Interspeech 2018
2-6 September 2018, Hyderabad

Multi-modal attention mechanisms in LSTM and its application to acoustic scene classification

Teng Zhang[1], *Kailai Zhang*[2], *Ji Wu*[3]

[123]Multimedia Signal and Intelligent Information Processing Lab
Department of Electronic Engineering, Tsinghua University, Beijing, P.R.China
zhangteng1887@gmail.com, zhang-kl13@tsinghua.org.cn, wuji_ee@mail.tsinghua.edu.cn

Abstract

Neural network architectures such as long short-term memory (LSTM) have been proven to be powerful models for processing sequences including text, audio and video. On the basis of vanilla LSTM, multi-modal attention mechanisms are proposed in this paper to synthesize the time and semantic information of input sequences. First, we reconstruct the forget and input gates of the LSTM unit from the perspective of attention model in the temporal dimension. Then the memory content of the LSTM unit is recalculated using a cluster-based attention mechanism in semantic space. Experiments on acoustic scene classification tasks show performance improvements of the proposed methods when compared with vanilla LSTM. The classification errors on LITIS ROUEN dataset and DCASE2016 dataset are reduced by 16.5% and 7.7% relatively. We get a second place in the Kaggle's YouTube-8M video understanding challenge, and multi-modal attention based LSTM model is one of our best-performing single systems.

Index Terms: acoustic scene classification, multi-modal attention, long short-term memory

1. Introduction

Sequence modeling problem has been the core issue for a great variety of sequence classification tasks. These tasks get a sequence as the input and output labels or targets of the sequence. Sequences can be a series of words in many natural language processing (NLP) applications such as name entity recognition, sentence classification and machine translation. Sequences can also be a sequential audio signal in audio processing domain such as speaker recognition, speech emotion classification and acoustic scene classification. In the field of acoustic scene classification [1][2], audio frames can be formulated as sequences. The primary objective of sequence modeling problem is to learn the vector representations of input sequences.

Long short-term memory (LSTM) model [3] is a frequently-used solution for sequence modeling and has shown significant improvement on text classification [4], acoustic scene classification [1] and video understanding[5]. In this paper, we follow the implementation described in [6] to implement our vanilla LSTM unit. For a standard LSTM model, it composes every frame in a sequence from the beginning to the end, and then give a final vector representation of the input sequence. LSTMs are explicitly designed to solve the long-term dependency problem. But for some intricate sequences especially in audio, the memory in LSTM models is not that credible, we also need to identify salient memory content to supplement to the final vector representation.

Attention mechanism is first introduced in machine translation task [7]. This mechanism is designed to take care of the positions of input sequences according to previous output. For sequence classification tasks, Shen [8] utilized the similarity between the final output vector of an LSTM model and the embedding vectors of the input sequence to calculate the attention weights, and then a weighted sum of all the embedding vectors is used as the representation of the input sequence. In [9], the author used the similarity between the final output vector of an LSTM model and the convolutional outputs of a convolutional neural network (CNN) model to make attention-based pooling over the convolutional outputs. Both of them used a LSTM model to get a global representation of input sequence, and then gave more attention to the embedding vectors or convolutional outputs when they were more similar to the global representation.

In this paper, we expand our previous work [10] and propose two different attention mechanisms on the basis of the vanilla LSTM. In the temporal dimension, the forget and input gates are reconstructed using the low-rank second order association between the temporal input and the previous hidden state, and then the final output vector presentation is replaced with a weighted sum of the output sequence, where the weights are calculated using a softmax layer of the gate values along the temporal dimension. Inside the LSTM unit, the weighted sum of difference vectors between the temporal input and its corresponding cluster center is stored as the memory content, where the weights are calculated using a softmax layer of the newly proposed cluster gates in semantic space and the cluster centers are learned jointly with the LSTM.

