INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Stochastic Convolutional Recurrent Networks for Language Modeling

Jen-Tzung Chien, Yu-Min Huang

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

Abstract

Sequential learning using recurrent neural network (RNN) has been popularly developed for language modeling. An alternative sequential learning was implemented by the temporal convolutional network (TCN) which is seen as a variant of one-dimensional convolutional neural network (CNN). In general, RNN and TCN are fitted to capture the long-term and the short-term features over natural sentences, respectively. This paper is motivated to fulfill TCN as the encoder to extract short-term dependencies and then use RNN as the decoder for language modeling where the dependencies are integrated in a long-term semantic fashion for word prediction. A new sequential learning based on the convolutional recurrent network (CRN) is developed to characterize the *local dependencies* as well as the *global semantics* in word sequences. Importantly, the stochastic modeling for CRN is proposed to facilitate model capacity in neural language model where the uncertainties in training sentences are represented for variational inference. The complementary benefits of CNN and RNN are merged in sequential learning where the latent variable space is constructed as a generative model for sequential prediction. Experiments on language modeling demonstrate the effectiveness of stochastic convolutional recurrent network relative to the other sequential machines in terms of perplexity and word error rate.

Index Terms: Latent variable model, language model, convolutional neural network, recurrent neural network

1. Introduction

Deep neural networks have been prosperously developed for a wide range of applications in speech processing [1], natural language processing [2] and computer vision [3]. Basically, deep models can handle high-dimensional data with the complicated mapping between input signals and output targets, and perform well for different classification and regression tasks. Nevertheless, it is still challenging to carry out a desirable generation task in presence of high-dimensional data. Meanwhile, sequence data in temporal and spatial domains are everywhere in real world and is ranged from speech signals to music signals, natural sentences and video streams, to name a few. When we deal with sequential learning and generation, it is important to predict or generate future targets based on all previous samples due to the causal property in signals. Such a prediction is called the autoregressive generation where the prediction at each time step is conditioned on all previous observations. Autoregressive generation is seen as a building block in many systems with temporal and spatial signals. This paper presents new stochastic modeling and sequential learning [4] for autoregressive generation where an inference and generative procedure based on a hybrid architecture of convolutional neural network (CNN) [5] and recurrent neural network (RNN) is proposed. In general, RNN [6, 7] is designed as a recurrent machine which identifies the temporal or spatial features from sequential patterns. The dynamic state or internal memory is evolved through time. RNN has been recognized as a popular solution to autoregressive model in different practical systems. Recently, the temporal convolutional network (TCN) [8, 9] was proposed for sequential learning using temporal data in spite of many successful spatial models for image data using CNN in computer vision. Typically, TCN is beneficial for parallel computation which provides rapid prediction. Multilayer TCN can capture the temporal hierarchy where different layers represent various sizes of receptive field. RNN and TCN are both feasible to sequential modeling. This study aims to combine TCN and RNN in construction of the so-called convolutional recurrent network (CRN) for sequential learning.

Generally, TCN is professional to capture temporal information with short-term features in local fields. RNN is specialized to identify long-term semantics in global contexts [10]. The proposed CRN would like to infer or encode local information via convolutional layers and then generate or decode each individual time sample via recurrent layers. CRN corresponds to implementing TCN as encoder and RNN as decoder. A hybrid model of TCN and RNN is established. The complementary local and global features are characterized. Importantly, the recurrent layers in CRN are used to relax the limitation of TCN where the size of receptive field is constrained by the number of layers. CRN allocates the recurrent layers on top of convolutional layers so that the insufficiency of long-term temporal characteristics in TCN can be compensated. Furthermore, the stochastic variant of CRN (SCRN) is proposed to improve the robustness of CRN for sequential prediction. The randomness of sequential latent variables is reflected in optimization procedure via variational inference [11, 12]. The variational lower bound of log likelihood, marginalized over latent variables, is maximized. SCRN is proposed as a new latent variable model. The experiments on language modeling are conducted to investigate the performance of different sequential machines. We demonstrate the performance of the proposed language model by comparing with RNN and TCN language models by using different datasets in terms of different metrics.

