Improving the Learning Power of Artificial Intelligence Using Multimodal Deep Learning

Eugene Yu. Shchetinin[1], *Leonid* Sevastianov[2]

[1]Financial University under the Government of Russian Federation, Department of Mathematics, RU-125993, Moscow, Russia
[2]Russian University of Peoples Friendship, Department of Informatics, RU-117198, Moscow, Russia

Abstract. Computer paralinguistic analysis is widely used in security systems, biometric research, call centers and banks. Paralinguistic models estimate different physical properties of voice, such as pitch, intensity, formants and harmonics to classify emotions. The main goal is to find such features that would be robust to outliers and will retain variety of human voice properties at the same time. Moreover, the model used must be able to estimate features on a time scale for an effective analysis of voice variability. In this paper a paralinguistic model based on Bidirectional Long Short-Term Memory (BLSTM) neural network is described, which was trained for vocal-based emotion recognition. The main advantage of this network architecture is that each module of the network consists of several interconnected layers, providing the ability to recognize flexible long-term dependencies in data, which is important in context of vocal analysis. We explain the architecture of a bidirectional neural network model, its main advantages over regular neural networks and compare experimental results of BLSTM network with other models.

1 Introduction

Paralinguistics is a branch of linguistics, main scope of which is analyzing of non-verbal aspects of language – such as tempo, pitch, tone and intonation. The main goal is to analyze not what was said, but how it was said. The field of computational paralinguistics deals with analysis and synthesis of paralinguistic phenomena and research in this area has become very active in recent years [1,2].

Computer emotion classification aims to extract and classify human emotions from audio source. Different models, which estimate different physical properties of voice, such as pitch, volume and harmonics, are used. Such classifiers usually serve as parts of security systems, mobile assistants and biometric research algorithms [3,4].

The main difficulty is finding such features that would be robust to outliers and will retain variety of human voice properties at the same time [5]. Moreover, the model used must be able to estimate features on a time scale for an effective analysis of voice variability. Usually this is done by extracting features based on a sliding-window scheme, which solves the problem of data normalization and helps to reduce overfitting [6-9].

Another challenge is absence of uniform standard of human emotion types. One of possible solutions is MPEG4 standard, which divides human emotions in 6 groups: aggression, disgust, happiness, sadness and surprise. The last 6th group is called “neutral” and is used in case of absence of any emotions.

With development of computational power, neural networks have become widely used in field of speech recognition and emotional classification. In this paper a paralinguistic model based on Bidirectional Long Short-Term Memory (BLSTM) neural network is described, which was trained for vocal-based emotion recognition. The main advantage of this network architecture is that each module of the network consists of several interconnected layers: output layer, input layer and forget gate. Modules of BLSTM network are interconnected, so model can use context both from the past and from the future, thus providing the ability to recognize flexible long-term dependencies in data, which is important in context of vocal analysis. In the next paragraph we describe the architecture more thoroughly.

2 Bidirectional neural network architecture

Given an input sequence $\mathrm{X} = (\mathrm{x}_1, \dots, \mathrm{x}_\mathrm{T})$, a standard RNN computes a sequence of hidden vectors $\mathrm{h} = (\mathrm{h}_1, \dots, \mathrm{h}_\mathrm{T})$ and output vectors $\mathrm{y} = (\mathrm{y}_1, \dots, \mathrm{y}_\mathrm{T})$, by recursively evaluating the following equations from time steps $\mathrm{t} = 1$ to $\mathrm{t} = \mathrm{T}$:

$$\mathrm{h}_\mathrm{t} = \mathrm{f}_\mathrm{act}(\mathrm{W}_\mathrm{xh}\mathrm{x}_\mathrm{t} + \mathrm{W}_\mathrm{hh}\mathrm{h}_{\mathrm{t}-1} + \mathrm{b}_\mathrm{h})$$

$$\mathrm{y}_\mathrm{t} = \mathrm{W}_\mathrm{hy}\mathrm{h}_\mathrm{t} + \mathrm{b}_\mathrm{y}$$

Where W denote weight matrices, b – bias vectors and fact is the activation function of the hidden layer.

A typical LSTM layer consists of a number of recurrently connected memory blocks [10]. Memory block is a structure, consisting of a several memory cells, sharing the same input and output gate. The main goal of memory block is to store information within the network. Since each memory block has as many gate units, as a single memory cell, it’s more efficient to use block structure (fig.1).