The rest of this paper is organized as follows: The proposed attention mechanisms are described in details in Section 2. The performance of proposed methods are compared on audio classification task and video classification task in Section 3 and Section 4. Sec.5 is the conclusion of our task.

2. Multi-modal attention mechanisms in LSTM

2.1. Preliminary: LSTM

In this section, we briefly describe the vanilla LSTM structure in [6]. The input audio can be represented as a sequence of vectors $\boldsymbol{X}_{1...T} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_T\}$. T is the audio length and the dimension of each vector $\boldsymbol{x}$ can be labeled as M, which is determined by the audio feature extraction methods. Let $\boldsymbol{h}_t$ be an N-dimensional hidden state in timestep t. Let $\boldsymbol{L}_{n,m} : R^n \to R^m$ be a biased linear mapping $x \to Wx + b$ for some W and b. The symbol $\odot$ represents element-wise multiplication. Then

10.21437/Interspeech.2018-1138

the LSTM can be described as following equations:

$$\begin{aligned}
&LSTM : \boldsymbol{x}_t, \boldsymbol{h}_{t-1}, \boldsymbol{c}_{t-1} \rightarrow \boldsymbol{h}_t, \boldsymbol{c}_t \\
&\begin{pmatrix} \boldsymbol{i}_t \\ \boldsymbol{f}_t \\ \boldsymbol{o}_t \\ \boldsymbol{c}_{\_} \end{pmatrix} = \begin{pmatrix} sigm \\ sigm \\ sigm \\ tanh \end{pmatrix} \boldsymbol{L}_{M+N,4N} \begin{pmatrix} \boldsymbol{x}_t \\ \boldsymbol{h}_{t-1} \end{pmatrix} \\
&\boldsymbol{c}_t = \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \boldsymbol{c}_{\_} \\
&\boldsymbol{h}_t = \boldsymbol{o}_t \odot tanh(\boldsymbol{c}_t)
\end{aligned} \tag{1}$$

The structure of LSTM allows it to learn the long-term dependencies easily. The long-term memory is stored in a vector of memory cell $\boldsymbol{c}_t \in R^n$, the short-term memory is another vector $\boldsymbol{c}_{\_}$ in Eq.1. As a practice, the final hidden state is utilized to represent the whole audio sequence.

2.2. Temporal Attention Mechanism

Not all frames in a sequence are equally informative for sequence classification tasks. For acoustic scene classification task, audio fragments that are too quiet or noisy contribute little to the audio theme. From the perspective of attention model, the vector representation of the whole audio sequence can be computed as a weighted sum of the hidden states $\boldsymbol{h}_t$ as Eq.2, α_t represents the contribution of each audio frame to the final vector representation $\boldsymbol{v}$. In this section, we introduce the temporal attention mechanism [7] to calculate α_t.

$$\boldsymbol{v} = \sum_{t=1}^{T} \alpha_t \boldsymbol{h}_t \tag{2}$$

In LSTM unit described as Eq.1, the input vector $\boldsymbol{x}_t$ and the previous hidden state $\boldsymbol{h}_{t-1}$ are used to decide when to keep or override information in the memory cell, which is closely related to the saliency of audio frames. Thus a softmax layer of $\boldsymbol{x}_t$ and $\boldsymbol{h}_{t-1}$ can be used to calculate α_t as Eq.3, where g is an attention function $R^{2N} \rightarrow R$ that calculates an un-normalized alignment score between $\boldsymbol{x}_t$ and $\boldsymbol{h}_{t-1}$.

$$\alpha_t = \frac{exp(g(\boldsymbol{x}_t, \boldsymbol{h}_{t-1}))}{\sum_{t=1}^{T} exp(g(\boldsymbol{x}_t, \boldsymbol{h}_{t-1}))} \tag{3}$$

