INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Phoneme Embedding and its Application to Speech Driven Talking Avatar Synthesis

Xu Li[1,2], *Zhiyong Wu*[1,2,3], *Helen Meng*[1,3], *Jia Jia*[1,2], *Xiaoyan Lou*[4], *Lianhong Cai*[1,2]

[1] Tsinghua-CUHK Joint Research Center for Media Sciences, Technologies and Systems,
Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China
[2] Tsinghua National Laboratory for Information Science and Technology (TNList),
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
[3] Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong SAR, China
[4] Beijing Samsung Telecom R&D Center, Beijing 100081, China

dongfangyixi@gmail.com, {zywu,hmmeng}@se.cuhk.edu.hk, jjia@tsinghua.edu.cn,
xiaoyan.lou@samsung.com, clh-dcs@tsinghua.edu.cn

Abstract

Word embedding has made great achievements in many natural language processing tasks. However, the attempt to apply word embedding to the field of speech got few breakthroughs. The reason is that word vectors mainly contain semantic and syntactic information. Such high level features are difficult to be directly incorporated in speech related tasks compared to acoustic or phoneme related features. In this paper, we investigate the method for phoneme embedding to generate phoneme vectors carrying acoustic information for speech related tasks. One-hot representations of phoneme labels are fed into embedding layer to generate phoneme vectors that are then passed through bidirectional long short-term memory (BLSTM) recurrent neural network to predict acoustic features. Weights in embedding layer are updated through backpropagation during training. Analyses indicate that phonemes with similar acoustic pronunciations are close to each other in cosine distance in the generated phoneme vector space, and tend to be in the same category after k-means clustering. We evaluate the phoneme embedding by applying the generated phoneme vector into speech driven talking avatar synthesis. Experimental results indicate that adding phoneme vector as features can achieve 10.2% relative improvement in objective test.

Index Terms: Phoneme embedding, talking avatar synthesis, bidirectional long short-term memory (BLSTM)

1. Introduction

Word embedding becomes popular in recent years with its successful applications in nature language processing (NLP) tasks. Word vectors are generated from large unlabeled text data, usually by training a neural network to predict particular words given the context information [1][2]. In this way, word vectors contain abundant semantic and syntactic information that is helpful in NLP tasks as reported in [3].

Several promising studies have been reported to apply word embedding in speech related tasks such as speech synthesis [4][5] and speech recognition [6][7]. However, the results are not as positive as those in NLP tasks. The main reason is the word embedding extracted sematic and syntactic information is difficult to be directly incorporated in speech related tasks.

In this work, we propose phoneme vector and apply it to speech driven talking avatar synthesis task. Different from word embedding, phoneme vectors are generated taking into account the acoustic speech characters representing the pronunciation of phoneme sequences. For phoneme embedding training, the input is phoneme labels, the output is corresponding acoustic features. Concretely, one-hot representations of phoneme labels are fed into embedding layer to generate phoneme vectors as the input of bidirectional long short-term memory (BLSTM) recurrent neural network (RNN) regression model to predict acoustic features. Weights of embedding layer are updated through back-propagation during training. Phoneme embedding analysis indicate that phoneme vector has several interesting characteristics. Phonemes with similar acoustic attributes are close to each other in cosine distance in the generated phoneme vector space, and tend to fall into the same category after k-means clustering.

Talking avatar has been widely used in human-computer interaction fields such as virtual reality, voice agent, computer assistant. Recent research [8] indicates that more representative acoustic features can generate more realistic and expressive visual gestures. In this work, we apply phoneme embedding for speech driven talking avatar with two main outstanding benefits. First, phoneme vectors contain acoustic information such as pronunciation, rhythm or emphatic that are essential in speech related tasks. Second, phoneme vector sequence can represent textual information (e.g. textual hints for lip movement) that are critical and closely correlated to visual gestures. Furthermore, in other speech related tasks such as text-to-speech (TTS) and text-to-visual-speech (TTVS) synthesis, phoneme embedding is very convenient to deal with the out-of-vocabulary (OOV) words that are not available at training time or even not in the dictionary as compared to word embedding.

