore

Email: alokendum@iisc.ac.in

SR No: 04-03-00-10-12-21-1-20134

Background

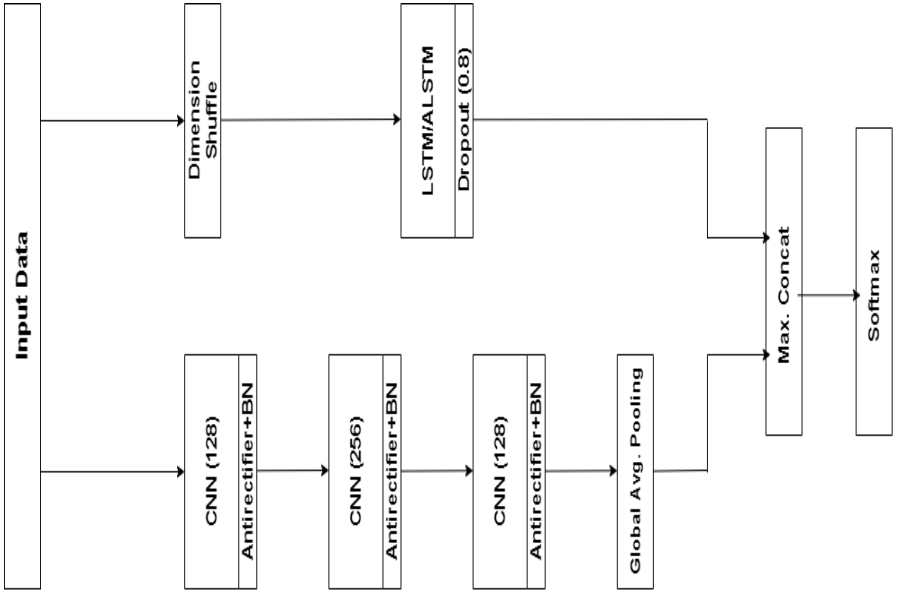
- ✓ 1D CNN’s and LSTM’s have proven to be remarkable in modelling time series data and performing classification tasks.
- ✓ In past years, several architectures were introduced for time series classification, like WEASEL, a feature based approach that represents time series patterns in language.
- ✓ Other models involve multi-scale CNN, fully connected CNN which are tailored to perform classification over univariate time series data.

Proposed Approach

- ✓ In this project, feature extraction of time series data takes place via a stack of 1D CNN layers and LSTM, in parallel.
- ✓ Same data is fed into 1D CNN layers and LSTM, feature vectors from their output is obtained and merged before feeding into dense layer for classification.
- ✓ Another model with minor variation of the one described above is also proposed where rather than using vanilla LSTM, an attention mechanism is applied after the LSTM module.

Technical Details

- ✓ **Three 1D CNN layers** are used with **128, 256 and 128 filter sizes** respectively followed by **antirectifier activation** and batch normalization after each.
- ✓ Global average pooling is applied after the final Conv. block.
- ✓ In parallel, **LSTM (or LSTM with attention)** with **128 cells** is used to extract features from the same data, followed **by high dropout rate of 0.8**. But, before feeding to LSTM, its is passed via a dimension shuffle layer.
- ✓ Both the feature vectors are then merged by **extracting maximum value** from each dimension of the two.
- ✓ Finally merged feature vectors are then passed to a dense layer, then via **Softmax for classification**.



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

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

Contributions (Novelty)

- ✓ In baseline paper, ReLU activation is used, here **antirectifier** activation is used, which unlike ReLU, dissent discards the negative values, hence reduces data loss but at the cost of doubling the dimension.
- ✓ For merging purpose, in paper simple concatenation is used. Here, extraction of **maximum value from each dimension** from both the feature vectors, LSTM and CNN stack is taking place. One can imagine it as a **1D maxpooling** along depth.
- ✓ For optimizers, I have used **Nadam** over most of dataset and Adam over one only. In paper Adam is used over all dataset.
- ✓ Looking at the class values, rather than using CCE loss, I tried MSE in one of the dataset and it outperformed baseline paper.

Results & Conclusion

- ✓ The model is tested over three UCR time series classification dataset with 390, 381 and 600 training samples respectively, each having 176, 99 and 80 sequence length respectively.
- ✓ Each model is targeted to learn over 2000 epochs but appropriate callbakcs were applied.
- ✓ For Adiac, MSE loss and Nadam is used. For Med.Images, CCE and Nadam is used and for MidPhxCorr, I have used CCE and Adam. Each model is trained with 32 batch size.

Table 1: Performance comparison my models with baseline paper models

Dataset	LSTM (Paper)	ALSTM (Paper)	Ft-LSTM (Paper)	Ft-ALSTM (Paper)	LSTM (My)	ALSTM (My)
Adiac	0.8593	0.8670	0.8849	0.8900	**0.8865**	**0.8696**
Med.Images	0.8013	0.7961	0.8066	0.7961	**0.8065**	**0.8068**
MidPhxCorr	0.8217	0.8400	0.8333	0.8433	**0.8378**	0.7838

Fig 1: Loss and Accuracy plot while training on Adiac