Feedforward Sequential Memory Networks based Encoder-Decoder Model for Machine Translation

Junfeng Hou*, Shiliang Zhang*, Lirong Dai*, Hui Jiang†
* National Engineering Laboratory for Speech and Language Information Processing
University of Science and Technology of China, Hefei, Anhui, P. R. China
† Department of Electrical Engineering and Computer Science, York University, Toronto, Ontario, Canada
E-mail: {hjf176,zsl2008}@mail.ustc.edu.cn, lrdai@ustc.edu.cn, hj@cse.yorku.ca

***Abstract*—Recently recurrent neural networks based encoder-decoder model is a popular approach to sequence to sequence mapping problems, such as machine translation. However, it is time-consuming to train the model since symbols in a sequence can not be processed parallelly by recurrent neural networks because of the temporal dependency restriction. In this paper we present a sequence to sequence model by replacing the recurrent neural networks with feedforward sequential memory networks in both encoder and decoder, which enables the new architecture to encode the entire source sentence simultaneously. We also modify the attention module to make the decoder generate outputs simultaneously during training. We achieve comparable results in WMT'14 English-to-French translation task with 1.4 to 2 times faster during training because of temporal independency in feedforward sequential memory networks based encoder and decoder.**

I. Introduction

Encoder-decoder model belongs to the end-to-end framework and is very popular and successful recently. Equipped with the powerful recurrent neural networks (RNNs) and attention mechanism, it yields competitive or state-of-the-art results in tasks like machine translation [1][2][3], image caption [4][5], speech recognition [6][7] and so on. Long short-term memory networks (LSTMs) [8][9] and gated recurrent units (GRUs) [10] are mostly used RNNs architectures for encoder-decoder models. Meanwhile deep convolutional neural networks (CNNs) have been used as encoder together with RNNs as decoder [11][12].

In the encoder-decoder model, the encoder can be treated as a memory encoding module to represent a word accompanying with different context. And the decoder can be treated as a language model and predicts output words based on the generated output history as well as the attended glimpse of the encoder sequence. Models like CNNs or RNNs can handle variable length sequences and memorize the context information of each word in the sequence so that they can be used as encoder model. However most encoder-decoder models employ RNNs as decoder while other powerful language models like CNNs [13] are rarely taken into account.

Recently CNNs and self-attention are adopted as both encoder and decoder in encoder-decoder model in [14] and [15] respectively which are very similar to our work. In their works, the models achieved better results as well as faster training and evaluation speed than RNNs based model in tasks like machine translation and summarization. While in our work, we employ different encoding and decoding modules.

In this paper, we use a novel sequence encoding model named as feedforward sequential memory networks (FSMNs) [16][17][18] to replace the RNNs model in both encoder and decoder module of the end-to-end framework. The FSMN model is a standard feedforward neural network with single or multiple memory blocks in hidden layers and can learn long-term dependency in language model [16] as well as speech recognition [17]. On account of the ability of memorizing the context information of a word, FSMN is used as encoder model. What's more, we also modify the attention module so that FSMN can be employed as decoder model and generates output symbols simultaneously during training. The FSMN based encoder-decoder model can be trained fast because of no recurrent connections. In the experiments we can achieve comparable results in WMT'14 English-to-French translation task and 1.4 to 2 times faster during training than RNN based encoder-decoder model.

II. Methods

A. Preliminaries: Feedforward Sequential Memory Networks

Feedforward sequential memory network (FSMN) is a standard feedforward neural network with single or multiple memory blocks in hidden layers. For instance, Fig. 1 shows a FSMN model with its first hidden layer equipped with one memory block. Given a sequence $\mathbf{X} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_T}\}$ where each $\mathbf{x_t} \in \mathbb{R}^D$ represents the input vector at time t, the corresponding hidden layer outputs are denoted as $\mathbf{H} = \{\mathbf{h_1^1}, \mathbf{h_2^1}, \ldots, \mathbf{h_T^1}\}$. Then we can use a vector $\mathbf{a}$ to encode $\mathbf{h_t^1}$ and its previous N histories into a fixed-sized representation $\tilde{\mathbf{h_t^1}}$ (called an N-th order FSMN) in the memory block (see (1)). As shown in Fig. 1, $\tilde{\mathbf{h_t^1}}$ can be fed into next hidden layer in the same way as $\mathbf{h_t^1}$. As a result, the activation of next hidden layer can be calculated by (2).