The rest of this paper is organized as follows. Section 2 describes the datasets and preprocessing procedures. Section 3 introduces the methods to generate phoneme vectors and to synthesize talking avatar. Analysis of phoneme vector is then detailed in Section 4 while talking avatar synthesis experiments are illustrated in Section 5. Finally, Section 6 concludes the work and discusses future work.

http://dx.doi.org/10.21437/Interspeech.2016-363

2. Data description and preprocessing

We adopt bimodal corpus that has been used by several previous studies [8][11]. The corpus contains 700 English utterances recorded by a female native English speaker, including 350 emphatic and 350 neutral utterances. The audio is in Microsoft WAV format with sampling rate of 16 KHz. The video frame rate is 25fps and well formed in AVI format. We divided the whole corpus into 3 parts randomly: 600 utterances as training set, 70 as test set and others as validation set.

35 dimensional Mel-generalized cepstral (Mgc) [12] and 1 dimensional lf0 feature that have been widely used in speech synthesis tasks are extracted as the acoustic features to generate phoneme vectors. In addition, for speech driven talking avatar, 384 dimensional low level descriptors (LLD) [13] are used as the input features to predict facial animation parameters. LLD contains 16 low level descriptors and their first order delta coefficients (32 dimensions in total) and 12 functionals. 16 low level descriptors include RMSE (root mean square signal frame energy), 12-order MFCC (Mel-frequency cepstral coefficients), Zero-crossing rate, Voicing probability and F0. As listed in Table 1, 12 functionals are applied to LLD contours to derive statistical features. The acoustic features are extracted with frame length of 40ms and frame shift of 10ms.

Table 1. 12 functionals applied to LLD contours

Functionals	**No.**
The max/min value of the contour	1~2
Range (max – min)	3
The absolute position of the max/min value	4~5
The arithmetic mean of the contour	6
The slope of a linear approximation of the contour	7
The offset of a linear approximation of the contour	8
The quadratic error computed as the difference of the linear approximation and the actual contour	9
The standard deviation of the values in the contour	10
The skewness and the kurtosis	11~12

As for visual features, facial animation parameters (FAPs) defined by MPEG-4 specification [9][10] are adopted, which include 66 low-level FAPs and 2 high-level FAPs. The low-level FAPs represent a complete set of basic facial actions; while the 2 high-level FAPs represent visemes and expressions. In this work, we focus on the 66 low-level FAPs.

Phoneme sequences are extracted from text in one-hot representation using Carnegie Mellon University Pronouncing Dictionary [14]. For the vowels in CMU dictionary phoneme set, they carry a postfix stress marker with ‘0’ for no stress, ‘1’ for primary stress and ‘2’ for Secondary stress. Totally, we have 66 phonemes, with 65 ones from CMU dictionary and one extra phoneme label for silence.

A well trained HMM model [15] is adopted to conduct frame-to-frame forced alignment between acoustic features and phoneme labels. Linear interpolation is used to interpolate FAPs to match the frame rate of acoustic features.

3. Methods

The overall structure of the system is shown in Figure 1, where three models are involved including (1) phoneme embedding model, (2) phoneme vector predictor, (3) DBLSTM regression model. Phoneme embedding step is illustrated within the red dashed box. Details of the three models are elaborated below.

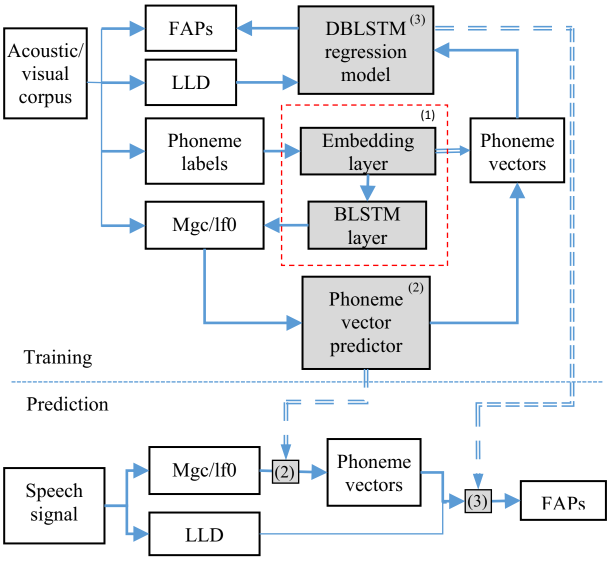

Figure 1: Overall structure of the system. Phoneme embedding step is illustrated within red dashed box