2. Sequential Learning

This paper constructs a new neural network structure which integrates convolutional neural network and recurrent neural network for sequential learning. In the literature, it has been common to integrate CNN and RNN for speech, image and video processing in different tasks and applications. For example, two-dimensional (2-D) CNN was used for representation of images while RNN was applied to capture temporal relations in video data [13]. In [14–16], CNN was concatenated with RNN for object recognition, sign language recognition and person identification. The hybrid CNN and RNN was also exploited for natural language processing in [17, 18]. In previous hybrid CNN and RNN models, the spatial model using 2-D CNN was employed. These models are different from 1-D CNN or TCN where the causality and dilation [19] are considered. To facili-

http://dx.doi.org/10.21437/Interspeech.2020-1493

tate sequential learning, TCN can act as a meaningful approach to capture temporal hierarchy. The upper layers are feasible to span larger receptive field with increasing window size. This paper builds the temporal hierarchy by using TCN where RNN is further allocated on top of TCN. The receptive field in the proposed CRN is represented with unbounded window in temporal domain. In addition, we develop a latent variable model for stochastic variant of CRN by using variational inference. Previously, stochastic modeling has improved the generalization in different learning tasks. In [20], a stochastic variant of RNN was proposed to build a recurrent latent variable model. Recently, the stochastic property was merged in TCN [21, 22] where RNN was excluded in temporal modeling. In [20–22], an additional latent variable was added to fulfill stochastic modeling in sequential learning based on variational inference. A simple linear transformation was used in either encoder or decoder. This study presents how a variational or stochastic CRN is formulated to improve sequential learning where TCN and RNN separately act as the encoder for local features and the decoder for global views in inference and generation stages, respectively. A meaningful two-stage inference is developed in a way similar to variational autoencoder (VAE) [11]. The sequential learning based on convolutional recurrent network is implemented as addressed in what follows.

3. Hybrid Network Architecture

Typically, temporal convolutional network (TCN) [23] performs the casual convolution and expands the receptive field by stacking a number of dilated convolutions. The size of receptive fields is determined by the number of layers. In general, TCN introduces a temporal hierarchy where the upper layers can access longer sub-sequences of input signals so as to learn representations at a larger time scale. TCN has an attractive architecture to capture temporal dependency at various time scales. On the other hand, RNN with gating mechanism [7, 24, 25] is feasible to capture temporal dependency with unbounded length. It is accordingly reasonable to incorporate RNN as the most upper layer so as to cope with the constraint in TCN. The deterministic variant of convolutional recurrent network (CRN) is developed.

