The Workshops of the Thirty-Second
AAAI Conference on Artificial Intelligence

Knowledge-Driven Feed-Forward Neural Network for Audio Affective Content Analysis

Sri Harsha Dumpala, Rupayan Chakraborty, Sunil Kumar Kopparapu
TCS Research and Innovation-Mumbai, India

Abstract

Machine learning techniques have shown great promises across domains, however they fail to impress when there is sparsity of training data. Work in the area of affective content analysis can not take complete advantage of machine learning techniques when there is a lack of sufficient training data. It is well known that recurrent neural networks (RNNs), particularly with long-short term memory (LSTM) units, perform better than feed-forward neural networks (FFNNs) on sequential data as they are architecturally designed to learn temporal relationships existing in the training data while FFNNs are not. But RNNs require sufficient training data to learn these temporal relationships. In this paper, we show that one can take advantage of a-priori knowledge about the temporal correlations in the training data even in a FFNN architecture. We call this the knowledge-driven FFNN or k-FFNN architecture. We show using the MediaEval dataset that the k-FFNN model not only outperforms FFNN, but also performs better than RNN models (i.e, Simple RNN, RNN with LSTM units and bi-directional RNN with LSTM units (BLSTM)), especially when the amount of training data is sparse.

1 Introduction

Affective content analysis in audio-visual clips is an active and nascent research area that refers to the automatic recognition of emotions elicited by clips, which can aid in emotion-based personalized content retrieval (Canini, Benini, and Leonardi 2013; Wang et al. 2012), audio-visual indexing (Li, Narayanan, and Kuo 2004; Zhang et al. 2010), summarization (Evangelopoulos et al. 2008; Furini and Ghini 2006; Katti et al. 2011), to name a few application areas. There are distinctively two modalities involved in such analysis, one is based on audio content and the other is based on visual content. Most often, fusing both the modalities, to make use of the complementary information, improves performance. Moreover, affective content analysis in videos can be categorized as encoded and decoded approaches. Encoded approach infers the affective content directly from the audio-visual features of the related video whereas decoded approach analyzes the affective state of the viewer while watching the videos. However, the latter approach that makes use of multiple viewers is followed for annotating the videos for affective content analysis.

Knowledge regarding the temporal relationship within a clip can play an important role for affective content analysis. Modern deep learning techniques like Recurrent Neural Networks (RNNs) (Elman 1990) have an inherent ability to automatically learn and exploit the temporal relationships amongst the sequences (Rumelhart, Hinton, and Williams 1988; Werbos 1988), and require huge training data to capture these correlations. Infact, such temporal correlations exist in LIRIS-ACCEDE dataset (LIR 2016; Baveye et al. 2015), a part of which is used in MediaEval "*The Emotional Impact of Movies Task*" (Med 2016a). It is one of the benchmarking initiative that has been taken for affective video content analysis with an aim for the users to find videos that fit for particular mood, age or preferences (Med 2016b). In MediaEval 2016, one of the main subtasks was to predict global emotion, given a short video clip of around 10 seconds duration. The participant's system is expected to predict a score of induced valence (negative or positive) and induced arousal (calm or excited) for the complete video clip. However, high values of Mean Squared Errors (MSE) (plus corresponding low values of Pearson Correlation Coefficients (PCC)) in most of the reported results highlighted the difficulties of achieving reasonable result even with the latest deep learning techniques in both the modalities (Pro 2016). Generally, each video clip in the dataset has an affective fade in at the beginning and an affective fade out at the end of the video clip, which are not reflected in the final annotation of the clip. This implicit knowledge about the temporal relationship of the affective contents remains unused. The perceived affect of a video clip has an affective fade in and affective fade out, which is not uniform for the entire duration of the clip. This motivates us to propose a technique which can exploit the knowledge regarding the temporal relationships within the sequence. This is advantageous, especially in a limited resource scenario.

