INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Deep Neural Factorization for Speech Recognition

Jen-Tzung Chien, Chen Shen

Department of Electrical and Computer Engineering, National Chiao Tung University, Taiwan

Abstract

Conventional speech recognition system is constructed by unfolding the spectral-temporal input matrices into one-way vectors and using these vectors to estimate the affine parameters of neural network according to the vector-based error backpropagation algorithm. System performance is constrained because the contextual correlations in frequency and time horizons are disregarded and the spectral and temporal factors are excluded. This paper proposes a spectral-temporal factorized neural network (STFNN) to tackle this weakness. The spectral-temporal structure is preserved and factorized in hidden layers through two ways of factor matrices which are trained by using the factorized error backpropagation. Affine transformation in standard neural network is generalized to the spectro-temporal factorization in STFNN. The structural features or patterns are extracted and forwarded towards the softmax outputs. A deep neural factorization is built by cascading a number of factorization layers with fully-connected layers for speech recognition. An orthogonal constraint is imposed in factor matrices for redundancy reduction. Experimental results show the merit of integrating the factorized features in deep feedforward and recurrent neural networks for speech recognition.

Index Terms: Spectro-temporal factorization, deep neural network, factorized error backpropagation, speech recognition

1. Introduction

Deep learning and matrix factorization have been rapidly growing in the areas of machine learning and signal processing with different applications ranging from speech recognition [1, 2] to computer vision, source separation [3, 4], music information retrieval [5] and natural language processing. Many extensions and realizations have been developing to discover the meanings and insights so as to improve the system performance from different perspectives. Matrix factorization for two-way decomposition can be also generalized to the tensor factorization for multiple-way observations [6]. In the literature, several works have been proposed to strengthen the modeling capability by integrating deep neural network and tensor factorization. In [7], a factored three-way restricted Boltzmann machine allowed the factors or states of hidden units to modulate the pairwise interactions with visible units in natural images. In [8, 9], a deep tensor neural network was constructed by cascading the double projection layer and the tensor layer so as to learn the complimentary features from two hidden vectors. In previous methods, the multi-way tensor weights were estimated to capture the relations between features or neurons. The inputs were still one-way vectors. In [10], the convolutional NN (CNN) [11] was developed to extract the spatial features through convolution layer followed by pooling layer where no factorization was performed. In [12, 13], a tensor classification network was proposed for image recognition where the multi-way inputs were factorized and fed into a classification neural network.

This study aims to extract the *factorized* spectro-temporal features from a speech signal and relax the limitation of using vector-based inputs in deep neural network based speech recognition. To improve the deep learning representation for speech recognition, we are motivated to retrieve the contextual features from both frequency and time domains through spectro-temporal factorization in a layer-wise neural network. From multi-way speech observations, we retrieve the structural information and fulfill the factorized representation in a deep neural factorization model. The factorization scheme is tightly merged into construction of a classification neural network for speech recognition. Importantly, we develop a matrix factorized error backpropagation algorithm with orthogonal constraint to train the regularized factor matrices in different ways and different layers by using the spectro-temporal input matrices from continuous speech. Due to the factorization in forward calculation, the gradients for minimization of cross entropy error function in backward calculation are obtained by transpose factorization. This algorithm is fulfilled to build a deep model by constructing the factorization layers followed by the fully-connected layers. An orthogonal regularization method is further introduced to reduce the redundancy in the layer-wise factorization parameters. Such a solution is generalizable to take multiple modalities and channels into account and can be realized for recurrent neural network based on long short-term memory. The performance of the proposed spectro-temporal neural factorization is investigated by noisy speech recognition using Aurora4 task.

2. Background Survey

We propose a spectro-temporal factorized neural network which seamlessly integrates a spectro-temporal transformation into a layer-wise neural network.