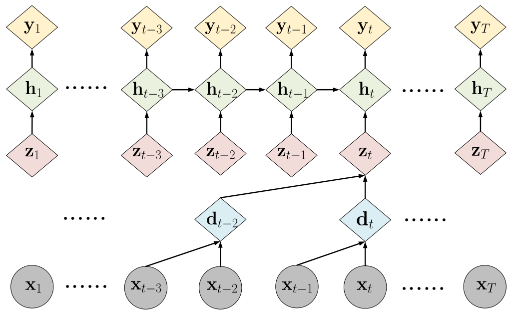

Figure 1: *Illustration for convolutional recurrent network.*

Figure 1 depicts the architecture of CRN consisting of input layer $\{\mathbf{x}_t\}$ and output layer $\{\mathbf{y}_t\}$ where RNN is built on top of TCN. TCN is seen as 1-D CNN with kernel size 2 and two hidden layers. Dilation is 1 and 2 in first layer $\{\mathbf{d}_t\}$ and second hidden layer $\{\mathbf{z}_t\}$, respectively. Information propagation is run in a bottom-up manner. More hidden layers can be applied. TCN acts as an encoder to extract local temporal features $\{\mathbf{z}_t\}$ with a receptive field containing four time steps while RNN serves as a decoder to capture global view via long-term recurrent codes $\{\mathbf{h}_t\}$. The size of receptive field is unbounded in RNN. Basically, convolution operation can extract local features efficiently as shown in many computer vision tasks. Nevertheless, in sequential learning, TCN still suffers from the limited window of receptive field caused by the specifications of dilation and kernel size. To relax this limitation, RNN is combined as a decoder to continuously capture long-term temporal semantics from the beginning $\mathbf{x}_1$. TCN and RNN are complementary with different functions which are mutually leveraged to build CRN. Take text modeling as an example, TCN encoder can be regarded as a new type of *word embedding*. Instead of embedding every words independently, TCN encodes a word by relating to its neighboring words. This is reasonable because there might be different meanings for a single word. Contextual information provides an efficient and meaningful way to calculate the *causal* and *dilated convolutional embedding* $\mathbf{z}_t$ for each word $\mathbf{x}_t$. Taking a look at previous words $\{\mathbf{x}_i\}_{i=t-3}^t$ is helpful to find precise meaning or embedding of current word $\mathbf{z}_t$. RNN decoder can accordingly identify long-range information from full text paragraph with limited computation overhead. This RNN extends the temporal hierarchy since the size of receptive field is continuously increasing to final time step T . Such a deterministic variant of CRN is implemented without probabilistic interpretation where the robustness in sequential learning is disregarded.

4. Latent Variable Representation

Although the combination of TCN and RNN can improve the model capability, the enhanced model in the resulting CRN usually induces the issue of robustness. Pursuing the stochastic property for an existing deterministic model is beneficial to handle this issue. Here, variational inference provides a theoretical bound for generation. Moreover, stochastic variants of autoregressive models in previous works [20–22] have shown significant improvement in characterizing complex and structural relation between $\{\mathbf{x}_t\}$ and $\{\mathbf{y}_t\}$. This study is accordingly motivated by implementing the stochastic CRN (SCRN) where the word embeddings from TCN encoder $\{\mathbf{z}_t\}$ are *random*. SCRN differs from CRN where *deterministic* variables $\{\mathbf{z}_t\}$ are assumed as shown by diamonds in Figure 1. SCRN is constructed with an inference model as encoder and a generative model as decoder. TCN is applied to infer or encode a Gaussian variable $\mathbf{z}_t$ from input signal $\mathbf{x}_t$ with mean $\boldsymbol{\mu}_t$ and standard deviation $\boldsymbol{\sigma}_t$ while RNN is adopted to generate output signal $\mathbf{y}_t$ from a set of Gaussian samples $\mathbf{z}_t$. The whole procedure of inference and generation is governed by variational inference as detailed as follows. Stochastic variant of CRN is established.

The proposed SCRN consists of prior network, inference network and generation network. First, the deterministic hidden state $\mathbf{d}_t^{(l)}$ at layer l and time t is calculated by 1-D convolution along a L-layer network in a bottom-up fashion based on

$$\mathbf{d}_t^{(l)} = \mathrm{Conv}\left(\mathbf{d}_t^{(l-1)}, \mathbf{d}_{t-j}^{(l-1)}\right), \quad 1 \le l \le L \qquad (1)$$

where $\mathbf{d}_t^{(0)} \triangleq \mathbf{x}_t$ and $j = 2^{l-1}$ means the dilation or the expansion window with skip. This hidden state $\mathbf{d}_t^{(l)}$ summarizes previous input signals $\mathbf{x}_{\le t} = \{\mathbf{x}_i\}_{i=1}^t$ within a receptive field ended at time t. The latent codes $\{\mathbf{d}_t^{(l)}\}$ are then transformed by a linear matrix to calculate Gaussian mean $\boldsymbol{\mu}_t$ and standard deviation $\boldsymbol{\sigma}_t$ for sampling of the latent variables $\{\mathbf{z}_t^{(l)}\}$ in a latent variable model. In particular, each sample $\mathbf{z}_t^{(l)}$ at layer l and time t is conditioned on the latent code $\mathbf{z}_t^{(l+1)}$ at a higher layer $l+1$ at time t and the deterministic state $\mathbf{d}_{t-1}^{(l)}$ at layer l at previous time $t-1$ [26]. Using SCRN, the inference model is built according to a *prior network* of $\mathbf{z}_t$ given by the historic samples $\mathbf{x}_{<t}=\{\mathbf{x}_i\}_{i=1}^{t-1}$

