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

Stochastic Recurrent Neural Network for Speech Recognition

Jen-Tzung Chien, *Chen Shen*

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

Abstract

This paper presents a new stochastic learning approach to construct a latent variable model for recurrent neural network (RNN) based speech recognition. A hybrid generative and discriminative stochastic network is implemented to build a deep classification model. In the implementation, we conduct stochastic modeling for hidden states of recurrent neural network based on the variational auto-encoder. The randomness of hidden neurons is represented by the Gaussian distribution with mean and variance parameters driven by neural weights and learned from variational inference. Importantly, the class labels of input speech frames are incorporated to regularize this deep model to sample the informative and discriminative features for reconstruction of classification outputs. We accordingly propose the stochastic RNN (SRNN) to reflect the probabilistic property in RNN classification system. A *stochastic* error backpropagation algorithm is implemented. The experiments on speech recognition using TIMIT and Aurora4 show the merit of the proposed SRNN.

Index Terms: neural network, variational inference, stochastic error backpropagation, speech recognition

1. Introduction

Deep learning has been recognized as a new trend for modern information systems ranging from computer vision [1] to speech recognition [2, 3, 4], speaker recognition [5], source separation, and text categorization [6], to name a few. A deep model based on multi-layer perceptron with many layers is capable of extracting different levels of abstraction from input data and classifying the complex patterns in an efficient way. Such a deep neural network (DNN) is seen as a nonparametric model with a generic and deterministic structure. The uncertainty in hidden neurons and the variation in model structure are disregarded [7, 6]. This paper concerns the issue of stochastic modeling in neural network construction. The nature of stochastic behavior is reflected to improve the representation learning [8] and pave an avenue to improve the generalization for unknown test speech data.

A number of contributive works have been proposed to address this issue and provide insightful solutions to combine the paradigms of probabilistic model and neural network. In [9], a generalized expectation-maximization [10] training procedure was proposed to build a stochastic feedforward neural network containing both deterministic and stochastic hidden neurons. In [11], a deep generative stochastic network was developed to fulfill an unsupervised probabilistic model based on a Markov chain with trainable back-propagation algorithm. In [12, 13], a neural variational inference procedure was proposed to calculate the gradients of variational lower bound to train the feedforward neural network with latent variable. In [14], a deep latent Gaussian model introduced a deep generative architecture to build a deep model with stochastic backpropagation algorithm. In [15], the variational auto-encoder (VAE) was realized as a latent variable model where the variational lower bound was maximized to construct an unsupervised stochastic network. The recognition model and generative model were embedded in VAE to encode the input data into latent variables and then reconstruct the original data by a decoding distribution using the samples obtained by posterior distribution.

This study presents a new stochastic learning for recurrent neural network (RNN). The stochastic RNN (SRNN) is developed and implemented for speech recognition. This SRNN is originated from the perspectives of generative stochastic network and variational auto-encoder. The idea of SRNN is to infer the distribution of hidden state at each time step and use this distribution to predict classification output. However, we face the optimization problem with an intractable posterior and an intractable expectation. Variational Bayesian inference is accordingly performed [16]. An inference network is trained to infer the variational distribution from a set of feature inputs and class targets. At the same time, a discriminative network is trained to find the random sample of hidden variables via a sampling method for decoding or generation of acoustic labels. A new supervised stochastic modeling of RNN is proposed through an encoder for hidden variables and a decoder for classification outputs. The neural parameters under such a latent variable model are estimated by maximizing the variational lower bound of log marginal likelihood which is composed of two parts. One is the Kullback-Leibler divergence between posterior distribution and variational distribution of latent variables. The other one is the cross entropy error function for network outputs and class targets. Beyond deterministic RNN, SRNN provides a stochastic point of view which accommodates the uncertainty in hidden states and facilitates the reconstruction for analysis of RNN. Beyond traditional RNN, the proposed VRNN characterizes the dependencies between latent variables across subsequent time steps.

2. Related Works

This study presents a variational learning for recurrent neural network. VAE and RNN are two fundamentals.