2.1. Spectro-temporal factorization

Spectro-temporal factorization is seen as a matrix factorization or a *bilinear transformation* which is a two-way realization of tensor factorization. According to the Tucker decomposition, a spectro-temporal matrix $\mathbf{X} = \{X_{ft}\} \in \mathcal{R}^{F \times T}$ with F frequency bins and T time frames is factorized to obtain a core matrix $\mathbf{A} = \{A_{lm}\} \in \mathcal{R}^{L \times M}$ by

$$\mathbf{X} = \mathbf{A} \times_1 \mathbf{U} \times_2 \mathbf{V} \qquad (1)$$

where $\times_n$ denotes the model-n product and $\mathbf{U} = \{U_{fl}\} \in \mathcal{R}^{F \times L}$ and $\mathbf{V} = \{V_{tm}\} \in \mathcal{R}^{T \times M}$ denote the factor matrices in two horizons. Each entry in core matrix is expressed by

$$X_{ft} = \sum_l \sum_m A_{lm} U_{fl} V_{tm}. \qquad (2)$$

This decomposition was solved by using the multi-linear singular value decomposition [6]. It is important that the *inverse* of Tucker decomposition in Eq. (1) can be obtained by

$$\mathbf{A} = \mathbf{X} \times_1 \mathbf{U}^{\dagger} \times_2 \mathbf{V}^{\dagger} \qquad (3)$$

http://dx.doi.org/10.21437/Interspeech.2017-892

where $\mathbf{U}^{\dagger} = (\mathbf{U}^{\top}\mathbf{U})^{-1}\mathbf{U}^{\top}$ is the pseudo-inverse of $\mathbf{U}$. The core matrix $\mathbf{A}$ is viewed as a *factorized feature* matrix of the spectro-temporal matrix $\mathbf{X}$ if dimensions L and M are smaller than F and T, respectively. This study adopts this property to construct a new factorized neural network topology based on spectro-temporal factorization.

2.2. Layer-wise neural network

In general, there are two crucial calculations in traditional vector-based neural network model. The first one is the affine transformation $\mathbf{U}\mathbf{x}$ and nonlinear activation $h(\cdot)$ in different layers in the forward pass. The hidden vector $\mathbf{z}$ corresponding to an input vector $\mathbf{x}$ is calculated by $\mathbf{z} = h(\mathbf{U}\mathbf{x})$ using the weight matrix $\mathbf{U}$ in affine transformation where a bias vector has been merged. The second one is the estimation of neural network parameters $\mathbf{U}$ in different layers in backward pass according to the vector-based error backpropagation algorithm by minimizing the cross-entropy error function for a classification network. The performance of feedforward neural network can be improved by constructing a deep model with many hidden layers. Such a network can be extended to build the recurrent neural network (RNN) where hidden units of previous time frame are augmented in input neurons. Temporal information can be accordingly captured. However, RNN suffers from the problem of gradient vanishing and exploding. Long short-term memory network [14] was proposed to deal with this problem. In speech recognition, each frame was formed by a spectro-temporal matrix $\mathbf{X}$ using F dimensional spectral vectors within a time window of T frames. This matrix was flattened into a one-way vector and fed into traditional neural networks for training and prediction. The spatial information in spectro-temporal matrix was partially disregarded so that the performance of speech recognition was constrained. CNN [11] was developed to catch spatial features by means of convolution layers and pooling layers. Although contextual information was characterized, there was no factorized features extracted to represent latent factors in different ways. The factorized information was missing in learning representation.

3. Spectro-temporal factorized neural network

This paper presents a deep neural factorization for speech recognition where spectro-temporal factorization is performed in a deep neural network.

3.1. Deep neural factorization

Given a spectro-temporal input matrix $\mathbf{X} = \{X_{ft}\}$, also known as the two-way tensor, the latent matrix $\mathbf{A}^{(1)} = \{A^{(1)}_{lm}\}$ in the first hidden layer is obtained by the factorization through two factor matrices $\mathbf{U} = \{U_{lf}\} \in \mathcal{R}^{L\times F}$ and $\mathbf{V} = \{V_{mt}\} \in \mathcal{R}^{M\times T}$ via

$$\mathbf{A}^{(1)} = \mathbf{X} \times_1 \mathbf{U} \times_2 \mathbf{V} = \sum_f \sum_t X_{ft}(\mathbf{u}_f \circ \mathbf{v}_t) \tag{4}$$