3.1. Phoneme embedding

Word vectors can capture semantic and syntactic information by training a neural network model related to the usage of words and expressions in language. Differ from that, the purpose of phoneme embedding is to capture the acoustic information (e.g. pronunciation characteristics) and represent such information in phoneme vectors. In realizing this, we train a neural network model similar to word embedding process. To capture time sequential context information, we adopt BLSTM RNN [16] for phoneme embedding modeling.

As shown in Figure 1, phoneme embedding model (block 1) is implemented as RNN with two hidden layers, where the first hidden layer is embedding layer and the second hidden layer is BLSTM layer. The input of phoneme embedding model is phoneme labels and the output is Mgc/lf0. After training, phoneme vectors can be retrieved from the embedding layer.

Phoneme label sequences are obtained from text analysis, and are transformed to one-hot representation. Let S be the number of training samples (i.e. speech utterances) in training set, N be the number of frames per each utterance, K be the number of distinct phonemes (66 in our work). The phoneme matrix **M**p with the shape $S\times N\times K$ is multiplied by embedding matrix **M**e with the shape $K\times V$ where V is the dimension of phoneme vector. In this way, we can derive phoneme vector matrix **M**v with the shape $S\times N\times V$. **M**v is then fed into BLSTM layer to derive the correlation map between phoneme vectors and acoustic features (Mgc/lf0). Let D be the dimension of acoustic features. Acoustic matrix **M**a with the shape $S\times N\times D$ is the output of BLSTM model. We use mean square error (MSE) between the predicted acoustic features and the ground truth as the loss function:

$$MSE = \frac{1}{D}\sum_{i=1}^{D}(\hat{Y}_i - Y_i)^2 \ , \tag{1}$$

where Y=**M**a*[s][n]* is the ground truth of acoustic feature at each time expansion step of BLSTM, while Y=H(**M**v*[s][n])* represents the corresponding predictions from BLSTM model. The weight of BLSTM model and embedding matrix **M**e are updated through backpropagation with:

$$\mathbf{M}\mathrm{e}[k][i] = \mathbf{M}\mathrm{e}[k][i] - lr * bperror[i] \ , \qquad (2)$$

where k is the phoneme order, **M**e*[k][i]* is the i-th dimension of phoneme embedding, *bperror[i]* is the backpropagated error of i-th node in **M**v*[s][n]* vector.

After training, each row of embedding matrix **M**e is the phoneme vector of the corresponding phoneme.

3.2. Speech driven talking avatar synthesis

For speech driven avatar synthesis, the input is acoustic features and the output is FAPs. To incorporate phoneme vectors into the task, we need predict phoneme vectors from acoustic input. Hence we train another DBLSTM model as phoneme vector predictor (block 2 in Figure 1) to map Mgc/lf0 to phoneme vectors. Then, DBLSTM regression model (block 3 in Figure 1) is trained which accepts LLD and phoneme vectors as the input to predict FAPs. Forced alignment between acoustic features (i.e. LLD, Mgc/lf0) and phoneme labels is realized by virtue of a well-trained HMM model. The regression models are trained by minimizing cross-entropy error between the output of DBLSTM regression model and the ground truth.

In the prediction stage, phoneme vectors are first predicted from the input Mgc/lf0 features by phoneme vector predictor, which are then concatenated with the LLD features frame-wisely. The concatenated features serve as the input of the DBLSTM regression model to predict FAPs. Finally, visual gestures including lip movements, facial expressions and head motions are rendered on a three-dimensional avatar using the technologies of our previous work [11].

4. Phoneme vector analysis

4.1. Cosine similarity

Cosine similarity is computed between each pair of phoneme vectors according to the following equation:

$$\text{Similarity} = \frac{A \cdot B}{\|A\| \cdot \|B\|} \ , \qquad (3)$$

where A and B denote a pair of phoneme vectors.