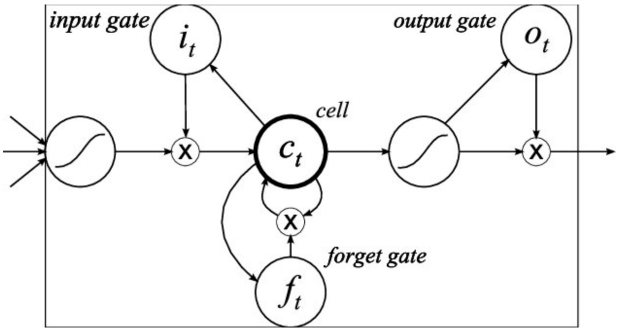

Fig. 1. Long Short-Term Memory Block

2.1 Memory Cells

Memory cell is a singular multiplicative unit of a LSTM network. Each memory cell is built around a central linear unit with a fixed self-connection, called nets. In addition to this, memory cell gets input from a multiplicative unit outj, called the output gate, and multiplicative unit inpj, called the input gate. Shortly, we have:

$$y^{out_j}(t) = f_{out_j}\left(net_{out_j}(t)\right);\ y^{in_j}(t) = f_{in_j}\left(net_{in_j}(t)\right);$$

$$net_{out_j}(t) = \sum_u w_{out_j u} y^u(t-1)$$

$$net_{in_j}(t) = \sum_u w_{in_j u} y^u(t-1)$$

$$net_{c_j}(t) = \sum_u w_{c_j u} y^u(t-1)$$

All these inputs convey useful information about the current state of the net. For example, an input gate may use inputs from other memory cells, to decide which information to store in its memory cells.

2.2 Input and output gates

Error signals within a memory cell cannot change, but they may get superimposed. The output gate has to learn which errors to trap, by appropriately scaling them. The input gate has to learn when to release errors, again by appropriately scaling them.

Distributed output representations usually require output gates, but not always both gates are necessary. Even in this case, however, output gates prevent the net’s attempts at storing long time lag memories from perturbing activations.

2.3 Forget gate

One of the main advantages of a LSTM model is ability to store data across the time lags, and carry error signals through time. However, in certain conditions this may contribute to a weakness. In a situation when an input is a continuous stream, the cell states may grow in unbounded fashion, causing saturation of output function and making the cell to degenerate into an ordinary BPTT unit. In order to overcome this problem LSTM network was modified with an addition of forget gate, which learns to reset memory blocks once their contents are useless.

A forget gate activation is calculated like any other gate activation function, and squashed with a sigmoid function:

$$y^{\varphi_j}(t) = f_{\varphi_j}(\sum_u w_{\varphi_j u} y^u(t-1))$$

Where y^{φ_j} functions as a weight of self-recurrent connections of the internal state $s_{c_j^n}$in the next equation:

$$s_{c_j^n} = y^{\varphi_j}(t) s_{c_j^n}(t-1) + y^{in_j}(t) g(net_{c_j^n}(t))$$

Where $s_{c_j^n}(0) = 0$. Bias weights for LSTM gates are initialized with negative values for input and output gates, and positive for forget gates. This way the initial forget gate activation is almost 1.0, which implies the cell will behave almost as a standard LSMT cell until it has learned to forget.

2.4 Vanishing gradient problem

If memory cell c_j inputs are mostly positive or mostly negative, then its internal state s_j will tend to drift away over time. This is potentially dangerous, for the $h_0(s_j)$ will then adopt very small values, and the gradient will vanish. One way to circumvent this problem is to choose an appropriate function h. But h(x) = x, for instance, has the disadvantage of unrestricted memory cell output range. A simple method might have been using ReLu as activation functions for the gates, but this approach leads to model diverge, due to the nature of LSTM gates [11].

A simple but effective way of solving drift problems at the beginning of learning is to initially bias the input gate in_j towards zero. Although there is a tradeoff between the magnitudes of$h_0(s_j)$ on the one hand and of y_{inj} andon the other, the potential negative effect of input gate bias is negligible compared to the one of the drifting effect [12].