$$p_\omega(\mathbf{z}_t|\mathbf{x}_{<t})=p_\omega(\mathbf{z}_t^{(L)}|\mathbf{d}_{t-1}^{(L)})\prod_{l=1}^{L-1}p_\omega(\mathbf{z}_t^{(l)}|\mathbf{z}_t^{(l+1)},\mathbf{d}_{t-1}^{(l)}) \quad (2)$$

which is expressed by using the posterior of $\mathbf{z}_t^{(l)}$ in different layer l given by the dilation embeddings $\mathbf{d}_{t-1}^{(l)}$ at previous time $t-1$

$$p_\omega(\mathbf{z}_t^{(l)}|\mathbf{z}_t^{(l+1)},\mathbf{d}_{t-1}^{(l)})=\mathcal{N}(\boldsymbol{\mu}_{p,t}^{(l)},\boldsymbol{\sigma}_{p,t}^{(l)}). \quad (3)$$

The Gaussian parameters of this posterior

$$[\boldsymbol{\mu}_{p,t}^{(l)},\boldsymbol{\sigma}_{p,t}^{(l)}]=f_p^{(l)}(\mathbf{z}_t^{(l+1)},\mathbf{d}_{t-1}^{(l)}) \quad (4)$$

is calculated by a fully connected (FC) network $f_p^{(l)}(\cdot)$ with parameter ω using $\mathbf{d}_{t-1}^{(l)}$ at time $t-1$. Notably, latent code $\mathbf{z}$ is computed in a top-down order in different layers while the dilation embedding $\mathbf{d}$ is calculated in a bottom-up order. The reason behind this property is to reliably embed a latent variable $\mathbf{z}$ with the rough global feature in higher layer in the beginning and then with the delicate local feature in lower layer in learning procedure.

Variational inference is introduced to infer a latent variable model where a variational posterior of $\mathbf{z}_t$ is calculated for prediction by using not only historic samples $\mathbf{x}_{<t}=\{\mathbf{x}_i\}_{i=1}^{t-1}$ but also current sample $\mathbf{x}_t$, namely $\mathbf{x}_{\le t}$

$$q_\phi(\mathbf{z}_t|\mathbf{x}_{\le t})=q_\phi(\mathbf{z}_t^{(L)}|\mathbf{d}_t^{(L)})\prod_{l=1}^{L-1}q_\phi(\mathbf{z}_t^{(l)}|\mathbf{z}_t^{(l+1)},\mathbf{d}_t^{(l)}). \quad (5)$$

This is implemented by an *inference network* where the variational posterior is calculated as a Gaussian

$$q_\phi(\mathbf{z}_t^{(l)}|\mathbf{z}_t^{(l+1)},\mathbf{d}_t^{(l)})=\mathcal{N}(\boldsymbol{\mu}_{q,t}^{(l)},\boldsymbol{\sigma}_{q,t}^{(l)}) \quad (6)$$

with the parameters of mean and variance

$$[\boldsymbol{\mu}_{q,t}^{(l)},\boldsymbol{\sigma}_{q,t}^{(l)}]=f_q^{(l)}(\mathbf{z}_t^{(l+1)},\mathbf{d}_t^{(l)}) \quad (7)$$