which is seen as a summation of outer products of all individual columns of factor matrices $\mathbf{u}_f$ and $\mathbf{v}_t$ along frequency and time horizons, respectively. After this bilinear transformation, we obtain a spectro-temporal feature matrix $\mathbf{Z}^{(1)} = \{Z^{(1)}_{lm}\} \in \mathcal{R}^{L\times M}$ through an activation function, i.e. $\mathbf{Z}^{(1)} = h(\mathbf{A}^{(1)})$. This feature matrix is then factorized and activated again as $\mathbf{A}^{(2)}$ and $\mathbf{Z}^{(2)}$ in the second hidden layer, respectively. The feedforward computation is run layer by layer. For speech recognition, a softmax function $s(\cdot)$ is used to calculate the posterior outputs $\mathbf{y} = \{y_k\}$ for K senones $\mathbf{y} = s(\mathbf{a}^{(l)}) = \frac{\exp(\mathbf{a}^{(l)})}{\sum_k \exp(a^{(l)}_k)}$ where $a^{(l)}_k = \langle \boldsymbol{\mathcal{W}}_{::k}, \mathbf{Z}^{(l-1)} \rangle$ is an entry of vector $\mathbf{a}^{(l)} \in \mathbb{R}^{K\times 1}$. Here, $a^{(l)}_k$ is calculated by the inner product of the output matrix $\mathbf{Z}^{(l-1)}$ of hidden units in previous layer $l-1$ and the parameter matrix $\boldsymbol{\mathcal{W}}_{::k}$. Matrix $\boldsymbol{\mathcal{W}}_{::k}$ is a partition of the parameter tensor $\boldsymbol{\mathcal{W}}$ which is only connected to the k-th output neuron. The cross-entropy error function is calculated from a set of N spectro-temporal matrices and their class targets $\{\mathbf{X}_n, \mathbf{r}_n\}_{n=1}^N$ via

$$E(\boldsymbol{\Theta}) = \sum_n E_n(\boldsymbol{\Theta}) = -\sum_n \sum_k r_{nk} \ln y_{nk} \tag{5}$$

where $\mathbf{r}_n = \{r_{nk}\}$ is the target vector of input matrix $\mathbf{X}_n$ obtained by 1-of-K coding scheme. $\mathbf{y}_n = \{y_{nk}\}$ is the posterior outputs of $\mathbf{X}_n$.

3.2. Factorized error backpropagation

To estimate model parameters $\boldsymbol{\Theta} = \{\mathbf{U}, \mathbf{V}, \boldsymbol{\mathcal{W}}\}$ of spectro-temporal factorized neural network (STFNN), we perform the stochastic gradient descent (SGD) algorithm by calculating the gradients of E_n with respect to individual parameters in $\boldsymbol{\Theta}$. Starting from the softmax layer, we calculate the local gradient $d^{(l)}_k$ of an output neuron k and then find the gradients $\frac{\partial E_n}{\partial \mathcal{W}_{lmk}}$ for updating three-way parameter tensor $\boldsymbol{\mathcal{W}} = \{\mathcal{W}_{lmk}\} \in \mathcal{R}^{L\times M\times K}$

$$\begin{aligned} \frac{\partial E_n}{\partial a^{(l)}_k} &= \sum_c \frac{\partial E_n}{\partial y_{nc}} \frac{\partial y_{nc}}{\partial a^{(l)}_k} = y_{nk} - r_{nk} \triangleq d^{(l)}_k \\ \frac{\partial E_n}{\partial \mathcal{W}_{lmk}} &= \frac{\partial E_n}{\partial a^{(l)}_k} \frac{\partial a^{(l)}_k}{\partial \mathcal{W}_{lmk}} = d^{(l)}_k Z^{(l-1)}_{lm}. \end{aligned} \tag{6}$$