In this paper, we propose to exploit the knowledge regarding the temporal relationships in a feed-forward Neural Network (FFNN) architecture. In addition, knowing that RNN architecture is capable of inherently learning the temporal relationships that exist in the sequential data, we consider different RNNs i.e, Simple RNN, RNN with long-short term memory (LSTM) units (Hochreiter and Schmidhuber 1997) and bidirectional RNN with LSTM units (BLSTM) (Schuster and Paliwal 1997), (Graves and Schmidhuber 2005), to

automatically capture the temporal information. But to learn that automatically using RNN, substantial amount of training data is required. Now the question that we are addressing here is "*Can a regular FFNN fed with temporal knowledge perform as well as an RNN?*" We attempt to capture the temporal relationship within the sequences to train k-FFNN (short form of knowledge-driven FFNN) and then subsequently show through extensive experiments that the knowledge of temporal relationship can enhance the performance of FFNN. Using MediaEval 2016 audio data ("*Emotional Impact of Movies*" task)(Med 2016a)), we conducted several experiments to establish that the performance of k-FFNN is comparable to RNNs (Simple RNN, LSTM and BLSTM networks) and in some scenarios k-FFNN outperforms RNNs. The main contributions of this paper are (a) incorporation of a-priori temporal knowledge in FFNN to construct a k-FFNN and (b) experimentally showing that not only the performance of k-FFNN is as good as RNN, but also better when training data is sparse.

2 Affective Content Analysis using Temporal Knowledge

For affective content analysis, we consider a traditional feed-forward neural network for learning the temporal information which we call k-FFNN, and also consider the RNN architecture for performance comparison. The primary difference between RNN and FFNN is the presence of the self feedback loop in hidden layer of RNN which adds memory over time. The question that we are addressing is can a regular FFNN fed with the sequence based a-priori knowledge (i.e. k-FFNN) perform as well as an RNN. In other words, if we had some a-priori knowledge about the sequence can we use it to train a FFNN so that we can avoid dependency on RNN. This is very useful, particularly in the scenarios where the training data is sparse and when we are aware of the temporal relationship within the input sequences a-priori.

2.1 Knowledge-driven FFNN

Let us consider a FFNN with I input, H hidden and O output nodes and data with sequences of length N. We assume that there exists some temporal relationship between ($\vec{g}_{1,1}$, $\vec{g}_{1,2}$,...,$\vec{g}_{1,N}$) and ($\vec{g}_{2,1}$, $\vec{g}_{2,2}$,...,$\vec{g}_{2,N}$), which can be represented as shown in Table 1 for k-FFNN. Namely, the output y_k associated with $\vec{g}_{1,1}$, $\vec{g}_{1,2}$,...,$\vec{g}_{1,N}$ is actually $f(1)y_1$, $f(2)y_1$,..., $f(N)y_1$ instead of y_1, y_1,...,y_1. The $f()$ is the mode in which the a-priori temporal knowledge existing between $\vec{g}_{1,1}$, $\vec{g}_{1,2}$,...,$\vec{g}_{1,N}$ in the training set. We elaborate the process of training the k-FFNN. We assume the usual back-propagation based weight updation. In case of k-FFNN, the error computed at the output as,

$$\epsilon = (o_1 - f(1)y_1)^2 \tag{1}$$

which is used to modify the weights ($^{(ih)}w$, $^{(ho)}w$) of neural network (Dumpala, Chakraborty, and Kopparapu 2017). We have used steepest gradient descent algorithm for weight updation. The "hidden to output" weights (i.e. $^{(ho)}w_k$ for $k = 1, 2, .., H$) and the "input to hidden" weights (i.e. $^{(ih)}w_{jk}$ for $j = 1, 2, ..., I; k = 1, 2, .., H$) for k-FFNN are modified as

