Emotion Recognition From Speech With Recurrent Neural Networks

Vladimir Chernykh
MIPT, IITP, Skoltech
Moscow
vladimir.chernykh@phystech.edu

Grigoriy Sterling
IITP
Moscow
sterling@phystech.edu

Pavel Prihodko
IITP
Moscow
prihodkop@gmail.com

Abstract

In this paper the task of emotion recognition from speech is considered. Proposed approach uses deep recurrent neural network trained on a sequence of acoustic features calculated over small speech intervals. At the same time special probabilistic-nature CTC loss function allows to consider long utterances containing both emotional and unemotional parts. The effectiveness of such an approach is shown in two ways. First one is the comparison with recent advances in this field. While second way implies measuring human performance on the same task, which also was done by authors.

1 Introduction

In the recent years human-computer interaction has become more and more interesting for data scientists. The goal of the most research is to make conversation between human and computer more native and natural. Obviously there are 2 necessary points to achieve this goal: make humans understand computer better and conversely. One can observe a great success in speech recognition (speech to text, STT). Nowadays machines can understand almost all human speech and the related services are used everywhere from Siri-like applications in smartphoes to voice control services. Totally, computer can understand *what* has been said. But this is not all the data in a voice, it also contains an information about *who* and *how* has spoken. The second question motivates us to learn how to recognize emotions from speech. Moreover, text to speech (TTS) services have the same challenging problem: machines can produce only intonationless speech.

1.1 Emotion recognition problem description

This paper is dedicated to emotional speech processing. The main goal is to learn how to classify speech by emotional state of a speaker. Unfortunately, there is no fairly strict definition what is emotion. Moreover, as it will be shown further, different people classify emotional speech differently. The second difficulty is about time properties of emotions. Often almost the whole utterance has no emotion (speaker is in neutral state), but emotionality is contained only in a few words or phonemes in an utterance.

Emotion recognition problem can be reformulated in mathematical terms as a classification task. In brief a function from the utterances space to the set of emotional states has to be constructed. In this space decision rule separates utterances with one emotion from the others. But what if people evaluates utterances differently? If we assume that the utterances emotional states are picked from an

unknown probability distribution then the model should predict and process these probabilities in the most sensible way.

1.2 Relation to prior works

At the moment the emotion recognition pipeline looks the following:

- Choose an emotional speech corpus
- Divide continuous audio signal into emotional utterances
- Calculate features for each utterance
- Select and train a classification model
- Develop a set of evaluation metrics and validate constructed model

There are a whole bunch of methods for each of the aforementioned steps. However, there is no panacea for a general problem setting of emotion recognition. One need to modify each step from the list in accordance with the specific properties of the task. The most effective approach takes in account lots of aspects like problem statement, goals, dataset structure, evaluation metrics, etc.

In this paper we consider utterance-level classification for speaker-free emotion recognition from natural speech. We decided to choose IEMOCAP database as a speech corpus because it satisfies all of properties of the task described above and there are several well-developed approaches on this emotional corpus to compare.

Before 2014 all research does not take into account time properties of the speech but uses only statistics aggregated over the utterances. The first attempt to consider a feature sequence was provided by Han et al. in [1] where they used DNN-based system with the following aggregation over time using statistical functions. They have shown some advantages of this approach, but accuracy was insufficient.

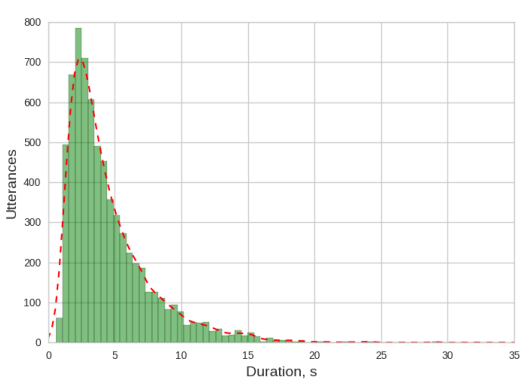
In the next year Lee and the same microsoft team [2] trained long short-term memory (LSTM) recurrent neural network with on a feature sequence and achieved about 60% accuracy on IEMOCAP database. They used a special loss function that enables to take into account the fact that emotionality in an utterance exists only in a few frames. Untill now it was the most effective workflow on IEMOCAP database. But the thing is that in their paper Lee et al. sieved the initial database in the way differs from ours. Firstly they considered another set of emotions and secondly they used only improvisation sessions (see 2 for details). Testing our approach under the same conditions we got almost the same quality — 56% mean class accuracy and 59% overall accuracy (see 5.3 for definitions). One more drawback of their work is the lack of code and the unclearness of the important details of the algorithm which makes it hard to reproduce.

2 Data description

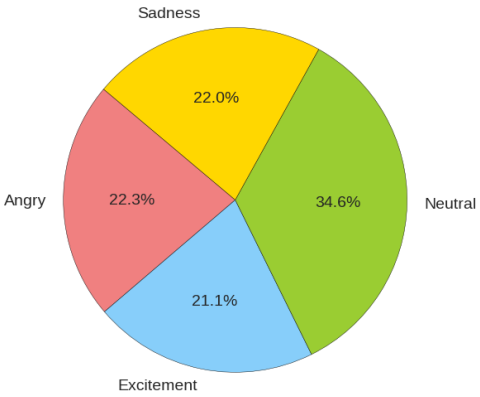
Regardless of model type, training procedure requires labelled emotional corpus. There are quite a few databases and its' overview can be found in [3, 4]. We carried out all experiments with an audio data from The Interactive Emotional Dyadic Motion Capture (IEMOCAP) database [5].