where $f_q^{(l)}(\cdot)$ denotes a variational FC network where the input $\mathbf{d}_t^{(l)}$ at time t is used. Notably, the history posterior in Eq. (2) using previous sample $\mathbf{x}_{<t}$ is treated as the *prior* for this inference network as the variational posterior at current time with sample $\mathbf{x}_t$ in Eq. (5). Variational parameter or inference network parameter ϕ contains those parameters from 1-D convolution network and variational FC network. Both prior parameter ω and inference parameter ϕ are varied at each layer l. As a result, the evidence lower bound (ELBO) $\mathcal{L}$ of log conditional likelihood $\log p(\mathbf{y}|\mathbf{x})$ for sequence-to-sequence learning from $\mathbf{x}=\{\mathbf{x}_t\}$ to $\mathbf{y}=\{\mathbf{y}_t\}$ is formulated as

$$\begin{aligned}\log p(\mathbf{y}|\mathbf{x})\ge&\mathbb{E}_{q_\phi(\mathbf{z}_{\le T}|\mathbf{x}_{\le T})}\Bigg[\sum_{t=1}^{T}\log p_\theta(\mathbf{y}_t|\mathbf{x}_{\le t},\mathbf{z}_{\le t})\\&-\mathcal{D}_{\text{KL}}(q_\phi(\mathbf{z}_t|\mathbf{x}_{\le t})\|p_\omega(\mathbf{z}_t|\mathbf{x}_{<t}))\Bigg]\triangleq\mathcal{L}\end{aligned} \quad (8)$$

where $\mathcal{D}_{\text{KL}}(\cdot)$ denotes the Kullback-Leibler divergence. Here, the *generative network* for output sample $\mathbf{y}_t$ at each time t is calculated by using an RNN (or a long short-term memory (LSTM)) $f_\theta(\cdot)$ given by a hidden state $\mathbf{h}_t$ and an augmented input vector

$$\mathbf{z}_t=[(\mathbf{z}_t^{(1)})^\top\cdots(\mathbf{z}_t^{(L)})^\top]^\top$$

from the samples in different layers in inference distribution $p_\phi(\mathbf{z}_t|\mathbf{x}_{\le t})$. The generative distribution $p_\theta(\mathbf{y}_t|\mathbf{x}_{\le t},\mathbf{z}_{\le t})$ in decoder is driven by LSTM hidden state $\mathbf{h}_t$ and calculated by

$$p_\theta(\mathbf{y}_t|\mathbf{h}_t)=\text{Mult}(f_\theta),\quad \text{where}\ \ \mathbf{h}_t=f_\theta(\mathbf{z}_t,\mathbf{h}_{t-1}) \quad (9)$$

where a multinomial probability is continuously updated for predicting each output word or target $\mathbf{y}_t$ at time t based on a hidden state $\mathbf{h}_t$ updated by $f_\theta(\cdot)$ using the input from stochastic state of TCN encoder $\mathbf{z}_t$. LSTM decoder contains parameter θ. A stochastic hybrid TCN and LSTM is implemented according to the stochastic gradient descent algorithm using the gradients $\frac{\partial\mathcal{L}}{\partial\omega}$, $\frac{\partial\mathcal{L}}{\partial\phi}$ and $\frac{\partial\mathcal{L}}{\partial\theta}$ for updating the parameters of prior network, inference network and generative network, respectively.

5. Experiments

The proposed stochastic convolutional recurrent network is evaluated for word prediction in language model and speech recognition as illustrated in the following experiments.