Input				Output		
				FFNN	k-FFNN	RNN
$g^1_{1,1}$	$g^2_{1,1}$	..	$g^I_{1,1}$	y_1	$f(1)y_1$	-
$g^1_{1,2}$	$g^2_{1,2}$	..	$g^I_{1,2}$	y_1	$f(2)y_1$	-
..	..	..	..	..	..	..
$g^1_{1,N}$	$g^2_{1,N}$	..	$g^I_{1,N}$	y_1	$f(N)y_1$	y_1
$g^1_{2,1}$	$g^2_{2,1}$	..	$g^I_{2,1}$	y_2	$f(1)y_2$	-
$g^1_{2,2}$	$g^2_{2,2}$	..	$g^I_{2,2}$	y_2	$f(2)y_2$	-
..	..	..	..	..	..	..
$g^1_{2,N}$	$g^2_{2,N}$	..	$g^I_{2,N}$	y_2	$f(N)y_2$	y_2

Table 1: Representation of input-output pairs used in our experiments for training FFNN, k-FFNN and RNNs.

$$\begin{aligned} ^{(ho)}w_k &\leftarrow {}^{(ho)}w_k + \Delta^{(ho)}w_k \\ ^{(ih)}w_{jk} &\leftarrow {}^{(ih)}w_{jk} + \Delta^{(ih)}w_{jk} \end{aligned} \tag{2}$$

where $\Delta^{(ho)}w_k$ and $\Delta^{(ih)}w_{jk}$ are presented as,

$$\Delta^{(ho)}w_k = \eta(o_1 - f(1)y_1).h_k \tag{3}$$

$$\Delta^{(ih)}w_{jk} = \eta(o_1 - f(1)y_1).h_k(1 - h_k)^{(ih)}w_{jk}g^j_{11} \tag{4}$$

2.2 Recurrent Neural Networks

RNNs are the powerful class of neural networks that include weighted connections within a layer (Elman 1990). This allows RNNs to store temporal information present within the input data. Simple RNNs (Elman 1990) suffer from the problem of vanishing and exploding gradients (Pascanu, Mikolov, and Bengio 2013). As a solution to address the vanishing gradient problem, long-short term memory (LSTM) architecture for RNNs was proposed (Hochreiter and Schmidhuber 1997). In LSTM networks, the hidden units are replaced by recurrently connected subnets, called memory blocks. Each memory block consists of a memory cell and three gates: input, output and forget gate which perform the analogous read, write and reset operations respectively, for the cell. These gates allow LSTM memory cells to store and access information over very long sequences, thereby avoiding the vanishing gradient problem.

Further, we also considered bidirectional RNNs (BRNNs), which can access the input data from both temporal directions (Schuster and Paliwal 1997). BRNNs contain two separate hidden layers to process the input data from both directions and both hidden layers are fed to the output layer. In this paper, we use BRNNs with LSTM units, which are called BLSTMs (Graves and Schmidhuber 2005).

All RNNs (Simple RNN, LSTM and BLSTM) considered in our analysis are trained using data as shown in Table 1. Each $\vec{g}_{i,j}$ is of dimension I and y_k represents the output value corresponding to the k^{th} sample in the training set, where the sequence length of each sample is N.

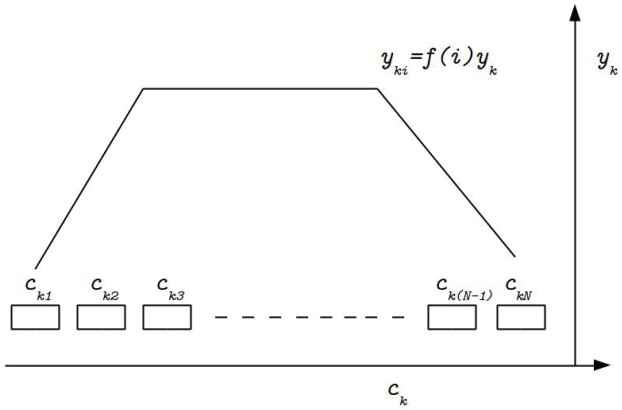

Figure 1: $\{c_k; y_k\}$ pair

3 Working Scenario and Dataset Preparation

MediaEval2016 *Emotional Impact of Movies Task* (sub-task: *Global prediction for short video excerpts*) is used in our analysis (Med 2016a). This dataset is part of the LIRIS-ACCEDE dataset (LIR 2016) and consists of video clips of duration 8-12 seconds. These video clips are annotated by viewers by their perceived emotion, in terms of arousal and valance with values in the range of $[0, 5]$.