The pairs of phonemes having the nearest cosine distance are list in Table 2, from which we can conclude that phonemes with similar pronunciation characteristics got the nearest cosine distance in the phoneme vector space. The results confirm that the proposed phoneme embedding model can capture phoneme pronunciation information within phoneme vectors.

4.2. Phoneme vector clustering

K-means clustering is adopted on the phoneme vectors with k=2, 20. Results are shown in Table 3 and Table 4 respectively.

For k=2, consonants and vowels tend to be in different categories. This confirms that phoneme vector captures the pronunciation characteristics from acoustic features in training. For k=20, we can observe that similar vowel with primary stress label (postfix '1') and no stress label (postfix '0') tend to fall into different categories (Cat. 5 and 6). Such results indicate that phoneme vector embodies not only pronunciation information but also emphasis and rhythm information that are important for most of the speech related tasks. Actually, experimental results with other k settings also indicate that phoneme vectors in the same category demonstrate similar pronunciation characters.

Table 2. *Phoneme pairs with closest cosine distance*

EY0,EY2	CH,T	G,UW2	AY0,AY1
AH2,AW2	F,TH	ER0,EY0	ER1,ER0
AA0,EH1	Y,IY1	OW0,OW2	IY0,IY2
UH1,UW1	DH,V	IT2,IY1	UW1,UH1
ZH,JH	N,M	EH2,EH1	AH0,IH0
EH0,IH1	S,Z	IH0,AH0	M,NG
AA2,AA1	L,OW2	AH1,IH1	AO2,OW2
UH2,AA0	K,P	AE2,AE0	W,OW2

Table 3. *K-means clustering result of phoneme vectors (k=2)*

Category 1	Category 2
'sil', 'N', 'DH', 'S', 'V', 'P', 'D', 'Z', 'K', 'T', 'W', 'B', 'G', 'M','TH', 'HH', 'F', 'SH', 'JH', 'CH', 'ZH'	'AA1', 'AH0', 'EH1', 'R', 'IY0', 'AH1', 'IY1', 'AE1', 'L', 'AY1', 'IH1', 'NG', 'OW1', 'AE0', 'AW1', 'UW1', 'ER1', 'Y', 'ER0', 'AY2', 'AE2', 'AO1', 'IH0', 'OY1', 'IH2', 'AY0', 'EH2', 'AO0', 'EY1', 'EY2', 'UW0', 'OW2', 'AO2', 'IY2', 'OW0', 'UH1', 'UH2', 'AA2', 'EH0', 'UW2', 'AA0', 'AW2', 'AH2', 'EY0'

Table 4. *K-means clustering result of phoneme vectors (k=20), with 6 categories listed*

Cat.1	Cat.2	Cat.3	Cat.4	Cat.5	Cat.6
'sil' 'S' 'D' 'Z' 'TH' 'F'	'N' 'NG' 'M'	'SH' 'JH' 'CH' 'ZH'	'IY0' 'IY1' 'Y' 'IY2'	'AA1' 'AH1' 'AE1' 'OW1' 'AO1'	'AH0' 'AY2' 'AY1' 'AY0' 'EY0' 'EY2' 'IH0'

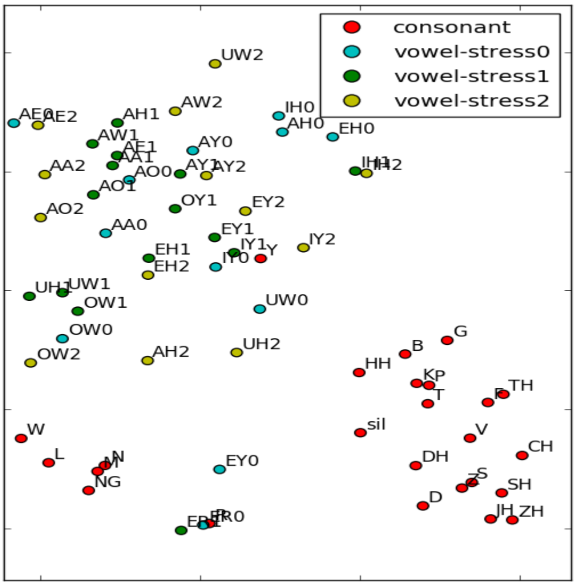

Figure 2: t-SNE visualization of phoneme vectors

4.3. T-SNE visualization

We adopt t-distributed stochastic neighbor embedding (t-SNE) visualization [17] to phoneme vectors. The result is shown in Figure 2. In the figure, the phoneme name is annotated aside the corresponding spots, the color of the spots represents the pronunciation stress of phonemes as presented in legend.