5.1. Experimental Setup

Three benchmark datasets were collected in evaluation of language models (LMs) [27–29]. The first dataset was the Penn Treebank (PTB) [30] which consisted of 929K training words, 73K validation words and 82K test words with 10K words in its vocabulary. This dataset was preprocessed by lowercasing the capital letters, removing the punctuations and replacing the numbers with a placeholder. The second dataset was sampled from the 1B-Word-Benchmark (1BWB) corpus consisting of 100 partitions [31, 32]. The vocabulary size was 191K words. One partition with 8M words (held-out set) was split into 50 disjoints to be used as the validation data and test data. Validation data were used to tune hyperparameters in different methods. There were 20 partitions from the other 99 disjoint partitions randomly sampled as the training data. Each partition was used to estimate an individual LM. Perplexity was measured by averaging over those from 20 models. The third dataset was the Wall Street Journal (WSJ) corpus for evaluation of continuous speech recognition (CSR) where LM was combined with acoustic model for sequential prediction [33]. The WSJ0 (SI-84) training set was adopted to estimate the acoustic models. Training data consisted of 7138 utterances from 83 speakers. The 1987-1989 WSJ corpus with 38M words was adopted to train LMs. A total of 330 and 333 utterances were sampled from the November 1992 ARPA CSR benchmark test data with vocabulary sizes ($|\mathcal{V}|$) of 5K and 20K, respectively. These utterances were used to examine the model perplexity and word error rate (WER). For comparison, LSTM (here denoted as RNN), TCN [8, 9], stochastic TCN (STCN) [22], CRN and stochastic CRN (SCRN) were implemented. In the implementation, each input sequence or sentence was trimmed to length 20 in order to learn long sequence in an efficient way. To remain the sequence information between individual minibatches, the hidden state $\mathbf{h}_0$ was initialized to zero in the first epoch and the last state of the previous epoch $\mathbf{h}_T$ was used as the initial state of next epoch $\mathbf{h}_0$ for the rest of minibatches. Each model was trained by stochastic gradient ascent algorithm [34–36] with an initial learning rate which was decreased by a factor of 2 when the validation loss started to increase. The mini-batch size was twenty. Gradient-clipping was applied with maximum norm 0.25. Initial values in LSTM states were zero. All parameters were initialized by a uniform distribution between [-1,1]. Recurrent dropout was used in each layer's outputs with a dropout rate 0.5 [37, 38]. The size of hidden states and the amount of kernels were adjusted in different models to obtain desirable performance using validation data. Model size and training time were included in the evaluation.

Table 1: *Perplexity (in three phases) and model size by using different combinations of TCN and RNN.*

Model	#Params	Train	Validation	Test
RCN	9.5M	125	129	125
RRN	10.4M	90	126	122
CRN	11.1M	79	119	**116**

Table 2: *Perplexity (in three phases) and model size using PTB.*

Model	#Params	Train	Validation	Test
RNN	8.8M	101	128	123
TCN	7.0M	100	138	131
STCN	13.1M	97	125	120
CRN	11.1M	79	119	116
SCRN	17.6M	69	113	**109**

5.2. Experimental Results

We claim that it is beneficial to encode short-term information using TCN earlier than long-term information using RNN. It is important to evaluate different two-stage architectures under comparable model size. In this comparison, the recurrent convolutional network (denoted as RCN) is carried out as RNN encoder and TCN decoder. The order of TCN and RNN is swapped. The recurrent recurrent network (denoted as RRN) implements a 2-layer RNN or RNN with two recurrent layers. The results of perplexity in training, validation and test phases using RCN, RRN and CRN are compared in Table 1. CRN obtains the lowest perplexity among three methods over three phases. The most significant improvement is attained in training phase. Interestingly, RRN obtains lower perplexities than RCN. These results confirm the selection of model architecture with TCN encoder and RNN decoder. The strengths of TCN and RNN are complementary. CRN boosts the advantages of TCN and RNN, so its modeling ability is better than both of them. The integrated TCN and RNN decreases the perplexity of language model. On the other hand, Table 2 compares different models in terms of model size and perplexity in three phases. CRN captures the temporal dependencies better than RNN (or LSTM) and TCN. TCN obtains higher perplexity but consumes smaller number of model parameters than RNN. This is due to the convolution calculation in TCN. In addition, STCN and SCRN perform better than TCN and CRN in perplexity, respectively. But, the model size is increased as well due to additional memory consumption from prior network and inference network in STCN and SCRN. With the stochastic property, SCRN outperforms the other models in training, validation and test phases. SCRN is more robust and accurate in word prediction than CRN. SCRN works better than STCN due to the hybrid architecture in sequential learning.