Here, we assume $\mathbf{Z}^{(l-1)} = \{Z^{(l-1)}_{lm}\} \in \mathcal{R}^{L\times M}$. These gradients can be expressed by $\nabla_{\boldsymbol{\mathcal{W}}_{::k}} E_n = d^{(l)}_k \mathbf{Z}^{(l-1)}$. After updating the parameter tensor $\boldsymbol{\mathcal{W}}$, we propagate the local gradients from $\mathbf{d}^{(l)} = \{d^{(l)}_k\}$ in layer l back to $\boldsymbol{\mathcal{D}}^{(l-1)} = \{\mathcal{D}^{(l-1)}_{lm}\}$ in layer $l-1$ by

$$\frac{\partial E_n}{\partial A^{(l-1)}_{lm}} = \sum_k \frac{\partial E_n}{\partial a^{(l)}_k} \frac{\partial a^{(l)}_k}{\partial Z^{(l-1)}_{lm}} \frac{\partial Z^{(l-1)}_{lm}}{\partial A^{(l-1)}_{lm}} \triangleq \mathcal{D}^{(l-1)}_{lm}. \tag{7}$$

Eq. (7) is also written in matrix form as $\boldsymbol{\mathcal{D}}^{(l-1)} = h'(\mathbf{A}^{(l-1)}) \odot (\boldsymbol{\mathcal{W}} \times_3 \mathbf{d}^{(l)})$ where $\odot$ denotes the element-wise product.

Having the local gradient matrix $\boldsymbol{\mathcal{D}}^{(l-1)}$, we can calculate the derivatives of E_n with respect to individual entries in two factor matrices U_{lf} and V_{mt} by

$$\begin{aligned} \frac{\partial E_n}{\partial U_{lf}} &= \sum_m \frac{\partial E_n}{\partial A^{(l-1)}_{lm}} \frac{\partial A^{(l-1)}_{lm}}{\partial U_{lf}} = \sum_m \mathcal{D}^{(l-1)}_{lm} \sum_t Z^{(l-2)}_{ft} V_{mt} \\ &= \langle \boldsymbol{\mathcal{D}}^{(l-1)}_{l:}, \mathbf{Z}^{(l-2)}_{f:} \times_2 \mathbf{V} \rangle \\ \frac{\partial E_n}{\partial V_{mt}} &= \langle \boldsymbol{\mathcal{D}}^{(l-1)}_{:m}, \mathbf{Z}^{(l-2)}_{:t} \times_1 \mathbf{U} \rangle \end{aligned} \tag{8}$$

where the latent matrix $\mathbf{Z}^{(l-2)} \in \mathcal{R}^{F\times T}$ in layer $l-2$ is assumed. After updating the STFNN parameters $\{\mathbf{U}, \mathbf{V}\}$, the local gradients $\boldsymbol{\mathcal{D}}^{(l-1)}$ in layer $l-1$ are further propagated back to $\boldsymbol{\mathcal{D}}^{(l-2)} = \{\mathcal{D}_{ft}^{(l-2)}\}$ in layer $l-2$ by

$$\frac{\partial E_n}{\partial A_{ft}^{(l-2)}} = h'(A_{ft}^{(l-2)}) \sum_l \sum_m \mathcal{D}_{lm}^{(l-1)} U_{lf} V_{mt} \triangleq \mathcal{D}_{ft}^{(l-2)} \tag{9}$$

or $\boldsymbol{\mathcal{D}}^{(l-2)} = h'(\mathbf{A}^{(l-2)}) \odot \left(\boldsymbol{\mathcal{D}}^{(l-1)} \times_1 \mathbf{U}^\top \times_2 \mathbf{V}^\top\right)$. These local gradients will be used to find the derivatives for updating the factor matrices in layer $l-2$. Notably, the local gradient $\boldsymbol{\mathcal{D}}^{(l-2)}$ in layer $l-2$ is calculated in a form of *transpose factorization* by using $\{\boldsymbol{\mathcal{D}}^{(l-1)}, \mathbf{U}^\top, \mathbf{V}^\top\}$ in layer $l-1$.