It consists of about 12 hours of audio-visual data from 10 actors. All recordings have a structure of dialogue between a man and a woman either scripted or improvised on the given topic. After collecting this audio-visual signals authors divided dialogues into small utterances of length mainly from 3 to 15 seconds (see fig. 1a for details) and then give them to the assessors to evaluate. Utterances are broken into two "streams" (man and woman voices) and thus they sometimes intersects (see fig. 2b). Then from 3 to 4 assessors were asked to evaluate each utterance based on both audio and video streams. The evaluation form contained 10 options (neutral, happiness, sadness, anger, surprise, fear, disgust frustration, excited, other). Here we take for analysis only 4 of them — anger, excitement, neutral and sadness. Figure 1b shows the distribution of considered emotions among the utterances.

Emotion was assigned to the utterance only if at least half of experts were consistent in their choice. And obviously it is not always like that. There is a 30% of utterances in which more than a half of experts gave different labels and emotion was not assigned at all. Moreover, from the remained utterances significantly less than a half have all experts consistently evaluate them (fig. 2a). This fact

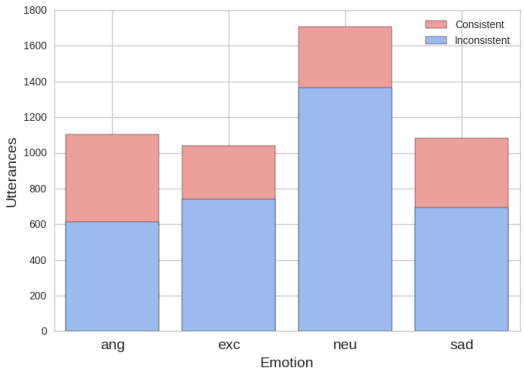


(a) Utterance duration distribution

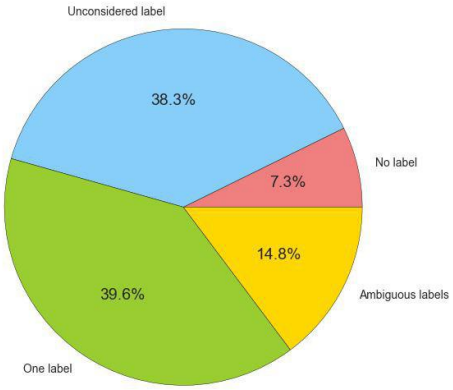


(b) Emotional labels distribution

Figure 1: Data overview



(a) Expert consistency



(b) Percentage of intersections

Figure 2: Markup details

illustrates that emotion is a subjective notion and there is no way to classify emotions precisely even if humans can't do that. In other words any model can not learn to recognize emotions, but can learn how experts label emotional utterances.

2.1 Technical details

Usually digital sound acquired as a converted analogous signal from a recording device. Audio wave is discretized and converted into time series with the given discretization frequency called sample rate. In IEMOCAP case, we need to have 16000 integers for one second of a sound with 16 kHz sample rate. Most popular approach to reducing this number is to separate an input signal into intersecting intervals (frames) of 0.1-1 seconds length and calculate acoustic features over it. This sequence of feature vectors represents an input sound in lower dimensional space with enough precision but anyway some information is lost. The other approach is to work with raw signal discretized with a lower frame rate. This approach is more accurate but needs significantly more computational power.

2.2 Features used

One of the sticking points in emotion recognition is what features should be used. Almost all possible features can be divided into 3 main classes:

- Acoustic features. They are calculated from the input signal and describe wave properties of a speech. Sometimes pitch-based features like chromagram powers are also included in acoustic set, but in some works they are considered separately or in prosodic group.

- Prosodic features. Generally, they try to measure peculiarities of speech like pauses between words, prosodies and loudness. Unfortunately, speech details such as prosodic features strongly depend on a speaker, and their use in speaker-free emotion recognition is debatable. In some works prosodic features improve the total efficiency, but we do not use it because the aforementioned reason.
- Linguistic features. The third kind of frequently used features based on semantic information from speech. We also do not use them because usually exact transcriptions are unavailable.

So in this paper we consider only acoustic features that measure wave properties of a signal. Along with Fourier frequencies and energy-based features Mel-frequency cepstral coefficients (MFCC) are usually used. Fourier transformation of the current time interval of the input signal can be considered as another signal and MFCC features measure its frequencies and powers.

Finally, 34 features are used. They include 12 MFCC, chromagram-based and spectrum properties like flux and roll-off. For all speech intervals we calculate features in 0.2 second window and moving it with 0.1 second step. For example, 4 seconds signal results in 78 vectors of dimension 34.

3 Problem statement

Emotion recognition task can be formulated as a multi-class classification problem in mathematical terms. This reformulation allows to apply wide range of well-established methods.