In our attention model, we use a function of the form $g(\boldsymbol{x}_t, \boldsymbol{h}_{t-1}) = \langle \boldsymbol{W}_x \boldsymbol{x}_t, \boldsymbol{W}_h \boldsymbol{h}_{t-1} \rangle$, where the matrices $\boldsymbol{W}_x$ and $\boldsymbol{W}_h$ are used to transform $\boldsymbol{x}_t$ and $\boldsymbol{h}_{t-1}$ into a representation of the same size. Thus the temporal attention LSTM unit can be modified as Eq.4. In this LSTM version, the input gate i_t and the forget gate f_t become scalers, the attention function g_t is represented as a weighted summation of i_t and f_t.

$$\begin{aligned}
&\begin{pmatrix} i_t \\ f_t \end{pmatrix} = \begin{pmatrix} sigm \\ sigm \end{pmatrix} \boldsymbol{x}_t \begin{pmatrix} \boldsymbol{W}_{i,x} \boldsymbol{W}_{i,h}^T \\ \boldsymbol{W}_{f,x} \boldsymbol{W}_{f,h}^T \end{pmatrix} \boldsymbol{h}_{t-1}^T \\
&g_t = \boldsymbol{x}_t (a_i \boldsymbol{W}_{i,x} \boldsymbol{W}_{i,h}^T + a_f \boldsymbol{W}_{f,x} \boldsymbol{W}_{f,h}^T) \boldsymbol{h}_{t-1}^T
\end{aligned} \tag{4}$$

Compared with the vanilla LSTM unit in Eq.1, the number of trainable parameters in Eq.4 reduces by about a half, which makes the optimization procedure faster. From the value of g_t, we can get the saliency of each audio frame to the audio theme.

2.3. Semantic Attention Mechanism

The memory mechanism inside vanilla LSTM structure in Eq.1 can be decomposed into an equivalent representation as Fig.1a. Among all trainable parameters, $\boldsymbol{L}_{M+N,3N}$ are utilized to calculate $\boldsymbol{i}_t$, $\boldsymbol{o}_t$ and $\boldsymbol{f}_t$ to determine the final memory cell $\boldsymbol{c}_t$ and output vector $\boldsymbol{h}_t$. Other parameters $\boldsymbol{L}_{M+N,N}$ can be marked as the local memory inside each LSTM cell. During test time, current input $\boldsymbol{x}_t$ and previous hidden state $\boldsymbol{h}_{t-1}$ are compared with the local memory matrix line by line, these similarities are then connected into an instant memory vector and participate in later calculations. This memory mechanism will lead to the convergence of the local memory to be a subset of the input vector space and result in the overfitting problem.

Bag-of-features (BOF) is popular for indexation and categorization applications because these vector representations can be compared with standard distances, and subsequently, be used by robust classification methods. Inside the LSTM unit, we attempt to reconstruct the memory content $\boldsymbol{c}_{\_}$ inspired by the BOF theory [11][12]. As shown in Fig.1b, we first define the local memory as a codebook $\zeta = \{\boldsymbol{d}_1, \boldsymbol{d}_2, ..., \boldsymbol{d}_N\}$ of N cluster centers of audio frames. Each audio frame $\boldsymbol{x}_t$ is associated to its nearest cluster center $\tilde{\boldsymbol{d}}_t = NN(\boldsymbol{x}_t, \zeta)$. From the idea of VLAD [11], we use the difference $\boldsymbol{x}_t - \tilde{\boldsymbol{d}}_t$ as the distribution of the frame with respect to the cluster center. This representation of instant memory is different from the similarities used in Fig.1a.