3.3. Orthogonal regularization

This paper further investigates the effect of regularization in estimation of STFNN parameters. We aim to reduce the redundancy in factorization or projection, or equivalently the correlation in the extracted features. An orthogonal constraint [15] is imposed in cross-entropy training. The training objective $E_n(\boldsymbol{\Theta})$ in Eq. (5) is modified by adding two penalty terms $E_p(\mathbf{U})$ and $E_p(\mathbf{V})$

$$E_n(\boldsymbol{\Theta}) + \lambda \underbrace{\left(\sum_{f=1}^{L} \sum_{l=f+1}^{L} \frac{|\langle \mathbf{u}_f, \mathbf{u}_l \rangle|}{\|\mathbf{u}_f\| \|\mathbf{u}_l\|} + \sum_{t=1}^{M} \sum_{m=t+1}^{M} \frac{|\langle \mathbf{v}_t, \mathbf{v}_m \rangle|}{\|\mathbf{v}_t\| \|\mathbf{v}_m\|} \right)}_{E_p(\mathbf{U}) + E_p(\mathbf{V})} \tag{10}$$

where λ is a regularization parameter and $\{\mathbf{u}_f, \mathbf{u}_l\}$ and $\{\mathbf{v}_t, \mathbf{v}_m\}$ denote two different columns in factor matrices $\mathbf{U}$ and $\mathbf{V}$, respectively. In the implementation, we need to calculate the gradient for individual regularization term, e.g.

$$\frac{\partial E_p(\mathbf{U})}{\partial \mathbf{u}_f} = \sum_{l=1}^{L} \frac{|\langle \mathbf{u}_f, \mathbf{u}_l \rangle|}{\|\mathbf{u}_f\| \|\mathbf{u}_l\|} \left[\frac{\mathbf{u}_l}{\langle \mathbf{u}_f, \mathbf{u}_l \rangle} - \frac{\mathbf{u}_f}{\langle \mathbf{u}_f, \mathbf{u}_f \rangle} \right]. \tag{11}$$

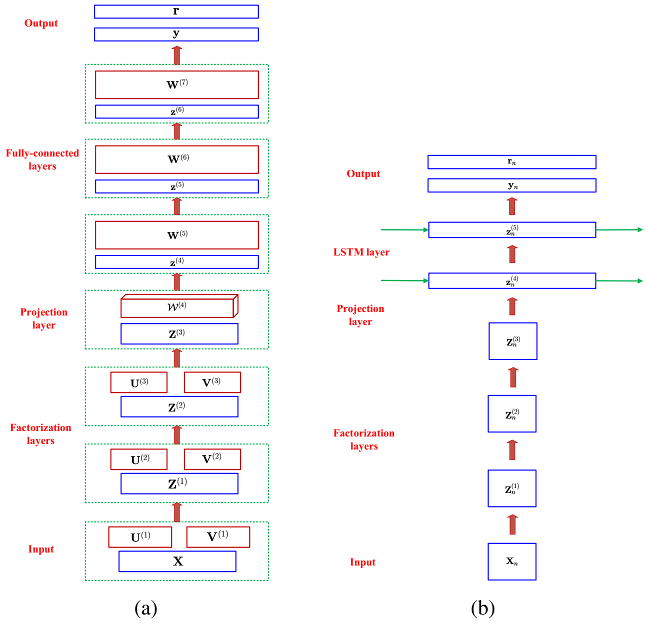


Figure 1: *Deep learning for STFNN with (a) fully-connected layers and (b) long short-term memory layer.*

3.4. Deep and recurrent neural networks

We implement two realizations of deep model for STFNN. In the first realization, as shown by Figure 1(a), three layers of factor matrices $\{\mathbf{U}^{(1)}, \mathbf{V}^{(1)}, \mathbf{U}^{(2)}, \mathbf{V}^{(2)}, \mathbf{U}^{(3)}, \mathbf{V}^{(3)}\}$ are used to factorize the input and feature matrices $\{\mathbf{X}, \mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}\}$, respectively. A projection tensor $\mathcal{W}^{(4)}$ is then arranged to project the deep feature matrix $\mathbf{Z}^{(3)}$ into a vector $\mathbf{z}^{(4)}$. This vector is then propagated along a number of fully connected (FC) layers using weight matrices $\{\mathbf{W}^{(5)}, \mathbf{W}^{(6)}, \mathbf{W}^{(7)}\}$ to obtain the posterior outputs for individuals senones or classes $\mathbf{y}$. Using this STFNN-FC model, the *front* layers are used to extract two-way features while the *back* layers are designed to classify these spatial features via a deep model.