Language models using different sequential machines are further investigated by using the tasks of 1BWB and WSJ. Table 3 compares different LMs in terms of model size, training time (relative to RNN) and perplexity of test data using 1BWB dataset. The variety of natural sentences in 1BWB dataset is much richer than that of PTB dataset. The amount of training

Table 3: *Perplexity, model size and training time using 1BWB.*

Model	#Params	Time	Test
RNN	72M	1x	145.3
TCN	69M	0.8x	148.2
STCN	143M	1.5x	133.8
CRN	129M	2.3x	125.7
SCRN	163M	2.9x	**108.3**

Table 4: *Perplexity and WER using WSJ.*

Model	Perplexity	WER (%)	
	$\lvert\mathcal{V}\rvert$=20K	$\lvert\mathcal{V}\rvert$=5K	$\lvert\mathcal{V}\rvert$=20K
RNN	119.4	4.33	11.9
TCN	118.1	4.28	12.0
STCN	100.8	3.85	11.3
CRN	97.3	3.29	10.7
SCRN	**89.3**	**2.91**	**9.6**

data and the number of vocabulary words in 1BWB are much larger than those in PTB. As we can see, the number of parameters in different models is greatly increased in 1BWB. The computation cost is closely related to the model size. TCN still performs worse than RNN in 1BWB task, but the hybrid architecture using CRN obtains better performance than individual architectures using TCN and RNN. The stochastic machines using STCN and SCRN still work better than the deterministic machines using TCN and CRN, respectively. In this set of experiments, the lowest perplexity is achieved by using the proposed SCRN. To see the effect of language models on speech recognition, Table 4 compares the perplexities and WERs of using different LMs under different vocabulary sizes ($|\mathcal{V}|$) where WSJ dataset is used. In this comparison, TCN and RNN have comparable performance in perplexity and WER. Again, the two-stage models using CRN and SCRN receive lower perplexity and WER than single-state models using RNN, TCN and STCN. The benefit of stochastic modeling in STCN and SCRN does help in this evaluation.

6. Conclusions

This paper proposed a new hybrid network architecture for sequential learning and language modeling. The proposed architecture integrated the complementary temporal features from temporal convolutional network and recurrent neural network. Short-term dependencies and long-term semantics in sequence data were represented in an integrated model. In particular, the probabilistic solution to this convolutional recurrent network was developed to enhance the robustness and the expressiveness in a sequential machine. This model characterized the temporal hierarchy with an extended receptive field by implementing variational inference via maximization over the evidence lower bound of log likelihood of training sentences. Experiments on different tasks illustrated the effectiveness of the proposed hybrid structure for word prediction in text modeling and speech recognition. The proposed hybrid architecture performed better than other architectures. Stochastic modeling did improve the performance of sequential machines. Future works include the extension to other sequential learning tasks in speech areas. Multi-scale temporal dependency with attention mechanism will be incorporated and investigated. The comparison with a fine-tuned language model based on the generative pretraining [39, 40] using the in-domain text will be conducted.

7. References

[1] D. Yu, G. Hinton, N. Morgan, J.-T. Chien, and S. Sagayama, "Introduction to the special section on deep learning for speech and language processing," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 4–6, 2012.

[2] J.-T. Chien and C.-H. Lee, "Deep unfolding for topic models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 2, pp. 318–331, 2017.

[3] J.-T. Chien and Y.-T. Bao, "Tensor-factorized neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1998–2011, 2018.

[4] J.-T. Chien, "Deep Bayesian natural language processing," in *Proc. of Annual Meeting of Association for Computational Linguistics : Tutorial Abstracts*, 2019.

[5] Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," *The Handbook of Brain Theory and Neural Networks*, vol. 3361, no. 10, p. 1995, 1995.

[6] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.