2.1. Variational auto-encoder

VAE [15] was proposed to estimate the distribution of hidden variables $\mathbf{z}$ and use this information to reconstruct original signal $\mathbf{x}$. This distribution characterizes the randomness of hidden units which provides a vehicle to reconstruct different realizations of output signals rather than a point estimate of outputs in traditional auto-encoder. Accordingly, it makes possible to synthesize the generative samples and analyze the statistics of hidden information of neural network. Figure 1 shows how the output $\hat{\mathbf{x}}$ is reconstructed from original input $\mathbf{x}$. The graphical model of VAE is depicted by Figure 2(a) which consists of an encoder and a decoder. Encoder is seen as a recognition model which identifies the stochastic latent variables $\mathbf{z}$ us-

http://dx.doi.org/10.21437/Interspeech.2017-856

ing a variational posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ with parameters $\boldsymbol{\phi}$ shown by dashed line. Latent variables $\mathbf{z}$ are sampled by using variational posterior. These samples $\mathbf{z}$ are then used to generate or reconstruct original signal $\hat{\mathbf{x}}$ based on the decoder or generative model using likelihood function $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ with parameters $\boldsymbol{\theta}$ shown by solid line. The whole model is formulated by using the variational Bayesian expectation maximization algorithm. Variational parameters $\boldsymbol{\phi}$ and model parameters $\boldsymbol{\theta}$ are estimated by maximizing the *variational lower bound* of log likelihood $\log p(\mathbf{x}_{\leq T})$ from a collection of samples $\mathbf{x}_{\leq T} = \{\mathbf{x}_t\}_{t=1}^{T}$. *Stochastic* error backpropagation is implemented for variational learning. This VAE was extended to other unsupervised learning tasks [13] for finding the synthesized images.

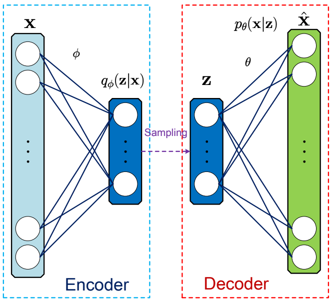

Figure 1: *Variational auto-encoder (VAE).*

2.2. Recurrent neural network

RNN is adopted as a nonlinear classification model to predict class posterior vector $\mathbf{y}_t$ from an input vector $\mathbf{x}_t$. One-hot encoding is used to express the target output vector $\mathbf{y}_t$. A basic RNN is composed of a chain of functional transformations in time horizon. The recurrent structure in RNN is crucial to learn temporal dependency from input time-series speech signals. The hidden unit $\mathbf{h}_t$ at time t is obtained from an D-dimensional input vector $\mathbf{x}_t$ at time t and a hidden vector $\mathbf{h}_{t-1}$ at time $t-1$ using a transformation $\mathcal{F}(\cdot)$ via $\mathbf{h}_t = \mathcal{F}(\mathbf{x}_t, \mathbf{h}_{t-1})$ with weight parameters $\mathbf{w}$. The output $\mathbf{y}_t$ is obtained from hidden unit $\mathbf{h}_t$ at the same time t through a transformation $\hat{\mathbf{y}}_t = \mathcal{F}(\mathbf{h}_t)$. The transformation is composed of an affine function and an activation function. The weight parameters $\mathbf{w}$ for connections from inputs $\{\mathbf{x}_t, \mathbf{h}_{t-1}\}$ to hidden units $\mathbf{h}_t$ and from hidden units $\mathbf{h}_t$ to output $\mathbf{y}_t$ are estimated by minimizing the cross-entropy error function $E_{\mathbf{w}} = -\frac{1}{2}\sum_{t=1}^{T} \mathbf{y}_t \log \hat{\mathbf{y}}_t$ from a collection of T input-output data pairs $\mathcal{D} = \{\mathbf{x}_t, \mathbf{y}_t\}$. Parameters $\mathbf{w}$ are trained by using mini-batches of $\mathcal{D}$ according to the stochastic gradient descent (SGD) algorithm where the gradient of objective function using a mini-batch is calculated for parameter updating. During SGD training, hidden units $\mathbf{h}_t$ are assumed to be deterministic. No random reconstruction is embedded in baseline RNN. This study develops a stochastic model of RNN to tackle classification problem for speech recognition.

3. Stochastic Learning for RNN

This paper presents a stochastic learning algorithm for RNN which incorporates VAE into RNN construction.