However, when we integrate this feature representation and the LSTM unit together, the procedure of computing the nearest cluster center $\tilde{\boldsymbol{d}}_t$ is not differentiable. In order to modify the LSTM unit using BOF theory, we propose to mimic VLAD in the LSTM unit and design a trainable memory cell to store the audio frames with VLAD representations. To construct a memory cell amenable to training via backpropagation, we first replace the non-differentiable $NN(\boldsymbol{x}_t, \zeta)$ with a weighted summation of all cluster centers as Eq.5. When α_t is a one-hot vector, this representation is equivalent to the original VLAD definition.

$$\tilde{\boldsymbol{d}}_t = \sum_{k=1}^{N} \alpha_{k,t} \boldsymbol{d}_k \tag{5}$$

Inspired by the attention theory, the assignment $\alpha_{k,t}$ can be computed using a softmax layer of $\boldsymbol{x}_t$ and $\boldsymbol{h}_{t-1}$, and the memory cell $\boldsymbol{c}_{\_}$ in Eq.1 can be replaced using the difference $\boldsymbol{x}_t - \tilde{\boldsymbol{d}}_t$ as Eq.6. In this LSTM unit, $\boldsymbol{L}_{M+N,N}$ is a biased linear mapping where each row corresponds to the trainable parameters for each cluster k. ζ is the trainable codebook, k is the row index in codebook ζ. $\boldsymbol{W}_x$ is the matrice that transform $\boldsymbol{x}_t$ into the semantic space spanned by the codebook ζ.

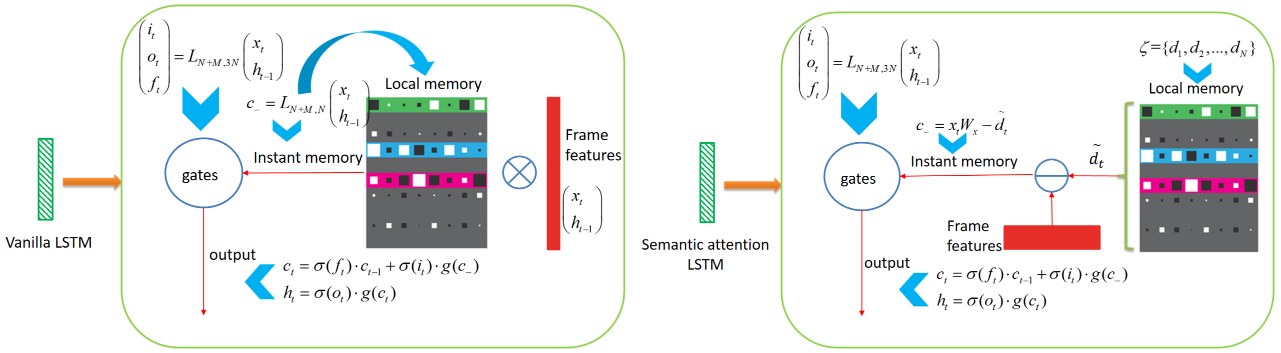
$$\begin{aligned}
&\boldsymbol{r}_t = \boldsymbol{L}_{M+N,N} \begin{pmatrix} \boldsymbol{x}_t \\ \boldsymbol{h}_{t-1} \end{pmatrix} \\
&\alpha_{k,t} = \frac{exp(r_{k,t})}{\sum_{k=1}^{N} exp(r_{k,t})} \\
&\tilde{\boldsymbol{d}}_t = \sum_{k=1}^{N} \alpha_{k,t} \boldsymbol{d}_k \\
&\boldsymbol{c}_{\_} = \boldsymbol{x}_t \boldsymbol{W}_x - \tilde{\boldsymbol{d}}_t
\end{aligned} \tag{6}$$

The number of trainable parameters in Eq.6 keeps pretty much the same with Eq.1, which makes the training process of this new LSTM unit as easy as the vanilla LSTM.

3. Acoustic Scene Classification Experiments

In this section, we employ LITIS ROUEN dataset [13] and DCASE2016 dataset [14] to conduct acoustic scene classification experiments.

Details of these datasets are listed as follows.