Let $\mathcal{D} = \{(\mathrm{X}, \mathbf{z})\}_{i=1}^{n}$ be the training set where $\mathbf{z}_i \in \mathcal{Z} = E^* = \{0 \dots k-1\}^*$ is the true sequence of labels and $\mathrm{X}_i \in \mathcal{X} = \left(\mathbb{R}^f\right)^*$ — corresponding multidimensional feature sequence. It's worth to mention that the lengths of these sequences $|\mathbf{z}_i| = U_i$ and $|\mathrm{X}_i| = T_i$ may not be the same in general case, the only constraint is the $U_i \leq T_i$ condition.

Also let's divide the dataset into train and validation sets $\mathcal{D} = \mathcal{D}_l \cup \mathcal{D}_v$ with sizes N_l and N_v correspondingly. $\mathcal{I}_l$ and $\mathcal{I}_v$ show the corresponding split of indexes.

Further we introduces the set of classification functions $\mathcal{F} = \{f : \mathcal{X} \mapsto \mathcal{Z}\}$ in which we want to find the best model. In case of neural network with the fixed architecture it is possible to associate the set of functions $\mathcal{F}$ with the network weights space $\mathcal{W}$ and thus function f and vector of weights $\mathbf{w}$ are interchangeable.

Next step it to introduce the loss function. As in almost all classification tasks here we are interested in accuracy error measure. But due to it's non-convexity we are unable to optimize it directly and thus we use the convex approximation which is denoted as $\mathcal{L}(\mathbf{z}, \mathbf{y})$ where $\mathbf{y}$ is the answer of the model.

Assume that there is an "perfect" classifier f^*. Our task is to find $f^*_{\mathcal{F}}$ that approximate f^* in the best possible way. It can be done using the risk minimization approach:

$$f^*_{\mathcal{F}} = \arg\min_{f \in \mathcal{F}} \mathop{\mathbb{E}}_{(\mathrm{X}, \mathbf{z})} \left[\mathcal{L}\left(\mathbf{z}, f(\mathrm{X})\right)\right].$$

But real data distribution in $\mathcal{X} \times \mathcal{Z}$ space is unknown and thus it is impossible to calculate the expected value properly. Therefore it is common to minimize an empirical risk:

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \frac{1}{N_l} \sum_{\mathcal{D}_l} \mathcal{L}\left(\mathbf{z}_i, f(\mathrm{X}_i)\right) = \arg\min_{\mathbf{w} \in \mathcal{W}} Q\left(\mathbf{w}, \mathcal{D}_l, \mathcal{L}\right).$$

Thereby we have $\hat{f}$ — approximation of $f^*_{\mathcal{F}}$ built based on the available data. Common approach here to reduce overfitting is to check the validation error from time to time during the optimization process and to stop when it starts growing.

4 Recognition methods

Throughout this paper we use a few recurrent neural network models for emotions detection in the human speech. The general structure stays more or less the same : input layer, few LSTM layers stacked over each other, dense classification layers and output softmax layer. One can address to the experiment section 5 for a more detailed architectures description. Below there is an overview of two main approaches to the network training that we used in this paper.

4.1 One-label approach

One-label approach implies that every utterance has only one emotional label notwithstanding it's length. It is the obvious approach regarding datasets labelled with the utterance annotation technique [5].

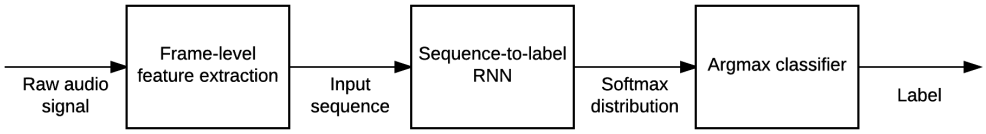


Figure 3: One-label approach pipeline

In details, let all the lengths of the label sequences are equal to 1, $U_i = |z_i| = 1$. Then vector $\mathbf{z}$ becomes scalar z

The recurrent neural network can be thought as a mapping from the input feature space $\mathcal{X}$ to the probability distribution over the emotional class labels parameterized with the model weights $\mathbf{y} = \mathrm{N}_{\mathbf{w}}(\mathrm{X}) \in [0; 1]^k$, where $\mathbf{y}$ is the output of the softmax layer.

The loss function here is the categorical cross-entropy loss. Below there is an objective function to minimize based on this loss:

$$Q(\mathbf{w}, \mathcal{D}_l, \mathcal{L}) = -\frac{1}{N_l} \sum_{i \in \mathcal{I}_l} \sum_{c=0}^{k-1} z_{i,c} \log y_{i,c}$$

where class labels z_i are encoded with the one-hot encoding and $\mathbf{y}_i = \mathrm{N}_{\mathbf{w}}(\mathrm{X}_i)$.

The final classification is made by means of agrmax rule over the outputs of the trained model:

$$\hat{f}(\mathrm{X}) = \arg\max_{c \in E} y_c$$

In the figure 3 one can see the whole workflow of the described approach.

Until recently this approach with different types of a models (i.e. HMM or Gaussian Mixtures) has been considered as a state-of-the-art.

The drawback of such an approach is obvious from it's name — it converts sequence of arbitrary length into one label. One more potential but nonetheless significant problem with one-label approach is on the fly work. In the online setting there is continuous audio flow without assessor's split into utterances. Thus the ability to predict the sequence of emotions per one utterance is crucial.

The most recent advance in the field of natural language processing is the use of so called end-to-end or sequence-to-sequence models which is described next in 4.2

4.2 CTC approach

Connectionist Temporal Classificaion (CTC) approach is the one among many end-to-end prediction methods. The main advantage of CTC is that it chooses the most probable label sequence regarding the various ways of aligning it with the initial sequence. The probability of the particular *labelling* is added up from the probabilities of every it's alignment.