3.1. Supervised variational auto-encoder

Originally, VAE was developed for unsupervised learning of a reconstructed signal from an original signal which is not fitted to deal with classification problem. Here, a supervised VAE is proposed by introducing the label information $\mathbf{y}$ for an input $\mathbf{x}$ as illustrated in Figure 2(b). Different from VAE, this supervised VAE encodes the latent variable $\mathbf{z}$ by using a recognition model $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{y})$ which is conditioned on speech sample $\mathbf{x}$ as well as target label $\mathbf{y}$. Latent variables $\mathbf{z}$ are then sampled to reconstruct target label $\mathbf{y}$ using a *conditional likelihood* $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}, \mathbf{z})$. This is different from unsupervised VAE which reconstructs the input $\mathbf{x}$ based on standard likelihood $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$.

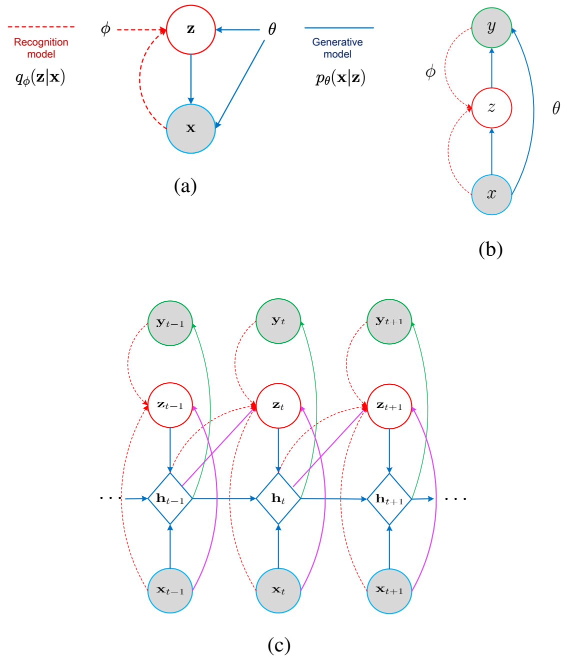

Figure 2: *Graphical representation for (a) unsupervised VAE, (b) supervised VAE and (c) stochastic RNN.*

3.2. Stochastic recurrent neural network

Although the supervised VAE is feasible to classification problem, the temporal information from a set of T speech frames $\{\mathbf{x}_{\leq T}, \mathbf{y}_{\leq T}\}$ is missing since time-dependent latent information is disregarded. Accordingly, we integrate the supervised VAE and RNN and carry out the stochastic RNN (SRNN) for sequence classification in speech recognition. To do so, we estimate a set of T time-dependent hidden units $\mathbf{h}_{\leq T}$ corresponding to the observed speech signals $\mathbf{x}_{\leq T}$ which are used to produce the RNN outputs $\mathbf{y}_{\leq T}$ as the class posteriors for classification task. The hidden units $\mathbf{h}_{\leq T}$ are characterized and generated by hidden variables $\mathbf{z}_{\leq T}$. Similar to VAEs in Figures 2(a) and 2(b), the proposed SRNN is equipped with an encoder and a decoder. This SRNN learns the temporal and stochastic information from time-series observations and hidden states. As demonstrated in Figure 2(c), the encoder in SRNN is designed to encode or identify the distribution $q_{\boldsymbol{\phi}}(\mathbf{z}_t|\mathbf{x}_t, \mathbf{y}_t, \mathbf{h}_{t-1})$ of latent variable $\mathbf{z}_t$ from input-output pair $\{\mathbf{x}_t, \mathbf{y}_t\}$ at each time t and hidden feature $\mathbf{h}_{t-1}$ at previous time $t-1$ as shown by dashed lines. Given the random samples $\mathbf{z}_t$ from variational distribution $q_{\boldsymbol{\phi}}(\cdot)$, the decoder in SRNN is introduced to realize the hidden units $\mathbf{h}_t = \mathcal{F}(\mathbf{x}_t, \mathbf{z}_t, \mathbf{h}_{t-1})$ at current time t as shown in solid lines. Hidden unit $\mathbf{h}_t$ acts as a realization or surrogate of hidden variable $\mathbf{z}_t$. The generative conditional likeli-