(a) *Vanilla LSTM (equivalent to Eq.1)*

(b) *Semantic attention LSTM (as shown in Eq.6)*

Figure 1: *Memory mechanism in LSTM.*

- *LITIS ROUEN dataset*: This is the largest publicly available dataset for ASC to the best of our knowledge. The dataset contains about 1500 minutes of acoustic scene recordings belonging to 19 classes. Each audio recording is divided into 30-second examples without overlapping, thus obtain 3026 examples in total. The sampling frequency of the audio is 22050 Hz. The dataset is provided with 20 training/testing splits. In each split, 80% of the examples are kept for training and the other 20% for testing. We use the mean average accuracy over the 20 splits as the evaluation criterion.
- *DCASE2016 dataset*: The dataset is released as Task 1 of the DCASE2016 challenge. We use the development data in this paper. The development data contains about 585 minutes of acoustic scene recordings belonging to 15 classes. Each audio recording is divided into 30-second examples without overlapping, thus obtain 1170 examples in total. The sampling frequency of the audio is 44100 Hz. The dataset is divided into 4 folds. Our experiments obey this setting, and the average performance will be reported.

3.1. Audio Pre-processing

For both datasets, the audio signal is transformed to frames using Short-time Fourier Transform with a frame length of 1024 and a frameshift of 220, the number of frequency filters is set to be 64. For both datasets, the examples are 30 seconds long. In the data preprocessing step, we first divide the 30-second examples into 1-second clips with 50% overlap. Then each clip is processed using LSTM model. The classification results of all these clips will be averaged to get an ensemble result for the 30-second examples. Some ASC systems benefited from using shorter windows or a higher number of frequency filters, whereas in our case, this configuration is a trade-off between effectiveness and performance.

3.2. Hyper-parameters and Evaluation

In acoustic scene classification tasks, we use the number of LSTM cells as 128, LSTM layers as 1, the learning rate of 0.001, l_2 weight is $1e^{-4}$, training is done using the Adam [15] update method. The outputs of LSTM models are followed by a deep neural network where the network architecture can be summarized as $128 \times 128 \times 19(15)$. The attention size of temporal attention LSTM in Eq.4 is 64, the codebook dimension of semantic attention LSTM in Eq.6 is 128. For DCASE2016 dataset, we set dropout rate as 0.5. All these methods are stopped after 100 training epochs.

In order to compute the results for each training-test split, we use the classification error over all classes. The final classification error is its average value over all splits.

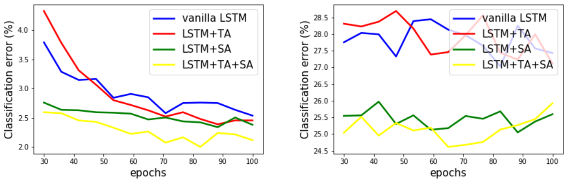
3.3. Results of Attention Mechanisms

Table 1 is the comparison of performance on both datasets after 100 training epochs. On LITIS Rouen dataset, LSTM structure performs much better than other algorithms such as CNN, deep neural network (DNN) and Nonnegative Matrix Factorization (NMF). The proposed temporal and semantic attention LSTM models show improvements when compared with vanilla LSTM. When we integrate these two attention mechanisms, our approach performs significantly better than the state-of-the-art result and obtains a classification error of 2.12%. On DCASE2016 dataset, LSTM is the worst model when compared with CNN, DNN and NMF. However, this phenomenon does not affect our conclusions. We also get consistent performance improvements with both attention mechanisms. As discussed in Section 2, the increase in performance is not at the cost of increasing the number of parameters.