Phoneme vector visualization gives us intuitive impression on phoneme vector's properties. We can figure out from the t-SNE visualization illustration that phoneme vectors tend to be grouped together based on their pronunciations. For example, "ZH", "Z", "S", "SH", "CH", "JH" are close to each other. Most of the consonants are in the lower part of the figure, with the exceptions that "Y" is close to "IY" and "R" is close to "ER". Vowels are in the upper left corner. The phonemes pronounced similar to "AA" with primary stress 1 (green spots) tend to be close to each other when comparing to phonemes with no stress 0 (blue spots) and/or secondary stress 2 (brown spots).

5. Experiments

5.1. Experiment setup

The neural networks are trained using Theano [18] with Adam [19] as optimizer.

For phoneme vector training, we adopt the RNN with two hidden layers and use tanh as the activation for each hidden layer. The first hidden layer is the embedding layer with 128 hidden units. The second hidden layer is the BLSTM layer with 128 forward units and 128 backward units.

For phoneme vector predictor, the input is 36 dimensional Mgc/lf0 and output is the phoneme vector. Three hidden layers are used including two BLSTM layers (with 128 forward and backward units) and one fully connected layer (with 128 units) and use tanh as the activation.

For DBLSTM regression model to predict FAPs, it FAP predict model adopt three hidden layers consist of Two 128 nodes BLSTM layers and a fully connected (FC) layer with 128 nodes, a 68 nodes fully connected layer as the prediction layer. The activation of BLSTM and first FC layer is tanh, linear activation function is adopt in the prediction layer. Besides, hidden layers with one BLSTM layer and one FC layer structure is also tested in the experiment. A dropout layer with rate 0.5 is added before the prediction layer.

To validate the performance of the proposed phoneme vector in speech driven talking avatar synthesis, we conducted a set of experiments comparing LLD features (LLD), one-hot phoneme representation concatenated with LLD features (One-hot vector + LLD) and phoneme vector concatenated with LLD features (Phoneme vector + LLD) as three kinds of input features of the DBLSTM regression model.

5.2. Objective evaluation

We adopt root mean squared error (RMSE) between predicted FAPs and the ground truth as the objective evaluation criterion. The results for different input features are shown in Table 5. To compare the performance of different network architectures of the regression model for FAP prediction, we implement two architectures that are represented in the second column of the table. B-F means one BLSTM layer and one FC layer, B-B-F indicate two BLSTM layers and one FC layer.

From the results, we can see that, for the same architecture of DBLSTM regression model, method with phoneme vector shows superior performance in RMSE, and achieve 10.2% relative improvement compared to the B-B-F architecture with LLD as input.

Table 5. *Results of RMSE with different input features and model architectures*

Input features	Model architecture	RMSE
LLD	B-F	420.93
	B-B-F	336.25
One-hot vector + LLD	B-F	380.40
	B-B-F	314.22
Phoneme vector + LLD	B-F	344.54
	B-B-F	**301.85**

Table 6. *Results of RMSE with different phoneme vector dimensions*

Phoneme vector dimension	RMSE
32	307.33
64	302.01
128	301.85
256	304.62

5.3. Phoneme vector dimension experiment

To find the most proper dimension of phoneme vector, we further conduct experiments comparing RMSE with different settings of phoneme vector dimensions. By setting the number of nodes in phoneme embedding layer (i.e. the column number of embedding matrix *Me*) to 32, 64, 128, 256, the results are shown in Table 6.

As can be seen, the model with 128 dimensional phoneme vector achieves the lowest RMSE. However, the RMSE values are not statistically different between different dimension settings. We can't confirm that 128 dimensional phoneme vector is the best choice. But we can discover that all dimension settings of phoneme vector achieve lower RMSE than one-hot representation (Table 5). Such results confirm that phoneme vectors provide better representative information for the speech driven talking avatar task.