More formally, let $\mathcal{D}_l$ still be the training set but unlike one-label approach the answer can be a sequence now.

Further we expand the label set E and introduce one more label — NULL. This label can be interpreted as a lack of emotion. Technically it reflects in one more unit in the last softmax layer of neural net. So the final label set is the following: $L = E \cup \{\text{NULL}\}$.

As in the one-label approach let's consider the work of the model as a mapping but here the codomain is also the sequence space $\mathrm{Y} = \mathrm{N}_{\mathbf{w}}(\mathrm{X}) \in [0; 1]^{(k+1) \times T}$. Here y_c^t is the output of the softmax layer and represents the estimation for the probability of observing class c in the moment t.

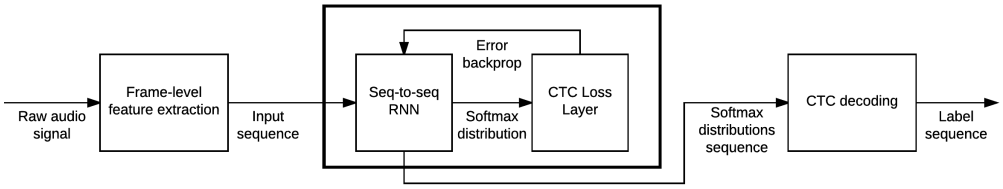


Figure 4: CTC approach pipeline

For every input X let's define the *path* $\boldsymbol{\pi}$ — it is the arbitrary sequence from L^* with the length T. Then the conditional probability of the path is

$$\mathrm{p}(\boldsymbol{\pi}|\mathrm{X}) = \prod_{t=1}^{T} y^t_{\pi_t}.$$

The problem is that paths contain NULL class labels which are unacceptable at the final output. So first need is to get rid of the NULLs. Therefore mapping $M : L^T \mapsto E^{\leq T}$ is introduced. It consist of the few stages:

1. Delete all consequent repeated labels,
2. Delete all NULLs

One can think of the following application: $M\left(-aa-b-b--ccc\right) = M\left(abb---bc-\right) = abbc$. From this example it is obvious that M is the surjective mapping. By means of it we transform paths to labellings. The method for computing the probability of the labelling is simple:

$$\mathrm{p}(\mathbf{l}|\mathrm{X}) = \sum_{\boldsymbol{\pi} \in M^{-1}(\mathbf{l})} p(\boldsymbol{\pi}|\mathrm{X}).$$

One could notice that the direct calculation of $\mathrm{p}(\mathbf{l}|\mathrm{X})$ requires summation over all corresponding paths which is very exhausting — there are $(k+1)^T$ possible paths. Thus Graves et al. in [6] derived a new efficient forward-backward dynamic programming algorithm for that. The initial idea was taken from Rabiner [7] HMM decoding algorithm.

Finally, the objective function to minimize in this approach is based on the maximal likelihood principle. We try to maximize the probabilities of all correct answers at the same time:

$$Q(\mathbf{w}, \mathcal{D}_l) = -\sum_{i \in \mathcal{I}_l} \log \mathrm{p}(\mathbf{z}_i|\mathrm{X}_i)$$

Neural network here plays a role of evaluator of probability measure p and the more it trains the more precise probability estimations it gives. To enable the neural network training with the standard gradient-based methods Graves et al. [6] also suggested differentiation technique naturally embedded into dynamic programming algorithm.

The final classification rule is the one that maximizes the probability of the answer:

$$\hat{f}(\mathrm{X}) = \arg\max_{\mathbf{l} \in E^{\leq T}} \mathrm{p}(\mathbf{l}|\mathrm{X})$$

In the figure 4 the pipeline for the CTC method is depicted in the way similar to one-label approach.

5 Emotion recognition experiments

In series of experiments we investigated different models and approaches to the emotion recognition task. All the code can be found in github repository [8]

As it was mentioned in section 2, data structure is the following — dialogues are broken into utterances by human assessors. Utterance is the least structural unit with emotion tag in original

labelling. But utterances considerably vary in length. Thus we decided to split them into overlapped frames of 0.2 seconds duration with overlapping of 0.1 second. The problem is that frames have no labels. While it is obvious that not all frames of angry utterance also can be referred as an angry frames.

Further in this paper, unless otherwise specified, the test set consists of 20% points randomly picked from dataset.

Also two neural network structures are mainly used:

- Simple LSTM. It consists of two consecutive LSTM layers with hyperbolic tangent activations followed by two classification dense layers.
- Bidirectional LSTM. Network contains two BLSM layers each of which splits into two parts — half of nodes are usual LSTM blocks and other half is BLSTM units.

More detailed diagrams are shown in the figure 10 and can be found in appendix A.

5.1 Framewise classification

The first method that was applied is framewise classification. The core of this method is to try to classify each frame separately. Under specified conditions we come up with the following workflow:

- Take two of the loudest frames from each utterance. Loudness in this context is the synonymous for spectral power
- Assign these frames with the emotion of the utterance
- Train the frame classification model on the derived dataset

Here we make an assumption that the emotion of the utterance contains not in all frames of it but only in the loudest ones. Experiment shows that 2 frames is the optimal number. Regarding the classification model we used Random Forest Classifier from scikit-learn package [9].