3 Feature extraction and dataset

In order to perform any kind of machine learning, we first need to collect data and extract features from it. In voice-based emotion classification data used is recordings, presented in audio format. Features that can be extracted from audio source, can be classified into two big groups: low-level descriptors (LLD) and high-level descriptors (HLD). In broader terms, low-level descriptorsare those related to the signal itself and have little or no meaning to the end-user. In other words, and thinking in terms of the audio domain, these descriptors cannot be heard. On the other hand, high-level descriptors are meaningful and might be related to semantic or syntactic features of the sound. They will be used to classify sound objects into the class they belong.

Here are some of the commonly used LLD's, which we have also used in our experiment:

- Pitch -the perceived frequency of sound. The autocorrelation method for pitch estimation was used to determine changes in speaking behavior in response to factors relating to stress, intonation and emotional changes.
- Formants and their bandwidths.A formant is the spectral shaping that results from an acoustic resonance of the human vocal tract. They play a vital role in identifying human emotions. A 13th order linear prediction (LP) filter was calculated on each voiced frame of speech. Next, the first three formant frequencies and bandwidths were calculated from the roots of the polynomial LP predictor.
- Energy - the area under the squared magnitude of the considered signal.
- Jitter & Shimmer - Frequency perturbation also called jitter refers to the short-term (cycle to cycle) fluctuations in pitch (F0). It is obtained by measuring the fundamental frequency (pitch) of each cycle of vibration, subtracting it from the previous F0 values, and dividing it by the average F0. Shimmer on the other hand is calculated in similar fashion; however, the period to period variability of the signal peak to peak amplitude is calculated instead.

All of these features were extracted by using an open-source audio analysis Python toolkit Librosa [13]. For each of the feature, we computed a delta coefficient and 10 statistic functionals: mean, standard deviation, kurtosis, skewness, minimum and maximum value, relative position, and range, all in all resulting in 160 total features for each recording.

For our experiment we used AIBO dataset, which contains 18216 recordings, classified by following types of emotions: neutral, calm, happy, sad, angry, fearful, disgust and suprised. It's important to outline that there is a big class imbalance: neutral class contributes to more than 60% of the dataset, while anger and positive classes contribute to 8.1% and 4.8% accordingly. This dataset was made of children interactions with a pet robot called "Aibo". The children were led to believe that the Aibo was responding to their commands, whereas the robot was actually controlled by a human operator. Operator caused the Aibo to perform a fixed, predetermined sequence of actions,sometimes disobediently, thereby provoking emotional reactions. The data was collected from 51 children, and all recordings were cut into 1 second long parts.

4 Experiment results

The model itself was implemented with Keras deep-learning library, using Python programming language. The first model we trained was a BLSTM neural network, with a 160-100-50-5 topology: an input layer with size 160, two hidden layers with 100 and 50 nodes accordingly, and an output layer of size 5. For comparison, we have also trained a one-directional LSTM model, and a regular RNN model with same topologies. In order to cope with the class imbalance, we have tried and compared two different methods: SMOTE – synthetic oversampling and random neutral-class undersampling [14].

The results of computer experiments can be seen on table 1: It shows the values of the emotion classification accuracy metrics (average accuracy, F1-metric, and average AUC) obtained after applying the trained models to the test sampleOn the Fig.2 the graphs of ROC analysis functions and values of AUC indicators for all classes of emotions are given. Fig.3 shows the processess of training and tesitng the BLSTM network.

Table 1. Comparison of different neural network architectures

Models	Model performances		
	Type of class balancing	**Accuracy Train**	**Accuracy Test**
LSTM	No class balancing	46.25	36.3
	SMOTE	44.9	40.5
	Undersampling	44.55	42.2
BLSTM	No class balancing	54.4	50.2
	SMOTE	56.1	48.6
	Undersampling	56.6	52.3
RNN	No class balancing	43.7	41.0
	SMOTE	40.9	36.2
	Undersampling	42.4	39.8

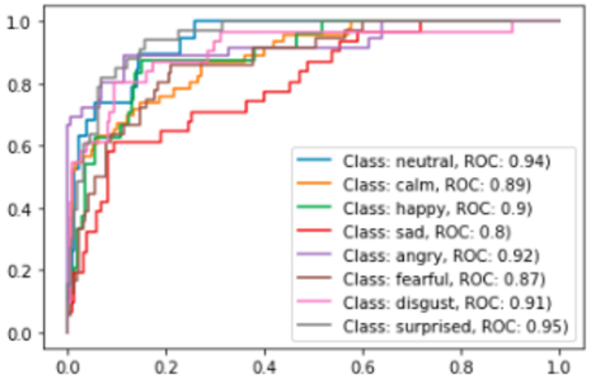

Fig. 2. ROC-AUC curves for emotion classes and their AUC.