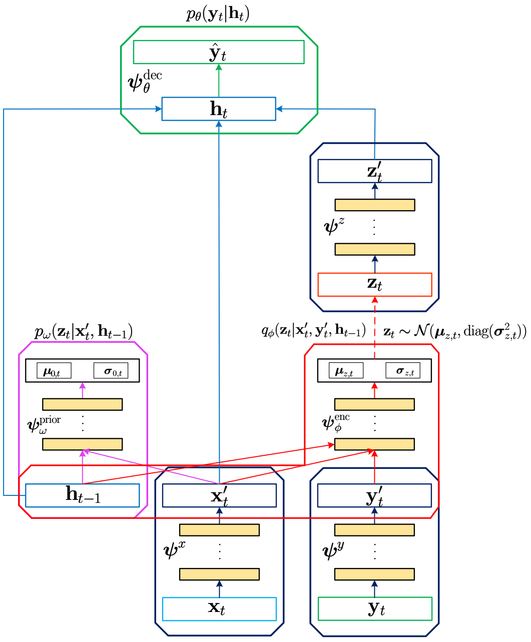
$p_\theta(\mathbf{y}_t|\mathbf{h}_t)$
$\hat{\mathbf{y}}_t$
$\boldsymbol{\psi}_\theta^{\text{dec}}$
$\mathbf{h}_t$
$\mathbf{z}'_t$
$\boldsymbol{\psi}^z$
$\mathbf{z}_t$
$p_\omega(\mathbf{z}_t|\mathbf{x}'_t, \mathbf{h}_{t-1})$
$q_\phi(\mathbf{z}_t|\mathbf{x}'_t, \mathbf{y}'_t, \mathbf{h}_{t-1})$
$\mathbf{z}_t \sim \mathcal{N}(\boldsymbol{\mu}_{z,t}, \text{diag}(\boldsymbol{\sigma}^2_{z,t}))$
$\boldsymbol{\mu}_{0,t}$
$\boldsymbol{\sigma}_{0,t}$
$\boldsymbol{\psi}_\omega^{\text{prior}}$
$\boldsymbol{\mu}_{z,t}$
$\boldsymbol{\sigma}_{z,t}$
$\boldsymbol{\psi}_\phi^{\text{enc}}$
$\mathbf{h}_{t-1}$
$\mathbf{x}'_t$
$\mathbf{y}'_t$
$\boldsymbol{\psi}^x$
$\boldsymbol{\psi}^y$
$\mathbf{x}_t$
$\mathbf{y}_t$

Figure 3: *System topology for stochastic RNN.*

hood $p_{\boldsymbol{\theta}}(\mathbf{y}_t|\mathbf{h}_t = \mathcal{F}(\mathbf{x}_t, \mathbf{z}_t, \mathbf{h}_{t-1}))$ is estimated to obtain the random output $\mathbf{y}_t$. Comparable with standard RNN, the hidden units $\mathbf{h}_t$ in SRNN are used to generate the outputs $\hat{\mathbf{y}}_t$ by $\hat{\mathbf{y}}_t \sim p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{h}_t)$. SRNN pursues the *stochastic* generation of classification outputs guided by the variational learning of hidden features. A stochastic learning of RNN is fulfilled by the following optimization procedure.

3.3. Optimization procedure

Figure 3 depicts the topology for implementation of SRNN. In model inference, we maximize the variational lower bound $\mathcal{L}$ of logarithm of conditional likelihood $p(\mathbf{y}_{\leq T}|\mathbf{x}_{\leq T}) = \prod_{t=1}^T \sum_{\mathbf{z}_t} p_{\boldsymbol{\theta}}(\mathbf{y}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{\leq t}) p_{\boldsymbol{\omega}}(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})$ which is decomposed into two terms containing parameters $\boldsymbol{\theta}$ and $\boldsymbol{\omega}$

$$\mathcal{L} \triangleq \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_{\leq T}|\mathbf{x}_{\leq T}, \mathbf{y}_{\leq T})}\left[\sum_{t=1}^T \left(\log p_{\boldsymbol{\theta}}(\mathbf{y}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{\leq t}) - \mathcal{D}_{\text{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{y}_{\leq t}, \mathbf{z}_{<t}) \| p_{\boldsymbol{\omega}}(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t}))\right)\right]. \tag{1}$$