We created a dataset of smaller *1-second* video clips by segmenting original video clips. Namely, a video clip of 10 seconds duration produced 10 *1-second* video clips. We retained the temporal relationship within the original video clip by naming the segmented *1-second* video clips, in an order. This enabled us to incorporate the temporal correlations, in terms of the affective fade in and affective fade out, between the segments to test the k-FFNN.

In our experiments, we considered only the audio obtained from video clips as the input and the corresponding annotated arousal or valence values as the desired output. For testing the hypothesis, we first extracted the audio from the original video clip and then segmented the audio into smaller non-overlapping *1-second* duration, so a movie clip of N seconds ($N \in [8, 12]$) duration, resulted in N audio clips each of *1-second* duration. For example, if c_k is the audio extracted from the original k^{th} video then,

$$c_k = \oplus_{i=1}^{N} c_{ki} \tag{5}$$

where $\oplus$ represents the concatenation of the audio c_{ki} for $i = 1, \cdots, N$. Note that there is a temporal relationship between c_{ki}'s because they are in a time sequence and are from a single video clip. This construction (eq. 5) helps us in building a dataset that can be used to analyze our hypothesis, namely, a FFNN driven with temporal knowledge can work as well as an RNN in terms of its overall performance when used for predicting the estimated emotion of a movie clip.

Let $\{c_k; y_k\}$ be the input output pair; where $y_k \in [0, 5]$ can be either valence (v_k) or arousal (a_k) associated with the audio c_k. As seen in Figure 1, the audio c_k is made up of the $c_{k1}, c_{k2}, \cdots, c_{kN}$ audio sequences. So for an RNN we have the input as $c_{k1}, c_{k2}, \cdots, c_{kN}$ while the output is the associated y_k (v_k or a_k). However, since the input $c_{k1}, c_{k2}, \cdots, c_{kN}$ are temporally related, we assumed that the perceived affective content value y_k (for valence v_k, and for arousal a_k) has a bearing on c_{ki}, namely,

$f(1)$	$f(2)$	-	-	-	-	$f(7)$	$f(8)$	Type
1	1	1	1	1	1	1	1	FFNN
0.75	0.9	1	1	1	1	0.9	0.75	Fn1
0.3	0.6	1	1	1	1	0.6	0.3	Fn2
0.1	0.2	1	1	1	1	0.2	0.1	Fn3

Table 2: Different $f(i)$ used in experiments, e.g. for $N = 8$.

$$y_{ki} = f(i)y_k \tag{6}$$

where $f(i)$ captures the known a-priori temporal knowledge.

However for both FFNN and RNN, the input data is the same while the output in case of RNN is known (y_k), we construct y_{ki} by repeating the value of y_k for each c_{ki} to train knowledge-driven FFNN.

4 Experimental Validation

In our experiments, we used the audio extracted from 7571 video clips (from MediaEval database) each of N (around $8 - 10$) seconds duration (Pro 2016). The database has arousal and valence value in the range $[0, 5]$ for all 7571 videos, namely $(c_k; y_k)$ for $k = 1, 2, \cdots, 7571$. We constructed $c_{k1}, c_{k2}, \cdots, c_{kN}$ each of 1 second duration from c_k of N second duration for $k = 1, 2, \cdots, 7571$ (see eq. (5)). For each $\{c_{kj}\}_{k=1,j=1}^{k=7571,j=N}$ we extracted 384 features (as were used in Interspeech 2009 Emotion Challenge (Schuller, Steidl, and Batliner 2009)) by using openSMILE toolkit (openSMILE 2017). Further, the feature dimension was reduced to 21 by using the correlation-based feature selection method (i.e. *CfsSubsetEval*) from WEKA Toolkit (WEKA 2016).