Table 1: *Acoustic scene classification results using different attention mechanisms. TA represents the temporal attention method. SA represents the semantic attention method.*

Model	LITIS Rouen (%)	DCASE2016 (%)
vanilla LSTM	2.54	27.4
LSTM+TA	2.45	26.6
LSTM+SA	2.38	25.6
LSTM+TA+SA	**2.12**	**25.3**
RNN-Gam [1]	3.4	-
CNN-Gam [2]	4.2	-
MFCC-GMM [14]	-	27.5
DNN-CQT [16]	3.4	21.9
Sparse-NMF [16]	5.4	**17.3**
DNN-Mel [17]	-	23.6
CNN-Mel [18]	-	24.0

Validation curves on both datasets are shown in Fig.2. After 100 training epochs, experiments on DCASE2016 dataset encounter severe overfitting problem, experiments on LITIS



(a) *LITIS ROUEN* (b) *DCASE2016*

Figure 2: *Validation curves on both datasets.*

ROUEN dataset has almost converged. For DCASE2016 dataset, the curves show that there is no obvious improvement with temporal attention mechanism, but the results with semantic and integrated attention mechanisms are consistent with Table 1. For LITIS ROUEN dataset, all results are consistent with Table 1.

4. Video Classification Experiments

In this section, we apply the multi-modal attention mechanisms to a large-scale video dataset YouTube-8M [5]. The dataset consists of about 7 million YouTube videos that were annotated with a vocabulary of 4716 tags from 24 diverse categories. The dataset is used in the YouTube-8M video understanding challenge conducted on Kaggle[1]. In the competition, the dataset is divided into three parts. The training set, validation set and test set contains 4.9, 1.4 and 0.7 million samples respectively.

4.1. Video Pre-processing

As described in [5], Google has pre-processed the videos and extract the frame-level image and audio features using state-of-art deep models. They first decode each video at 1 frame-per-second up to the first 360 seconds. Then the decoded image frames are fed into the publicly available Inception network [19] trained on ImageNet [20]. The ReLu activation of the last hidden layer is fetched and followed by the PCA and quantization operation. Finally, each image is converted into a 1024-dimensional feature vector. The audio frames are fed into the network trained in [21], following with the same operations as images. So each audio is converted into a 128-dimensional feature vector. Thus we use the frame-level 1152-dimensional image and audio feature vectors as the input sequences in this task.

4.2. Hyper-parameters and Evaluation

In video classification task, we use the number of LSTM cells as 1024, LSTM layers as 1, learning rate of 0.001, l_2 weight is $1e^{-8}$, training is done using the Adam update method. The outputs of LSTM models are followed by a Mixture-of-Expert [22] classification model where the number of mixtures is 8. The attention size of temporal attention LSTM in Eq.4 is 64, the codebook dimension of semantic attention LSTM in Eq.6 is 1024.

In this task, the performance is evaluated using Global Average Precision (GAP) at 20. This metrics is calculated as follows. For each video, the most confident 20 label predictions are selected along with the confidence values. The tuples of the form $\{video, label, confidence\}$ from all the videos are then put into a long list sorted by confidence values. This list of predictions is then evaluated with the Average Precision (Eq.7), in which $p(i)$ is the precision and $r(i)$ is the recall given the first i predictions.

$$AP = \sum_{i=1}^{N} p(i)\Delta r(i) \tag{7}$$

4.3. Results of Attention Mechanisms

In this section, experiments with temporal and semantic attention mechanisms are carried out separately.

Table 2 shows the results. Vanilla LSTM model described in Sec.2.1 get GAP of 0.8080 on the test part of YouTube-8M dataset, and the best result on the validation set is reached when the LSTM model has been trained for 5 epochs. The proposed temporal and semantic attention LSTM models show significant performance improvements when compared with the vanilla LSTM. Specifically, semantic attention mechanism performs a little better than temporal attention mechanism on both GAP and convergence speed. When these two attention mechanisms are integrated, the GAP metrics reaches 0.8172 and the best result on the validation set is reached only after 3 epochs.