The first term $\log p_{\boldsymbol{\theta}}(\mathbf{y}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{\leq t})$ is seen as a negative cross entropy error function which is obtained by

$$\sum_{c=1}^C \sum_{t=1}^T y_{tc} \log \left[s(\mathbf{W}^{\text{dec}} \mathbf{h}_t(\mathbf{x}_t, \mathbf{z}_t, \mathbf{h}_{t-1}) + \mathbf{b}^{\text{dec}}) \right]_c \tag{2}$$

where $s(\cdot)$ is a softmax operator and $\boldsymbol{\theta}^{\text{dec}} = \{\mathbf{W}^{\text{dec}}, \mathbf{b}^{\text{dec}}\}$ denotes the weight parameters of decoding or *discriminative network* $\boldsymbol{\psi}_{\boldsymbol{\theta}}^{\text{dec}}(\mathbf{h}_t)$ for C class posteriors $\hat{\mathbf{y}}_t = \{\hat{y}_{tc}\}$. The second term is an expected Kullback-Leibler (KL) divergence between distributions $q_{\boldsymbol{\phi}}(\cdot)$ and $p_{\boldsymbol{\omega}}(\cdot)$. In maximization of Eq. (1), we first calculate the prior distribution of latent variable $\mathbf{z}_t$ which is Gaussian distributed by $p_{\boldsymbol{\omega}}(\mathbf{z}_t|\mathbf{x}'_t, \mathbf{h}_{t-1}) = \mathcal{N}(\boldsymbol{\mu}_{0,t}, \text{diag}(\boldsymbol{\sigma}^2_{0,t}))$ with the mean and variance calculated by a *prior network* $[\boldsymbol{\mu}_{0,t}, \boldsymbol{\sigma}^2_{0,t}] = \boldsymbol{\psi}_{\boldsymbol{\omega}}^{\text{prior}}(\mathbf{x}'_t, \mathbf{h}_{t-1})$ using the encoding weights $\boldsymbol{\omega}$. Then, the variational distribution is calculated at each speech frame by using an Gaussian $q_{\boldsymbol{\phi}}(\mathbf{z}_t|\mathbf{x}'_t, \mathbf{y}'_t, \mathbf{h}_{t-1}) = \mathcal{N}(\boldsymbol{\mu}_{z,t}, \text{diag}(\boldsymbol{\sigma}^2_{z,t}))$ with mean and variance calculated by an *inference network* $[\boldsymbol{\mu}_{z,t}, \boldsymbol{\sigma}^2_{z,t}] = \boldsymbol{\psi}_{\boldsymbol{\phi}}^{\text{enc}}(\mathbf{x}'_t, \mathbf{y}'_t, \mathbf{h}_{t-1})$ using the encoding weights $\boldsymbol{\phi}^{\text{enc}}$. Here, $\mathbf{x}'_t$ and $\mathbf{y}'_t$ denote the encoded features of $\mathbf{x}_t$ and $\mathbf{y}_t$ with reduced dimensions by using feature extractors $\boldsymbol{\psi}^x(\mathbf{x}_t)$ and $\boldsymbol{\psi}^y(\mathbf{y}_t)$, which can be also expressed by neural network weights $\boldsymbol{\phi}^x$ and $\boldsymbol{\phi}^y$, respectively. The recognition or encoding phase consists of four sets of encoding weights $\{\boldsymbol{\phi}^x, \boldsymbol{\phi}^y, \boldsymbol{\phi}^{\text{enc}}, \boldsymbol{\omega}\}$.