The data representation formats used for training FFNN, k-FFNN and RNN are shown in Table 1, where $\vec{g}_{k,j}$ represent the features extracted from c_{kj}. As shown in Table 2, we used a variety of $f(i)$s in our experiments to capture the affective fade-in and affective fade-out.

4.1 Experimental Analysis

The performance of the proposed k-FFNN is compared with that of different RNNs i.e., simple RNN (here also represented as RNN), LSTM and BLSTM. In this analysis, all FFNN, k-FFNN and RNN systems are implemented using Keras deep learning toolkit (KER 2017). For all systems, only a single hidden layer is considered. Number of units in the hidden layer are selected by using 5-fold cross-validation on a validation set, where the number of units are varied from 11 (half the sum of number of input (i.e., 21) and output units (i.e., 1)) to 44 (twice the sum of number of input and output units). Sigmoid units are used for the hidden layer. The input layer has 21 linear units and the output layer has a single linear unit.

The system performance is evaluated in terms of Mean Squared Error (MSE) and Pearson Correlation Coefficient

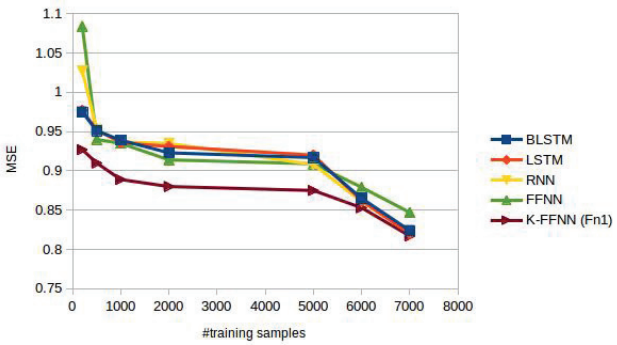

Figure 2: MSEs for different sizes of training set (arousal prediction).

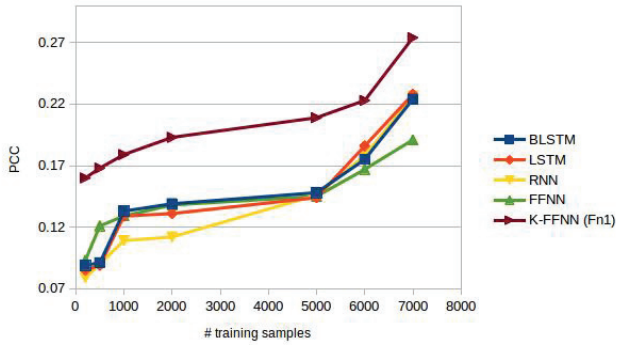

Figure 3: PCCs for different sizes of training set (arousal prediction).

(PCC). PCC along with MSE is used as a performance metric as PCC provides a better evaluation of the systems trained on datasets with a non-uniform distribution of output values as observed for both arousal and valence. For the considered metrics, lower the MSEs and higher the PCC values, better is the performance of the system.

The MSE and PCC values are computed for each complete clip for all the systems (FFNN, k-FFNN and RNNs). In case of RNNs, single output value y_k is obtained for the given c_k containing the sequence $c_{k1}, c_{k2}, \cdots, c_{kN}$. Subsequently, the computation of MSE and PCC is straight forward in case of RNNs. In case of a k-FFNN, for each subsegment $c_{k1}, c_{k2}, \cdots, c_{kN}$ corresponding to the audio clip c_k, arousal or valence values are generated. To compute the MSE and PCC values for each audio clips c_k, the output values obtained for each clips are scaled with a value depending on the function selected during training. Then the mean of the values obtained at all subsegments is computed and compared with the original value y_k assigned to that audio clip to obtain the MSE and PCC values. If $y_1^{'}$, $y_2^{'}$, $y_3^{'}$, ..., $y_N^{'}$ are the output obtained for all the audio segment corresponding to the audio clip c_k, then

$$Y^{'} = \sum_{i=1}^{N} y_i^{'}(1/f(i)) \tag{7}$$

is the predicted arousal or valence value for the audio clip c_k.