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

[8] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[9] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[10] J.-T. Chien and Y.-M. Huang, "Stochastic convolutional recurrent networks," in *Proc. of International Joint Conference on Neural Networks*, 2020.

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

[12] J.-T. Chien and C. Shen, "Stochastic recurrent neural network for speech recognition," in *Proc. of Annual Conference of International Speech Communication Association*, 2017, pp. 1313–1317.

[13] Y.-M. Huang, H.-H. Tseng, and J.-T. Chien, "Stochastic fusion for multi-stream neural network in video classification," in *Proc. of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 2019, pp. 69–74.

[14] M. Liang and X. Hu, "Recurrent convolutional neural network for object recognition," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3367–3375.

[15] R. Cui, H. Liu, and C. Zhang, "Recurrent convolutional neural networks for continuous sign language recognition by staged optimization," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7361–7369.

[16] N. McLaughlin, J. Martinez del Rincon, and P. Miller, "Recurrent convolutional network for video-based person re-identification," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1325–1334.

[17] X. Wang, W. Jiang, and Z. Luo, "Combination of convolutional and recurrent neural network for sentiment analysis of short texts," in *Proc. of International Conference on Computational Linguistics: Technical Papers*, 2016, pp. 2428–2437.

[18] X. Zhang, F. Chen, and R. Huang, "A combination of RNN and CNN for attention-based relation classification," *Procedia Computer Science*, vol. 131, pp. 911–917, 2018.

[19] S. Chang, Y. Zhang, W. Han, M. Yu, X. Guo, W. Tan, X. Cui, M. Witbrock, M. A. Hasegawa-Johnson, and T. S. Huang, "Dilated recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 77–87.

[20] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," in *Advances in Neural Information Processing Systems*, 2015, pp. 2980–2988.

[21] G. Lai, B. Li, G. Zheng, and Y. Yang, "Stochastic wavenet: A generative latent variable model for sequential data," *arXiv preprint arXiv:1806.06116*, 2018.

[22] E. Aksan and O. Hilliges, "STCN: Stochastic temporal convolutional networks," *arXiv preprint arXiv:1902.06568*, 2019.

[23] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proc. of International Conference on Learning Representation*, 2016.

[24] K. Cho, B. Van M., C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[25] C. Tallec and Y. Ollivier, "Can recurrent neural networks warp time?" *arXiv preprint arXiv:1804.11188*, 2018.

[26] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther, "Ladder variational autoencoders," in *Advances in Neural Information Processing Systems*, 2016, pp. 3738–3746.

[27] J.-T. Chien, "Association pattern language modeling," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1719–1728, 2006.

[28] J.-T. Chien and C.-H. Chueh, "Joint acoustic and language modeling for speech recognition," *Speech Communication*, vol. 52, no. 3, pp. 223–235, 2010.

[29] J.-T. Chien, "Hierarchical Pitman-Yor-Dirichlet language model," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 8, pp. 1259–1272, 2015.

[30] T. Mikolov and G. Zweig, "Context dependent recurrent neural network language model," in *Proc. of IEEE Workshop on Spoken Language Technology*, 2012, pp. 234–239.

[31] C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, P. Koehn, and T. Robinson, "One billion word benchmark for measuring progress in statistical language modelling," in *Proc. of Annual Conference of International Speech Communication Association*, 2014, pp. 2635–2639.

[32] 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.

[33] J.-T. Chien and C.-Y. Kuo, "Markov recurrent neural network language model," in *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop*, 2019, pp. 807–813.

[34] H. Robbins and S. Monro, "A stochastic approximation method," *The Annals of Mathematical Statistics*, pp. 400–407, 1951.

[35] J. Kiefer, J. Wolfowitz *et al.*, "Stochastic estimation of the maximum of a regression function," *The Annals of Mathematical Statistics*, vol. 23, no. 3, pp. 462–466, 1952.

[36] 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.

[37] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 1019–1027.

[38] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[39] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[40] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019.