In the generation or decoding phase, we first apply the feature extractor $\boldsymbol{\psi}^z(\mathbf{z}_t)$ with parameter $\boldsymbol{\theta}^z$ to estimate the feature $\mathbf{z}'_t$ corresponding to latent variable $\mathbf{z}_t$. The variable $\mathbf{z}_t$ is sampled from the Gaussian distribution $q_{\boldsymbol{\phi}}(\mathbf{z}_t|\mathbf{x}'_t, \mathbf{y}'_t, \mathbf{h}_{t-1})$ which was obtained in encoding phase. Then, we calculate the conditional likelihood $p_{\boldsymbol{\theta}}(\mathbf{y}_t|\mathbf{h}_t)$ at each time t. This likelihood is estimated from the outputs of discriminative network $\boldsymbol{\psi}_{\boldsymbol{\theta}}^{\text{dec}}(\mathbf{h}_t)$ with parameters $\boldsymbol{\theta}^{\text{dec}}$. This likelihood is used to calculate the class posteriors $\hat{\mathbf{y}}_t$ by using the inputs from $\mathbf{h}_t = \mathcal{F}(\mathbf{x}'_t, \mathbf{z}'_t, \mathbf{h}_{t-1})$ which is a function of $\mathbf{x}'_t$, $\mathbf{z}'_t$ and $\mathbf{h}_{t-1}$ with parameters $\boldsymbol{\theta}^h$. There are three sets of parameters $\{\boldsymbol{\theta}^z, \boldsymbol{\theta}^{\text{dec}}, \boldsymbol{\theta}^h\}$ in SRNN decoder. Importantly, the expectation in variational lower bound Eq. (1) is calculated by using L samples $\{\mathbf{z}_t^{(l)}\}$ obtained via variational distribution $q_{\boldsymbol{\phi}}(\mathbf{z}_t|\mathbf{x}'_t, \mathbf{y}'_t, \mathbf{h}_{t-1})$

$$\tilde{\mathcal{L}} = \sum_{t=1}^T \left(\frac{1}{L} \sum_{l=1}^L \log p_{\boldsymbol{\theta}}(\mathbf{y}_t|\mathbf{x}_t, \mathbf{z}_t^{(l)}, \mathbf{h}_{t-1}) - \mathcal{D}_{\text{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}_t^{(l)}|\mathbf{x}_t, \mathbf{y}_t, \mathbf{h}_{t-1}) \| p_{\boldsymbol{\omega}}(\mathbf{z}_t^{(l)}|\mathbf{x}_t, \mathbf{h}_{t-1})) \right) \tag{3}$$

where KL divergence between two Gaussians is given by

$$\frac{1}{2}\left[\log\|\text{diag}(\boldsymbol{\sigma}^2_{0,t})\| - \log\|\text{diag}(\boldsymbol{\sigma}^2_{z,t})\| - K + \text{Tr}\left((\text{diag}(\boldsymbol{\sigma}^2_{0,t}))^{-1} \text{diag}(\boldsymbol{\sigma}^2_{z,t}) \right) + (\boldsymbol{\mu}_{0,t} - \boldsymbol{\mu}_{z,t})^\top (\text{diag}(\boldsymbol{\sigma}^2_{0,t}))^{-1} (\boldsymbol{\mu}_{0,t} - \boldsymbol{\mu}_{z,t}) \right]. \tag{4}$$

However, directly sampling $\mathbf{z}_t$ using the Gaussian distribution with mean $\boldsymbol{\mu}_{z,t}$ and variance $\boldsymbol{\sigma}^2_{z,t}$ obtained by the encoding network $\boldsymbol{\psi}_{\boldsymbol{\phi}}^{\text{enc}}(\mathbf{x}'_t, \mathbf{y}'_t, \mathbf{h}_{t-1})$ is unstable with high variance. We apply the re-parameterization trick [14] to resolve this problem. Namely, we sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and use this sample to determine the sample of latent variable by $\mathbf{z}_t^{(l)} = \boldsymbol{\mu}_{z,t} + \boldsymbol{\sigma}_{z,t} \odot \boldsymbol{\epsilon}^{(l)}$ where $\odot$ denotes the element-wise multiplication. The stochastic training procedure for encoding weights $\{\boldsymbol{\phi}^x, \boldsymbol{\phi}^y, \boldsymbol{\phi}^{\text{enc}}, \boldsymbol{\omega}\}$ and decoding weights $\{\boldsymbol{\theta}^z, \boldsymbol{\theta}^{\text{dec}}, \boldsymbol{\theta}^h\}$ is realized by a *stochastic gradient variational Bayes estimator* where the gradients of $\tilde{\mathcal{L}}$ with respect to these weights are calculated for estimating the individual weights.

4. Experiments

In the experiments, we carry out the proposed SRNN for speech recognition using TIMIT and Aurora4 tasks.