The MSE and PCC values obtained for different systems by considering training sets of different sizes are shown in Figure 2 and Figure 3, respectively. As shown in Figure 2, the MSE obtained for k-FFNN (Fn1) is always lower or equal to that of the MSE obtained for RNNs across all sizes of training sets. In particular, k-FFNN outperforms RNNs when limited data is used for training. For instance, k-FFNN has an improvement of 0.05 in MSE over RNN (MSE is 0.927 for k-FFNN and 0.977 for RNN) when only 200 samples are used for training. However, the MSE obtained for k-FFNN is similar to that of RNN when 6814 (90% of dataset) samples are used for training. The MSE is lower for FFNN compared to RNN for smaller training sets but are higher when the training set size is increased (MSE is 0.940 and 0.847 for FFNN and 0.953 and 0.820 for RNN when 500 and 6814 samples are considered for training). This shows that the performance of k-FFNN in terms of MSE is better than FFNN and RNN, especially for smaller training sets.

Function	MSE	PCC
Fn1	0.820	0.274
Fn2	0.871	0.185
Fn3	1.55	0.059

Table 3: Performance (MSEs and PCCs) of different k-FFNN systems for arousal prediction.

It can be observed from Figure 3 that the PCCs are consistently higher for k-FFNNs compared to RNNs. Similar to MSEs, the differences in PCCs of k-FFNN and RNN are higher for smaller training sets (difference = 0.08 (0.079 for RNN and 0.16 for Fn1) when 200 train samples are used) and gradually decreases when the size of the training set is increased (difference = 0.048 (0.226 for RNN and 0.274 for k-FFNN), when 6814 train samples are considered). The PCC of FFNN is lower compared to k-FFNN for all sizes of training set. PCC is higher for FFNN compared to RNNs for smaller training sets but is lower when size of the training set is increased

Table 3 shows the MSE and PCC values when different functions (shown in Table 2) are used to represent the temporal information for arousal prediction. As given in Table

System	MSE	PCC
k-FFNN (Fn1)	0.319	0.128
k-FFNN (Fn2)	0.454	0.029
k-FFNN (Fn3)	0.762	-0.051
RNN	0.331	0.126
LSTM	0.327	0.124
BLSTM	0.329	0.122
FFNN	0.343	0.106

Table 4: Performances (MSEs and PCCs) of different k-FFNN systems for valence prediction.

3, the performances (MSEs and PCCs) of the k-FFNN system trained using Fn1 are higher than the systems trained using Fn2 and Fn3. Also, the performance of the k-FFNN systems trained using Fn2 and Fn3 is lower than FFNN. This shows that a proper choice of function to represent the temporal information is critical for the performance of k-FFNN systems, and the performance of k-FFNN may even degrade when the considered function is not appropriate.

Table 4 shows the MSE and PCC values obtained for different systems (trained on 6814 utterances) for valence prediction. It can be observed that the performance of k-FFNN system (using Fn1) is better than RNN and FFNN systems. But the performance of k-FFNN systems (using Fn2 and Fn3) is lower than RNN and even FFNN. Hence the observations made from arousal prediction are further supported by these results.

5 Conclusions

In this paper, we have shown how one can use the known a-priori temporal knowledge about the sequential data to enhance the performance of FFNN architecture for affective content analysis. RNNs are able to implicitly learn the temporal correlations that exist within the data sequence. But RNNs are advantageous when (a) one is not explicitly aware of the temporal relationship between the sequential data and (b) when there is a large amount of training data. In this paper, we address the scenario where there is insufficient (or limited) training data, and in addition, a-priori temporal knowledge about the training data is explicitly known. We show that k-FFNN, which exploits the known a-priori sequential knowledge in the training data, helps to give better performances in limited data resources. We show using the MediaEval dataset that the k-FFNN models not only outperforms FFNN, but also performs better than RNN models (i.e, Simple RNN, LSTM and BLSTM) especially when the amount of training data is sparse.