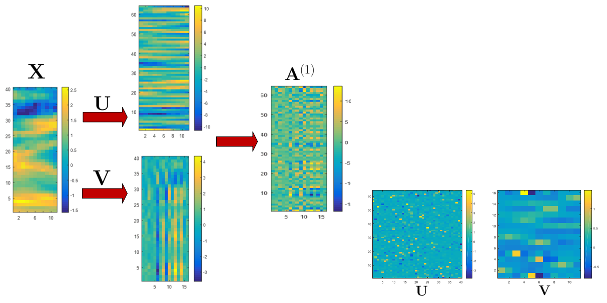
In the second realization, as shown by Figure 1(b), we develop a factorized recurrent neural network based on STFNN where the long short-term memory (LSTM) [14] is applied. In this implementation, the layer-wise factorized features $\{\mathbf{Z}_n^{(1)}, \mathbf{Z}_n^{(2)}, \mathbf{Z}_n^{(3)}\}$ are extracted through the factorization layers for each input matrix $\mathbf{X}_n$ at time frame n by using factor matrices $\{\mathbf{U}^{(1)}, \mathbf{V}^{(1)}, \mathbf{U}^{(2)}, \mathbf{V}^{(2)}, \mathbf{U}^{(3)}, \mathbf{V}^{(3)}\}$. We then project the feature matrix $\mathbf{Z}_n^{(3)}$ into a vector $\mathbf{z}_n^{(4)}$ using a 3-way parameter tensor $\boldsymbol{\mathcal{W}}$ and treat this deep feature vector as a pseudo-input to feed into the LSTM layer to find latent state $\mathbf{z}_n^{(5)}$ before calculating the posterior outputs $\mathbf{y}_n$ for speech recognition. Cross entropy error function between posterior outputs $\{\mathbf{y}_n\}$ and targets $\{\mathbf{r}_n\}$ is calculated during optimization procedure. The temporal information of the estimated spectro-temporal features can be captured in the resulting STFNN-LSTM model.

4. Experiments

4.1. Experimental setup

In the experiments, we evaluated the proposed method by using Aurora4 which was extended from the Wall Street Journal (WSJ0) corpus with noise contamination under different noise type and signal-to-noise ratio. The training set contained 7137 utterances from 83 speakers with 14 hours. The evaluation set had 4620 utterances (330 utterances × 14 test sets) from 8 speakers, with 40.19 minutes of speech data, sampled from 5K-word closed vocabulary based on the WSJ0 NOV-92 corpus. The 14 test sets were grouped into four conditions: A (clean data), B (noisy data), C (clean data with channel distortion), and D (noisy data with channel distortion). The individual and averaged WERs (%) over different conditions were reported. An additional validation set was used.

The implementation was performed by using Kaldi Toolkit [16] followed by the recipe shown in [17]. An initial Gaussian mixture model - hidden Markov model (GMM-HMM) system was first realized with the given phone set, the HMM topology and the tying of context-dependent states where a decision tree was used. The model structure of deep neural network (DNN)-HMM system was constructed from the class labels generated from the forced alignment based on the speaker adaptive training using GMM-HMM. For deep learning, the increasing number of layers made the model more complex and led to the gradient vanishing problem. System parameters were hard to converge. The greedy layer-wise pre-training was applied to assure stable parameters. The input matrix at each frame n consisted of a context window of 11 frames (± 5 frames) of 40-dimensional log Mel frequency bank features, namely $\mathbf{X}_n \in \mathcal{R}^{40 \times 11}$. There were 2035 output units for individual triphone tied states. We adopted the SGD training by using the mini-batch with 256 frames. Adam algorithm [18] was performed in SGD training In implementation of STFNN, the orthogonal constraint was only imposed for the factor matrices in factorization layers. Orthogonal regularization was compared with ℓ_2 regularization where regularization parameter λ was selected from validation data. No sparsity constraint was applied. The topologies of DNN, STFNN, STFNN-FC and STFNN-LSTM with different settings were evaluated. Single-layer LSTM with 1024 memory cells was implemented by referring [19]. Weights in LSTM initialized with an uniform distribution. We delayed the LSTM outputs by five frames to help making the decisions for the current frame [20]. Spectro-temporal information was analyzed [21].