4.1. Experimental setup

TIMIT is an acoustic-phonetic continuous speech corpus for evaluation of phoneme recognition. This corpus contained 3696 sentences from 462 speakers for training and 192 utterances from 24 speakers for testing. A separate development set of

50 speakers was used to select the best result. All models are trained to recognize 49 phonemes. We kept 10% of training data as validation data for hyperparameter tuning and topology configuration. Phone error rate (PER) (%) was reported for development set and test set. Aurora4 is a noisy English broadcast news speech database [17]. Training set contained 7,137 utterances from 83 speakers with 14 hours. Test set consisted of 4,620 utterances from 8 speakers. Vocabulary size was 5K. Evaluation sets were grouped into subset A (clean data), B (noisy data), C (clean data with channel distortion) and D (noisy data with channel distortion). Word error rate (WER) (%) was averaged over different subsets. A small set of training data was held out for validation. Kaldi toolkit [18] was used in our implementation. The baseline GMM-HMM triphone system was trained by using the Kaldi recipe for TIMIT and Aurora4. The GMM-HMM and RNN hybrids used the phoneme bigrams in TIMIT and the word trigrams in Aurora4 as the language model during decoding. GMM-HMM was used to find the state alignments of training utterances. The networks were trained by using 40 dimensional fMLLR features in TIMIT and 40 log Mel frequency bank features in Aurora4. The input at each frame was characterized by 440 features where the contextual 5 frames (40×11) were considered. Output units corresponded to 147 (49 phonemes × 3 states) context independent states in TIMIT and 2,035 tied context dependent states (via a decision tree) in Aurora4 [19]. The mini-batch size of 256 frames was adopted in SGD training where Adam algorithm [20] was performed. ReLU activation function was used.

In the experiments, we compare the recognition performance of using GMM-HMM [21], DNN, RNN, deep RNN (DRNN) [22] and the proposed SRNN. Pre-training was applied. Tikhonov and ℓ_2 regularization [19] was used in optimization with backpropagation through time. DNN and DRNN were implemented by referring to [22]. RNN was deterministic realization of SRNN with consistent topology. In implementation of SRNN, the input of 440 dimensional speech feature vector was fed to find C=147 or 2,035 outputs for different phonetic states. The feature extractors calculated $\mathbf{x}'_t$ and $\mathbf{y}'_t$ with the same dimension 250. Single layer with ReLU activation was specified. Using $\mathbf{x}'_t$ and $\mathbf{y}'_t$ at time t and hidden state $\mathbf{h}_{t-1}$ at time $t-1$, the prior network and the inference network were constructed using ReLU activation with 150 dimensional hidden units. The outputs of two networks were calculated by linear activation and viewed as 100 dimensional mean and variance for Gaussians in prior distribution $p_{\omega}(\mathbf{z}_t|\mathbf{x}'_t)$ and variational distribution $q_{\phi}(\mathbf{z}_t|\mathbf{x}'_t, \mathbf{y}'_t, \mathbf{h}_{t-1})$. The latent variable $\mathbf{z}_t$ was sampled and transformed to 150 dimensional $\mathbf{z}'_t$. The 150 dimensional hidden state $\mathbf{h}_t$ was obtained by RNN with linear activation. This hidden state was then forwarded through a hidden layer with 450 units for calculating 513 dimensional activation vector $\mathbf{a}_t$. ReLU activation was applied for finding C dimensional class posteriors $\hat{\mathbf{y}}_t$. In the implementation, we first trained the inference network and discriminative network by maximizing the first term in Eq. (3). After convergence, we used the trained parameters as the initialization to optimize the first and second terms to find prior network and fine-tine the inference and discriminative networks.

4.2. Experimental result

Table 1 reports the speech recognition performance by using different methods on TIMIT task where PERs of development set and test set are included. PERs of test set are higher than those of development set. PERs are reduced by using neural network models. Recurrent neural networks perform better than feedforward neural network. From this set of experiments, we find that RNN is improved by increasing the depth of hidden layers (DRNN) as well as expressing the stochastic behavior of hidden state (SRNN). The lowest PERs are achieved by using SRNN. Table 2 lists the WERs of using various models on Aurora4 task where four environmental conditions are investigated. Averaged WER over 14 subsets from four conditions is also shown. The results show that SRNN can improve the noise robustness in speech recognition especially for the conditions contaminated with noise (conditions B and D). The improvement of SRNN is larger than that of other methods. Averaged WER is reduced from 14.38% of baseline RNN to 11.90% of SRNN.