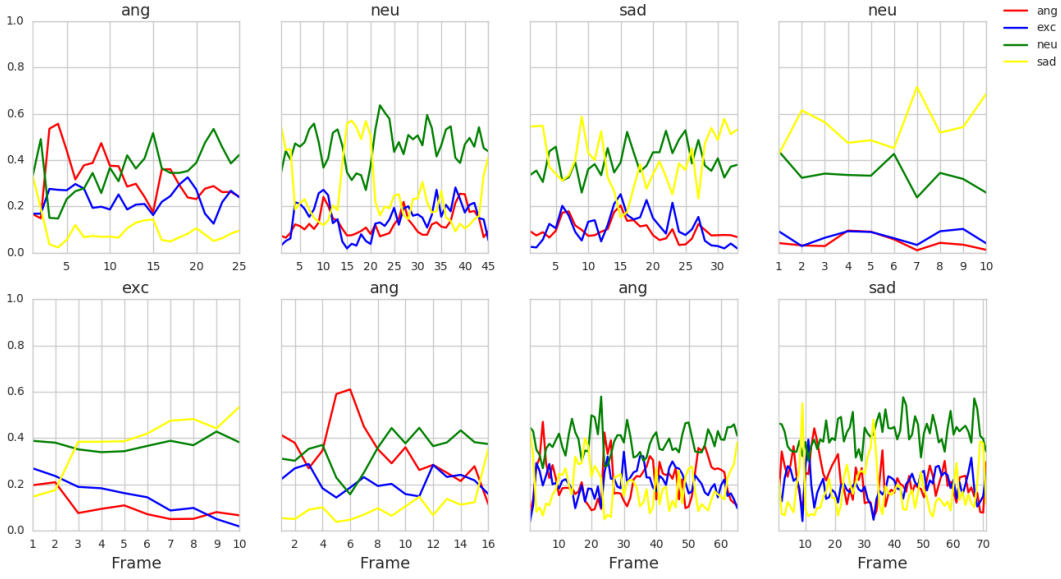


Figure 5: Framewise classification

In the figure 5 one can observe the results of this method for random utterances from the test set. In glance it looks somewhat reasonable with short utterances. On the longer ones it becomes sawtooth and unstable.

The next step is to classify utterances based on its' artificial labellings. Simple majority voting algorithm gives about 44% accuracy. Taking into account that neutral class is about 36% of a dataset it looks not very good. Moreover, the thing is that error distribution in that case looks unnatural in

comparison with the human one got by us. 70% of the answers are neutral, which implicitly confirm our assumptions that most of the frames in the utterances don't have any emotion.

Applying simple LSTM network doesn't give any significant improvement and BLSTM network drastically overfits and we can't escape from this effect without significant loss in quality.

Summarizing, frames pre-classification shows itself not brilliant in this task. We believe that the main reason is lack of markup on the frame level. Choosing two frame from utterance is too inaccurate method for labelling samples. On par with it, we can state that hypothesis about neutral character of most part of the frames holds.

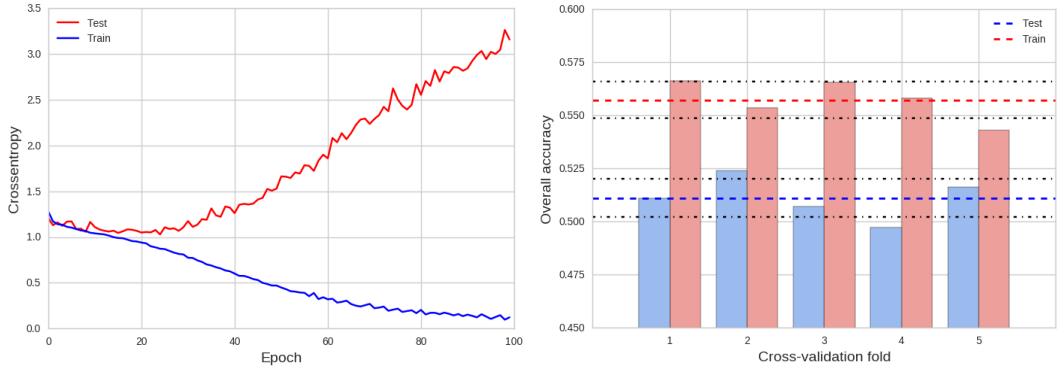
5.2 Utterance-level classification

The idea of the following experiment is to use frame features themselves as an input to the neural network model. There is a rationale behind it, because with the framewise approach we don't get any additional information after the intermediate RFC stage. The key idea in utterance level classification is that RNN can learn sufficient features from input features stream itself and made the final classification based on them.

Here we experiment with two approaches described in section 4. 5-fold cross-validation technique is used to evaluate and compare them.

5.2.1 One-label approach

BLSTM network outperforms Simple LSTM network within this approach. In the figure 6 the performance analysis is shown.



(a) Loss evolution

(b) 5-fold CV results

Figure 6: BLSTM performance in one-label approach

As one can see from the plots in figure 6a that 20-22 epochs is enough for training and after that model starts to overfit. Accuracy evolution curves can be found in figure 11a placed in appendix A. In the bar chart 6b the results of CV are shown. Number of epochs for training was chosen accordingly to previously obtained number — 20.