(a)

(b)

Figure 2: *(a) Spectro-temporal factorization in STFNN. (b) Estimated factor matrices* $\{\mathbf{U}, \mathbf{V}\}$ *in two ways.*

4.2. Experimental results

Figure 2(a) illustrates an example about how a spectro-temporal input matrix of an Aurora4 speech frame $\mathbf{X}$ using STFNN is factorized into a two-way features matrix $\mathbf{A}^{(1)}$. Figure 2(b) displays the values of the estimated factor matrices $\mathbf{U}$ and $\mathbf{V}$ in frequency and time horizons in the first hidden layer, respectively. We can see that $\mathbf{X}$ transformed by using the estimated $\mathbf{U}$ and $\mathbf{V}$ reflects the characteristics of spectral (row) and temporal (column) features, respectively. These features are behaviored as the dictionaries in spectral and temporal domains. The transformed matrix $\mathbf{A}^{(1)}$ is seen as a set of two-dimensional atoms or features corresponding to $\mathbf{X}$. The estimated weight parameters $\mathbf{U}$ and $\mathbf{V}$ are seen as the *sparse* filters for two horizons. The effect of spectro-temporal factorization in STFNN is obvious. The factorized information is helpful for learning representation which is employed in speech recognition.

Table 1: *Comparison of WERs (%) and number of parameters (in millions) by using different models.*

Model	Conditions				Avg.	#Par.
	A	B	C	D		
GMM	7.29	12.97	12.61	27.66	18.83	–
DNN5	3.46	7.98	10.78	22.09	13.90	21.1
LSTM3	4.37	8.33	9.66	20.42	13.32	36.3
CNN	4.15	8.13	10.98	19.97	13.12	23.9
STFNN5	4.42	10.39	11.03	23.58	15.66	3.9
STFNN2-FC3	4.09	7.72	10.80	19.61	12.78	18.8
STFNN2$^{\perp}$-FC3	4.11	7.62	10.87	18.57	12.29	18.8
STFNN3-LSTM	3.79	7.02	10.71	18.41	11.93	17.9
STFNN3$^{\perp}$-LSTM	3.70	6.88	10.68	18.05	11.71	17.9

Table 1 shows the comparison of WER and number of parameters by using GMM, DNN, LSTM, CNN, STFNN, STFNN-FC and STFNN-LSTM which contain different topologies. ℓ_2 regularization is applied in different models. DNN5 means DNN with 5 hidden layers. Each hidden layer has 1024 units. LSTM3 means 3 stacks of LSTM where each stack contains 1024 cells [22]. Experimental setup of CNN is referred to [10]. STFNN5 means STFNN with 5 matrix factorized (MF) layers without fully connected (FC) layers. In MF layers, we empirically use the hidden matrix size as 30×8. STFNN2-FC3 means 2 MF layers followed by 3 FC layers (the same as Figure 2(a)). Two realizations are investigated. STFNN2$^{\perp}$-FC3 indicate the realization with orthogonal regularization. The size of hidden matrices and hidden vectors in STFNN-FC and STFNN-LSTM is the same as that of DNN and STFNN. STFNN3-LSTM denotes 3 MF layers followed by LSTM layer (the same as Figure 2(b)). STFNN3-LSTM denotes 3 MF layers followed by LSTM layer. STFNN3$^{\perp}$-LSTM denotes the same model with orthogonal regularization.

In the experiments, DNN performs much better than GMM for speech recognition while LSTM and CNN can further reduce WERs. Deep model using STFNN5 does not work better than that using DNN5, LSTM3 and CNN. It is because that FC layers in STFNN-FC are required to reliably estimate the class posteriors. This concept is similar to the idea of CNN connecting with FC layers before calculating the class posteriors. However, the parameter size of STFNN5 is dramatically reduced because of factorization in different layers. This shows the pros and cons of STFNN when balancing the trade-off between system performance and model compactness. By using STFNN2-FC3, the resulting WER is smaller that using DNN5, LSTM3 and CNN. Orthogonal regularization works slightly better than ℓ_2 regularization in different realizations of STFNN2-FC3 and STFNN3-LSTM. The best result is achieved by STFNN3$^{\perp}$-LSTM with the parameter size which is even smaller than DNN5. In this realization, STFNN encodes the common features from spectro-temporal factorization and applies them in decoding by using LSTM. Such a encoding/decoding procedure provides a meaningful solution which is fitted to many encoder/decoder neural models. The proposed STFNN is potential to achieve desirable performance with a compact model.