Table 1: *Comparison of PERs (%) on TIMIT task.*

Model	Dev	Test
GMM-HMM	23.2	24.3
DNN	21.1	22.3
RNN	20.8	21.7
DRNN	20.0	20.9
SRNN	**18.3**	**19.8**

Table 2: *Comparison of WERs (%) on Aurora4 task.*

Model	Conditions				Avg.
	A	B	C	D	
GMM-HMM	7.29	12.97	12.61	27.66	18.83
DNN	4.33	9.13	11.83	24.69	15.65
RNN	4.10	8.31	10.02	22.89	14.38
DRNN	3.29	7.09	8.71	20.83	12.82
SRNN	3.33	6.25	8.35	19.56	**11.90**

5. Conclusions

We have presented a new stochastic learning of recurrent neural network to tackle the deterministic assumption in conventional recurrent neural network. This deep model was constructed by merging the supervised variational auto-encoder into recurrent neural network so that we obtained the random samples of hidden variables to reconstruct classification outputs for speech recognition. The encoder for recognition of hidden variables and the decoder for generation of classification outputs were driven by a variational learning algorithm which maximized the variational lower bound of logarithm of conditional likelihood. A supervised learning was carried out for phoneme recognition and noisy speech recognition. An error backpropagation algorithm was developed to estimate the model parameters for feature extraction, prior network, inference network and discriminative network. The gradients of variational lower bound with respect to weight parameters in different processing units were calculated. The advantage of the proposed model was illustrated through the experiments on speech recognition using TIMIT and Aurora4. It was shown that lower phone error rates and word error rates were obtained by using stochastic recurrent neural network when compared with deep neural network and other types of recurrent neural networks. Future study will extend the stochastic recurrent neural network to gated recurrent unit or memory augmented neural network. The extension to language model will be examined for speech recognition.

6. References

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

[2] 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.

[3] 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.

[4] J.-T. Chien and Y.-C. Ku, "Bayesian recurrent neural network for language modeling," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 2, pp. 361–374, 2016.

[5] N. Li, M.-W. Mak, and J.-T. Chien, "DNN-driven mixture of PLDA for robust speaker verification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1371–1383, 2017.

[6] J.-T. Chien and C.-H. Lee, "Deep unfolding for topic models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[7] C.-H. Lee and J.-T. Chien, "Deep unfolding inference for supervised topic model," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016, pp. 2279–2283.

[8] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[9] Y. Tang and R. R. Salakhutdinov, "Learning stochastic feedforward neural networks," in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds., pp. 530–538. 2013.

[10] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of Royal Statistical Society B*, vol. 39, no. 1, pp. 1–38, 1977.

[11] Y. Bengio, E. Thibodeau-Laufer, G. Alain, and J. Yosinski, "Deep generative stochastic networks trainable by backprop," in *Proc. of International Conference on Machine Learning*, 2014, pp. 226–234.

[12] A. Mnih and K. Gregor, "Neural variational inference and learning in belief networks," in *Proc. of International Conference on Machine Learning*, 2014, pp. 1791–1799.

[13] Y. Miao, C. Ox, L. Yu, and P. Blunsom, "Neural variational inference for text processing," in *Proc. of International Conference on Machine Learning*, 2016, pp. 1727–1736.

[14] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proc. of International Conference on Machine Learning*, 2014, pp. 1278–1286.

[15] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. of International Conference on Learning Representation*, 2014.

[16] S. Watanabe and J.-T. Chien, *Bayesian Speech and Language Processing*, Cambridge University Press, 2015.

[17] D. Pearce and J. Picone, "Aurora working group: DSR front end LVCSR evaluation AU/384/02," *Inst. for Signal & Inform. Process., Mississippi State Univ., Tech. Rep*, 2002.

[18] 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.

[19] J.-T. Chien and T.-W. Lu, "Tikhonov regularization for deep neural network acoustic modeling," in *Proc. of IEEE Spoken Language Technology Workshop*, 2014, pp. 147–152.

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

[21] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, "End-to-end continuous speech recognition using attention-based recurrent NN: First results," *arXiv preprint arXiv:1412.1602*, 2014.

[22] 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.