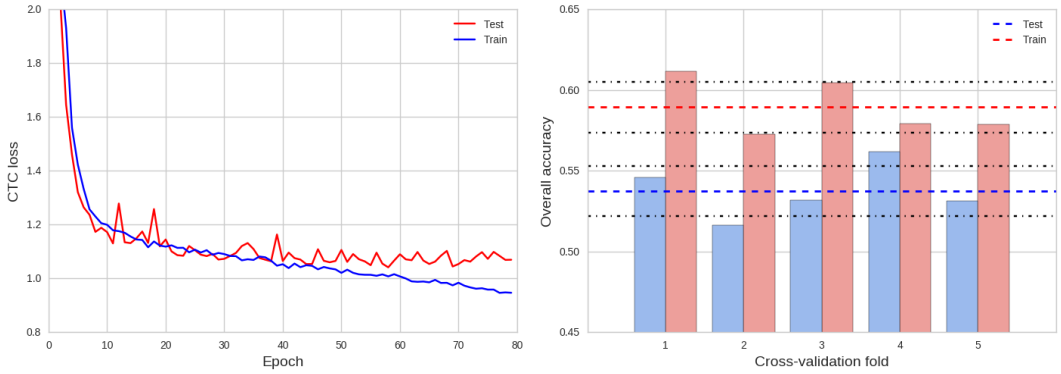
5.2.2 CTC approach

Performance indicators for CTC approach are given in the figures 7 and 11b in appendix A.

It is worth to mention that BLSTM with CTC loss requires approximately two times more epochs to train properly comparing to crossentropy loss. The interesting and important feature to notice is that network with CTC loss does not overfit.

5.3 Comparison

As we considering multiclass classification task with a little unbalanced classes, it is essential to look not only on overall accuracy but on class accuracies too. Thus we introduce two type of metrics:



(a) Loss evolution

(b) 5-fold CV results

Figure 7: BLSTM performance in CTC approach

- Overall (weighted) accuracy. It is an accuracy on the entire test set.
- Mean class (unweighted) accuracy. Is is average classification accuracies for each class.

In the table 1 the aforementioned metrics are shown for each method we used throughout the paper. The last line of the table corresponds to the human assessors who was asked to relabel this dataset. All the details about how and why it was done are in the sections 5.4 and 6

Table 1: Methods accuracies comparison

Method	**Overall accuracy**	**Mean class accuracy**
Framewise	45%	41%
One-label	51%	49%
CTC	54%	54%
Human	69%	70%

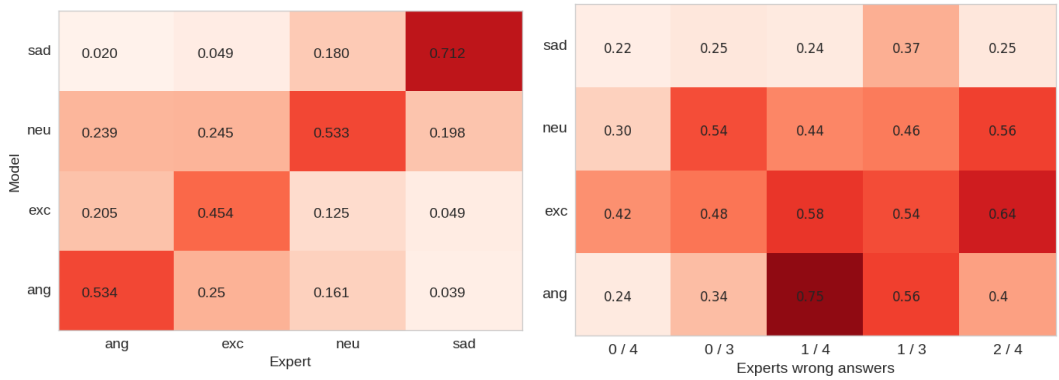
5.4 Error structure

We also decided to further investigate the CTC model because Graves in [6] reports about huge gap in quality over classical models. While here the gain is about 3-5%. It is still significant but not the breakthrough. For that reason the error structure of our most successful model is studied.

First of all, it's useful to take a look on predictions distribution in comparison with "ground truth" from experts. This distribution is shown in the figure 8a. Busso in his work [5] mentions that audio signal plays the main role in sadness recognition while angry and excitement are better detected via video signal which accompanies audio during the assessors work. This hypothesis seems to be true in relation to our model. Sadness recognition percentage is much higher than the others. It also maybe interesting to compare it with the human error structure shown in the figure 9a

Also as it was said in section 2 dataset has not the homogeneous structure in sense of expert answers reliability. It is interesting to see (fig. 8b) how the model predictions depend on the expert confidence degree. For that purpose we first differentiate utterances by the confidence degree of experts. On the x-axis one can see the number of experts whose answer differs from the final emotion assigned to the utterance. In each cell of a table there is a model error percentage with particular emotion corresponding to y coordinate and particular confidence level corresponding to x coordinate.

In fact this matrix gives curious information — if we take in account only those utterances in which experts were consistent then we get 65% accuracy. It sounds much more promising that 54% and this transition can be treated legal because emotion is something very subjective. Recognizing a truck or a flower on a picture can be evaluated objectively and in tasks like that computers have already outperform humans [10]. But emotion is something different which can't be defined properly. Thus



(a) Answers distribution

(b) Misclassification rates

Figure 8: BLSTM CTC error structure

even if human can't be sure in right understanding of emotions then our model can't guarantee it as well.