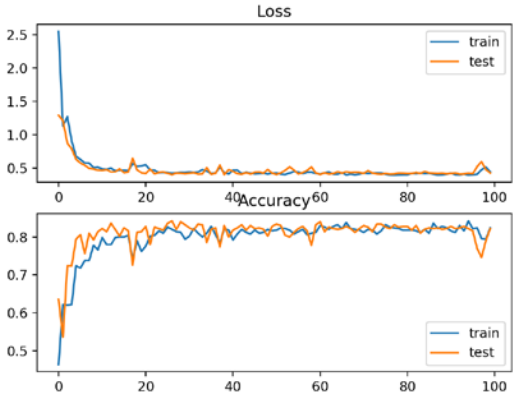

Fig. 3. Training graphs for the BLSTM neural network ensemble model. Upper graph: loss function-number of epochs; lower graph: accuracy-number of epochs

5 Conclusion

In this paper we presented a BLSTM neural network in application to voice-based emotion classification. The model shown much better results, compared to one-directional recurrent neural network, and slightly outperformed a one-directional LSTM network. This increase in accuracy was able by using memory blocks, which is a distinct feature of LSTM architecture. Further improvement of these results may lay in increasing the amount of data and features, used for training, and in stacking or blending different types of neural network architectures.

Based on the results of the research, the following conclusions are made. It is obvious that speech alone is not enough for high accuracy of emotion recognition, it is necessary to use additional data sources (videos, facial expressions, gestures, etc.). in many ways, the success of the algorithm depends on the quality of the training database. It should include all the main types of emotions displayed by experts, preferably in equal proportions. For this purpose, it is necessary to replenish and expand existing databases by creating new records, for example, using generative neural networks, as well as using transfer learning.

References

1. B. Schuller, IEEE Signal Processing Magazine, **29**(4), 97-101 (2012)
2. B. Schuller, S. Steidl, A. Batliner, A. Vinciarelli, F. Burkhardt, R. van Son, Computer Speech and Language, Special Issue on Next Generation Computational Paralinguistics, (2014)
3. B. Schuller, A. Batliner, *Computational Paralinguistics: Emotion, Affect and Personality in Speech and Language Processing*, (Wiley, 2013) DOI:10.1002/9781118706664
4. B. Schuller, F. Eyben, G. Rigoll, *Static and Dynamic Modelling for the Recognition of Non-Verbal Vocalisations in Conversational Speech*, Perception in Multimodal Dialogue Systems: 4th IEEE Tutorial and Research Workshop on Perception and Interactive Technologies for Speech-Based Systems, **5078** of Lecture Notes on Computer Science (LNCS), (Springer, Berlin/Heidelberg, 2008)
5. J. Wagner, F. Lingenfelser, E. Andre, Proc. of Interspeech, Lyon, France, **2013**, 168-172 (2013)
6. A. Janicki, Proc. of Interspeech, Lyon, France, **2013**, 153-157 (2013)
7. A. Stuhlsatz, C. Meyer, F. Eyben, T. Zielke, G. Meier, B. Schuller, Proc. of ICASSP, Prague, Czech Republic, (2011) DOI: 10.1109/ICASSP.2011.5947651
8. R. Brueckner,B. Schuller, Proc. of Interspeech, Portland, OR, USA, **2012**, 290-293 (2012)
9. R. Brueckner, B. Schuller, Proc. of ASRU, Olomouc, Czech Republic, (2013) DOI: 10.1109/ASRU.2013.6707757
10. S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, A Field Guide to Dynamical Recurrent Neural Networks, Kremer and Kolen, Eds. IEEE Press, **14**, 237-243 (2001)
11. F. Gers, N. Schraudolph, J. Schmidhuber, Journal of Machine Learning Research, **3**, 115-143 (2002)
12. M. Schuster, K. Paliwal, IEEE Transactions on Signal Processing, **45**(11), 2673 - 2681 (1997) DOI: 10.1109/78.650093
13. Librosa reference manual. https://www.Librosa.org,
14. L.A. Sevastianov, E.Yu. Shchetinin, Inform. Primen., **14**(1), 63–70 (2020) https://doi.org/10.14357/19922264200109