6. Conclusion and future work

In this paper, phoneme embedding is proposed and applied in speech driven talking avatar synthesis. Phoneme vectors are extracted from the neural network that models the relationship between phoneme labels and acoustic features. Phoneme vector sequences contain contextual acoustic and textual information. We adopt phoneme vector to the task of speech driven talking avatar synthesis and achieve the positive result in experiments. In the future, we plan to investigate the use of phoneme vector in end-to-end speech recognition or text-to-speech synthesis.

7. Acknowledgement

This work is supported by National Basic Research Program of China (2012CB316401), National High Technology Research and Development Program of China (2015AA016305), Natural Science Foundation of China (NSFC) (61375027, 61433018, 61370023 and 61171116), joint fund of NSFC-RGC (Research Grant Council of Hong Kong) (61531166002, N_CUHK404/15) and Major Program for National Social Science Foundation of China (13&ZD189).

8. References

[1] Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin, "Neural probabilistic language models," *Innovations in Machine Learning*, pp. 1137–1155. Springer, 2003.

[2] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky, S. Khudanpur, "Recurrent neural network based language model," in *Proc. INTERSPEECH*, pp. 1045–1048, 2010.

[3] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, "Natural language processing (almost) from scratch," *The Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[4] O. Watts, S. Gangireddy, J. Yamagishi, S. King, S. Renals, A. Stan, M. Giurgiu, "Neural net word representations for phrase-break prediction without a part of speech tagger," in *Proc. ICASSP*, pp. 2599–2603, 2014.

[5] H. Lu, S. King, O. Watts, "Combining a vector space representation of linguistic context with a deep neural network for text-to-speech synthesis," in Proc. Speech Synthesis Workshop, pp. 281–285, 2013.

[6] T. He, X. Xiang, Y. Qian, K. Yu, "Recurrent neural network language model with structured word embedding for recognition," in *Proc. ICASSP*, 2015.

[7] S. Bengio, G. Heigold, "Word embedding for speech recognition," in *Proc. INTERSPEECH*, 2014

[8] X. Lan, X. Li, Y. Ning, Z. Wu, H. Meng, J. Jia, L. Cai, "Low level descriptors based DBLSTM bottleneck feature for speech driven talking avatar," in *Proc. ICASSP*, 2016.

[9] Z. Wu, S. Zhang, L. Cai, H. Meng, "Real-time synthesis of Chinese visual speech and facial expressions using MPEG-4 FAP features in a three-dimensional avatar," in *Proc. ICSLP*, pp. 1802-1805, 2006.

[10] J. Jia, S. Zhang, H. Meng, Y. Wang, L. Cai, "Emotional audio-visual speech synthesis based on PAD," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, pp. 570-582, 2011.

[11] J. Jia, Z. Wu, S. Zhang, H. Meng, L. Cai, "Head and facial gestures synthesis using PAD model for an expressive talking avatar," *Multimedia Tools and Applications*, vol. 73, pp. 439-461, 2014.

[12] K. Tokuda, T. Kobayashi, T. Masuko, S. Imai, "Mel-generalized cepstral analysis – a unified approach to speech spectral estimation," in *Proc. ICSLP*, 1994.

[13] B. Schuller, S. Steidl, A. Batliner, "The INTERSPEECH 2009 emotion challenge," in *Proc. INTERSPEECH*, pp. 312-315, 2009.

[14] The Carnegie Mellon University Pronouncing Dictionary, http://www.speech.cs.cmu.edu/cgi-bin/cmudict#lookup.

[15] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. ICASSP*, pp. 1315–1318, 2000.

[16] S. Hochreiter, J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, 1997.

[17] Van der Maaten, L.J.P.; Hinton, G.E. "Visualizing High-Dimensional Data Using t-SNE," Journal of Learning Research 9: 2579-2605, Nov 2008.

[18] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, Y. Bengio. "Theano: A CPU and GPU Math Expression Compiler," in Proc. SciPy, 2010.

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