One more interesting thing to notice here is not only the quantity of inconsistent answers but also this answers themselves. As it was discussed previously some experts give answers not the same as the final emotion of the utterances. These answers can be arbitrary emotion and here we select only those four considered before. Thereby in the first row of the table 2 there is the percentage of inconsistent answers from utterances treated as having emotion with the name from the header of the column fallen into considered four emotions. And in the second row there is the percentage of model answers that coincide with inconsistent expert in this case.

Table 2: Residual accuracy

	Angry	**Excitement**	**Neutral**	**Sadness**
Considered ratio	17%	22%	36%	39%
Model accuracy	51%	73%	71%	74%

In other words table 2 shows how frequently the errors of our model coincide with the human divergence in emotion assessment.

All these facts and ideas lead us to the final and still ongoing part of the research described in section 6.

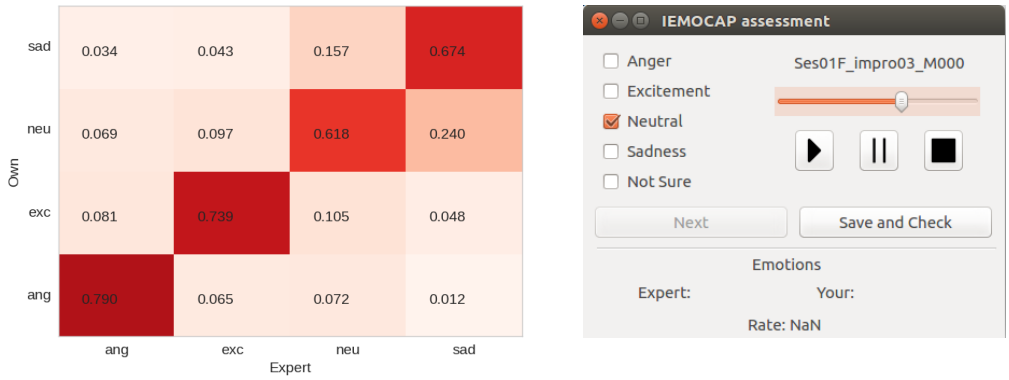
6 Markup investigation

Observing inconsistency of experts and other problems of the markup described in the sections 5.4 and 2 we come with the idea that it will be interesting and useful to see how humans perform in that task.

This question is not new and previously arised in the papers. Altrov in [11] collected the corpus of Estonian speech and asked people of different nationalities to evaluate it. Almost all nationalities (Latvians, Italians, Finns, Swedes, Danes, Norwegians, Russians) were close to Estonians geographically and culturally. Nevertheless Estonians perform much better than any other nationality showing about 69% mean class accuracy. All other people perform 10-15% worse and the only emotion that they recognise well was sadness.

In our research we developed a simple interface (fig. 9b) for relabelling speech corpus to see how well humans can solve this task. In the figure 9a you can see the results of the experiments taken.

Both types of accuracies we considered before are about 70% (tab. 1). These numbers confirm the idea that the emotion is the notion treated differently by different people. And having this 70% as a milestone we can say that our model performs very good.



(a) Human error structure

(b) Labelling interface

Figure 9: Assessor's analysis

7 Future work

For future we see two main directions of the research:

- Improve model characteristics. Now there are plenty of ways to do it. First thing is to further experiment with the net structure but based on our experience it can not greatly reduce the quality. The second way is to use more sophisticated feature generation framework or to come up with the new one. The final way that we see is to try to use raw audio signal as an input for the model. Recently Google DeepMind group shows [12] that it is possible to use raw audio signal to generate speech and more than that this algoritm greatly surpasses all other advances in this field. We believe that technical details of their work allow to adapt it to the emotion recognition task and this way looks the most promising.

- Improve markup and collect new dataset. As we showed in this paper both the markup and dataset collection methodology have a strong and crucial effect on the properties of the model. So one way it to try to relabel IEMOCAP dataset using developed interface and the new methodology which allows to tune our model better. Other way which we like the most is to collect new database in Russian language. In fact there is no russian emotional speech corpus which we can get freely. Now we are moving our labelling interface to web and soon it will be ready to launch. Also we think that there is no need to hire actors to read dialogues. Nowadays there are a lot of stand-ups, TED-talks etc. in the internet where one man speaks considerable amount of time with emotions. Our work is to split it and give it to the assessors. This way it will be even more natural than the artificially played emotions from actors.

8 Conclusion

In this paper we proposed a novel approach for emotion recognition from audio. There are two main advantages of a method:

- Through the CTC loss function it accounts for the fact that emotionality may be contained only in a few frames in the utterance

- It can predict the sequence of emotions for one utterance

Also we showed that the results are comparable with the state-of-the art ones in this field. Moreover we analyzed model answers and error distribution along with human performance and came to the conclusion that emotion is a very subjective notion and even if humans outperform computer the difference is not so significant.

References

[1] K. Han, D. Yu, and I. Tashev. Speech emotion recognition using deep neural network and extreme learning machine. In *Interspeech 2014*, 2014.

[2] J. Lee and I Tashev. High-level feature representation using recurrent neural network for speech emotion recognition. In *Interspeech 2015*, 2015.