5. Conclusions

This paper presented a hybrid approach to matrix factorization and neural network which preserved the structure and factorization information of a spectro-temporal matrix in a layer-wise feedforward neural network through factorization over spectral and temporal domains. We implemented the deep neural factorization for speech recognition. A factorized error backpropagation algorithm was developed to realize the cross-entropy training for speech recognition. Backpropagation procedure was shown in a form of transpose factorization. An orthogonal constraint was imposed in finding the estimated factor matrices. Experimental results on noisy speech recognition demonstrated that the proposed deep models with combination of fully-connected layers or LSTM layer performed better than DNN, LSTM and CNN models with smaller parameter size. The estimated factor matrices did capture the property in frequency and time horizons. The orthogonal regularization performed better than ℓ_2 regularization. The proposed algorithm is generic and shall be extended to different classification problems in speech areas. The solution to regression problem will be discovered and examined. We are extending the study on multi-way factorized neural network in presence of multi-way observations by combing different modalities and channels.

6. References

[1] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[2] G. Saon and J.-T. Chien, "Large-vocabulary continuous speech recognition systems: A look at some recent advances," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 18–33, 2012.

[3] A. Cichocki, R. Zdunek, A. H. Phan, and S. Amari, *Nonnegative Matrix and Tensor Factorizations - Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation*. Wiley, 2009.

[4] C.-C. Hsu, T.-S. Chi, and J.-T. Chien, "Discriminative layered nonnegative matrix factorization for speech separation," in *Proc. of Annual Conference of International Speech Communication Association*, 2016, pp. 560–564.

[5] J.-T. Chien and P.-K. Yang, "Bayesian factorization and learning for monaural source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 1, pp. 185–195, 2016.

[6] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

[7] M. A. Ranzato, A. Krizhevsky, and G. E. Hinton, "Factored 3-way restricted Boltzmann machines for modeling natural images," in *Proc. of International Conference on Artificial Intelligence and Statistics*, 2010, pp. 621–628.

[8] D. Yu, L. Deng, and F. Seide, "The deep tensor neural network with applications to large vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 2, pp. 388–396, 2013.

[9] B. Hutchinson, L. Deng, and D. Yu, "Tensor deep stacking networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1944–1957, 2013.

[10] O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 22, no. 10, pp. 1533–1545, 2015.

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, pp. 2278–2324, 1998.

[12] Y.-T. Bao and J.-T. Chien, "Tensor classification network," in *Proc. of IEEE International Workshop on Machine Learning for Signal Processing*, 2015, pp. 1–6.

[13] J.-T. Chien and Y.-T. Bao, "Tensor-factorized neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2017.

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

[15] S. Zhang, H. Jiang, and L. Dai, "Hybrid orthogonal projection and estimation (HOPE): a new framework to learn neural networks," *Journal of Machine Learning Research*, vol. 17, no. 37, pp. 1–33, 2016.

[16] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding*, 2011.

[17] J.-T. Chien and T.-W. Lu, "Deep recurrent regularization neural network for speech recognition," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015, pp. 4560–4564.

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

[19] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. of International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[20] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. of Annual Conference of International Speech Communication Association*, 2014, pp. 338–342.

[21] D. Palaz, M. Magimai-Doss, and R. Collobert, "Analysis of cnn-based speech recognition system using raw speech as input," in *Proc. of Annual Conference of International Speech Communication Association*, 2015.

[22] J.-T. Chien and A. Misbullah, "Deep long short-term memory networks for speech recognition," in *Proc. of International Symposium on Chinese Spoken Language Processing*, 2016.