References

Baveye, Y.; Dellandra, E.; Chamaret, C.; and Chen, L. 2015. Liris-accede: A video database for affective content analysis. *IEEE Transactions on Affective Computing* 6(1):43–55.

Canini, L.; Benini, S.; and Leonardi, R. 2013. Affective recommendation of movies based on selected connotative features. *IEEE Transactions on Circuits and Systems for Video Technology* 23(4):636–647.

Dumpala, S. H.; Chakraborty, R.; and Kopparapu, S. K. 2017. k-ffnn: A priori knowledge infused feed-forward neural networks. *CoRR* abs/1704.07055.

Elman, J. L. 1990. Finding structure in time. *Cognitive science* 14(2):179–211.

Evangelopoulos, G.; Rapantzikos, K.; Potamianos, A.; Maragos, P.; Zlatintsi, A.; and Avrithis, Y. 2008. Movie summarization based on audiovisual saliency detection. In *2008 15th IEEE International Conference on Image Processing*, 2528–2531.

Furini, M., and Ghini, V. 2006. An audio-video summarization scheme based on audio and video analysis. In *3rd IEEE Consumer Communications and Networking Conference*, volume 2, 1209–1213.

Graves, A., and Schmidhuber, J. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks* 18(5):602–610.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Katti, H.; Yadati, K.; Kankanhalli, M.; and Tat-Seng, C. 2011. Affective video summarization and story board generation using pupillary dilation and eye gaze. In *2011 IEEE International Symposium on Multimedia*, 319–326.

2017. François chollet "keras". https://github.com/fchollet/keras/.

Li, Y.; Narayanan, S.; and Kuo, C. C. J. 2004. Content-based movie analysis and indexing based on audiovisual cues. *IEEE Transactions on Circuits and Systems for Video Technology* 14(8):1073–1085.

2016. Liris-accede database. http://liris-accede.ec-lyon.fr/database.php.

2016a. The 2016 emotional impact of movies task. http://www.multimediaeval.org/mediaeval2016/emotionalimpact/index.html.

2016b. Mediaeval:emotional impact of movies task. http://www.multimediaeval.org/.

openSMILE. 2017. http://www.audeering.com/research/opensmile.

Pascanu, R.; Mikolov, T.; and Bengio, Y. 2013. On the difficulty of training recurrent neural networks. In *ICML*, 1310–1318.

2016. Mediaeval 2016 proceedings. http://ceur-ws.org/Vol-1739/MediaEval_2016_paper_6.pdf.

Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J. 1988. Neurocomputing: Foundations of research. Cambridge, MA, USA: MIT Press. chapter Learning Representations by Back-propagating Errors, 696–699.

Schuller, B. W.; Steidl, S.; and Batliner, A. 2009. The INTERSPEECH 2009 emotion challenge. In *INTERSPEECH*, 312–315.

Schuster, M., and Paliwal, K. K. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45(11):2673–2681.

Wang, J.-C.; Yang, Y.-H.; Wang, H.-M.; and Jeng, S.-K. 2012. The acoustic emotion gaussians model for emotion-based music annotation and retrieval. In *ACM Multimedia*, MM '12, 89–98. New York, NY, USA: ACM.

WEKA, T. 2016. http://www.cs.waikato.ac.nz/ml/weka/.

Werbos, P. J. 1988. Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks* 1(4):339 – 356.

Zhang, S.; Huang, Q.; Jiang, S.; Gao, W.; and Tian, Q. 2010. Affective visualization and retrieval for music video. *IEEE Transactions on Multimedia* 12(6):510–522.