Table 2: *Video classification results using different attention mechanisms.*

Model	Performance		Convergence Speed
	Hit@1	GAP	
vanilla LSTM	0.8631	0.8080	5 epochs
LSTM+TA	0.8668	0.8152	4 epochs
LSTM+SA	0.8687	0.8163	3 epochs
LSTM+TA+SA	**0.8702**	**0.8172**	3 epochs

5. Conclusions

In this work, we propose two different attention mechanisms to modify the vanilla LSTM in sequence classification tasks. The temporal attention method is able to pick out salient frames and produce a better expression for an input sequence. The semantic attention method solves the overfitting problem caused by the memory mechanism in LSTM cells. For acoustic scene classification task, the attention based LSTM structures show consistent performance improvements when compared with vanilla LSTM. On LITIS ROUEN dataset, our approach is able to perform significantly better than the state-of-the-art result, and obtains a relative reduction of 16.5% on classification error. On DCASE2016 dataset, LSTM model is not the best performing structure, but we also get a relative reduction of 7.7% on classification error when compared with vanilla LSTM. As a supplementary experiment, we achieve a performance improvement of 1.1% on YouTube-8M video dataset.

We plan to apply attention based LSTM model to other sequence modeling tasks such as text and speech. And intuitively, we think the position of tokens in the sequence also carries useful sequence information, more investigation will be conducted on this aspect.

6. Acknowledgements

This work is partly funded by National Natural Science Foundation of China (Grant No: 61571266)

[1] visit https://www.kaggle.com/c/youtube8m

7. References

[1] H. Phan, P. Koch, F. Katzberg, M. Maass, R. Mazur, and A. Mertins, "Audio scene classification with deep recurrent neural networks," *arXiv preprint arXiv:1703.04770*, 2017.

[2] H. Phan, L. Hertel, M. Maass, P. Koch, R. Mazur, and A. Mertins, "Improved audio scene classification based on label-tree embeddings and convolutional neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1278–1290, 2017.

[3] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[4] J. Y. Lee and F. Dernoncourt, "Sequential short-text classification with recurrent and convolutional neural networks," *arXiv preprint arXiv:1603.03827*, 2016.

[5] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "Youtube-8m: A large-scale video classification benchmark," *arXiv preprint arXiv:1609.08675*, 2016.

[6] W. Zaremba and I. Sutskever, "Learning to execute," *arXiv preprint arXiv:1410.4615*, 2014.

[7] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[8] S.-s. Shen and H.-y. Lee, "Neural attention models for sequence classification: Analysis and application to key term extraction and dialogue act detection," *arXiv preprint arXiv:1604.00077*, 2016.

[9] M. J. Er, Y. Zhang, N. Wang, and M. Pratama, "Attention pooling-based convolutional neural network for sentence modelling," *Information Sciences*, vol. 373, pp. 388–403, 2016.

[10] H.-D. Wang, T. Zhang, and J. Wu, "The monkeytyping solution to the youtube-8m video understanding challenge," *arXiv preprint arXiv:1706.05150*, 2017.

[11] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 3304–3311.

[12] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5297–5307.

[13] A. Rakotomamonjy and G. Gasso, "Histogram of gradients of time-frequency representations for audio scene classification," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 1, pp. 142–153, 2015.

[14] A. Mesaros, T. Heittola, and T. Virtanen, "Tut database for acoustic scene classification and sound event detection," in *Signal Processing Conference (EUSIPCO), 2016 24th European*. IEEE, 2016, pp. 1128–1132.

[15] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[16] V. Bisot, R. Serizel, S. Essid, and G. Richard, "Feature learning with matrix factorization applied to acoustic scene classification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1216–1229, 2017.

[17] Q. Kong, I. Sobieraj, W. Wang, and M. Plumbley, "Deep neural network baseline for dcase challenge 2016," *Proceedings of DCASE 2016*, 2016.

[18] D. Battaglino, L. Lepauloux, N. Evans, F. Mougins, and F. Biot, "Acoustic scene classification using convolutional neural networks," *DCASE2016 Challenge, Tech. Rep.*, 2016.

[19] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[20] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

[21] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold *et al.*, "Cnn architectures for large-scale audio classification," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 131–135.

[22] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the em algorithm," *Neural computation*, vol. 6, no. 2, pp. 181–214, 1994.