[3] D. Ververidis and C. Kotropoulos. A review of emotional speech databases. In *Panhellenic Conference on Informatics*, pages 560–574, 2003.

[4] The Association for the Advancement of Affective Computing. http://emotion-research.net/wiki/Databases, 2014.

[5] C. Busso, M. Bulut, C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. Chang, S. Lee, and S. Narayanan. Iemocap: Interactive emotional dyadic motion capture database. *Journal of Language Resources and Evaluation*, 42(4), 2008.

[6] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. *Proceedings of the 23rd International Conference on Machine Learning*, 2006.

[7] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), 1989.

[8] V. Chernykh, G. Sterling, and P. Prihodko. https://github.com/vladimir-chernykh/emotion_recognition, 2016.

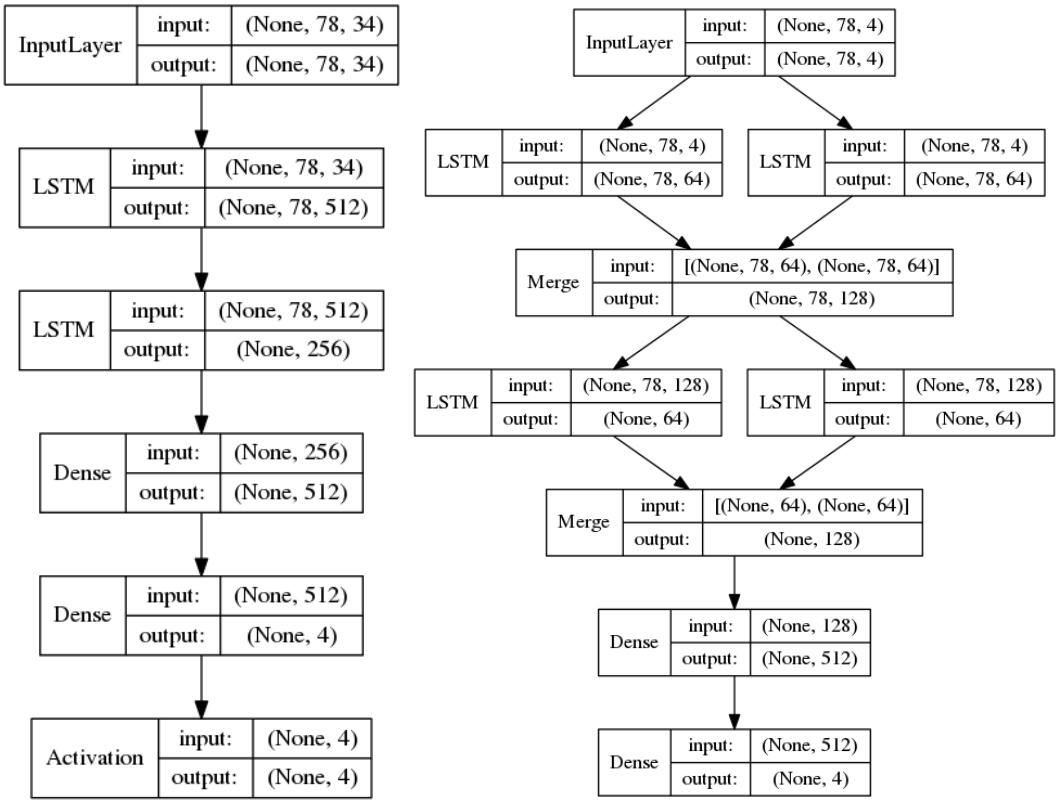
[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12, 2011.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV 2015*, 2015.

[11] R. Altrov and H. Pajupuu. The influence of language and culture on the understanding of vocal emotions. *Journal of Estonian and Finno-Ugric Linguistics*, 6(3), 2015.

[12] A. Van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *ArXiv e-prints*, 2016.

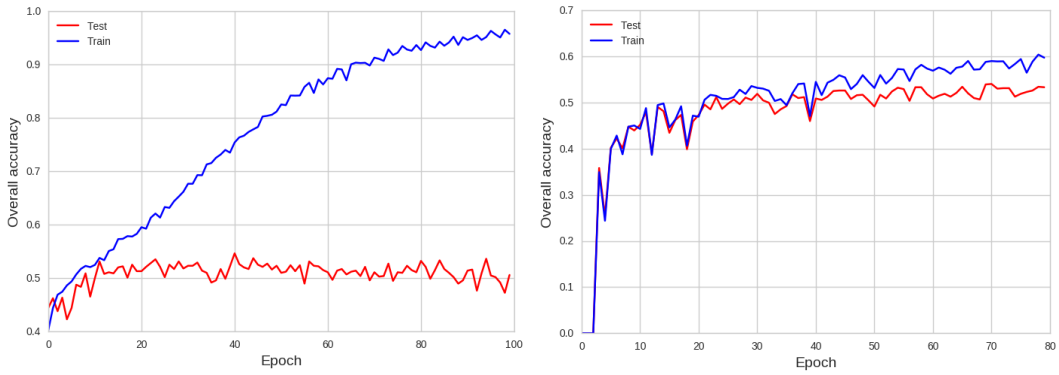
Appendix A



(a) Simple LSTM network

(b) BLSTM network

Figure 10: Network architectures



(a) One-label approach

(b) CTC approach

Figure 11: Accuracy evolution for BLSTM architecture