$$\tilde{\mathbf{h_t^1}} = f\left(\sum_{i=0}^{N} a_i \cdot \mathbf{h_{t-i}^1}\right) \quad (1)$$

$$\mathbf{h_t^2} = f\left(W\mathbf{h_t^1} + \tilde{W}\tilde{\mathbf{h_t^1}} + b\right) \quad (2)$$

where the vector $\mathbf{a} = a_0, a_1, \cdots, a_N$ are learnable coefficients, and $f()$ is activation function. The memory representations $\tilde{\mathbf{h_t^1}}$ are computed with zeros paddings when $t \leq N$.

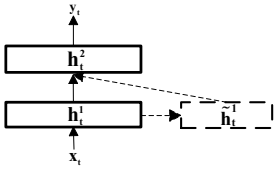

Fig. 1. Feedforward sequential memory networks (FSMN) with single memory block. The memory blocks $\tilde{\mathbf{h}}_\mathbf{t}^\mathbf{1}$ and $\mathbf{h}_\mathbf{t}^\mathbf{2}$ are calculated simultaneously for each input $\mathbf{x_t}$. As there is no recurrent connection in FSMN, the gradient for $\mathbf{h}_\mathbf{t}^\mathbf{1}$ can also be computed simultaneously for each time.

Equation (1) is named as scalar FSMN. The $f()$ function is an identity function in [16][17][18] for simplicity while we use rectified linear (ReLU) activation function here. More powerful models like vector FSMN are also introduced in [16][17][18] but we only adopt scalar FSMN here.

The hidden units $\mathbf{h}_\mathbf{t}^\mathbf{2}$ can be regarded as a fixed-size representation of the long input history before time t, which is similar to RNNs' hidden states. Considering that, we propose a new encoder-decoder model based on FSMNs. The detail equations can be seen in section II-C.

B. FSMN based Encoder-Decoder Model

A natural way to apply FSMN to encoder-decoder model is shown in Fig. 2. We only present the FSMN based encoder-decoder model with single memory block here for simplicity. More than one memory blocks in FSMN encoder or decoder can be used similar to FSMN models in [16][17][18]. Since compressing all the necessary information of a source sentence into a fixed-length vector is difficult, soft attention mechanism is introduced to the encoder-decoder model. We adopt attention module in the last hidden layer of the decoder and only one layer attention is used which is different from [14]. All the hidden states for each input are in a memory bank denoted as $\{\mathbf{h}_\mathbf{1}^\mathbf{2}, \mathbf{h}_\mathbf{2}^\mathbf{2}, \cdots, \mathbf{h}_\mathbf{T}^\mathbf{2}\}$. When decoding, the model selects useful terms in the bank based on the current context and combines them into a context vector $\tilde{\mathbf{H}}_\mathbf{t}$. The attention weights (shown in blue lines) in time t are depend on the encoder memory bank $\{\mathbf{h}_\mathbf{1}^\mathbf{2}, \mathbf{h}_\mathbf{2}^\mathbf{2}, \cdots, \mathbf{h}_\mathbf{T}^\mathbf{2}\}$ and decoder history $\tilde{\mathbf{v}}_\mathbf{t}^\mathbf{1}$ and $\mathbf{v}_\mathbf{t}^\mathbf{1}$. $\tilde{\mathbf{H}}_\mathbf{t}$, $\tilde{\mathbf{v}}_\mathbf{t}^\mathbf{1}$ and $\mathbf{v}_\mathbf{t}^\mathbf{1}$ are combined to predict the next output word.

To predict the output words simultaneously during training, both encoder and decoder must process sequences parallelly. Obviously, FSMN based encoder can process input word sequence simultaneously. Then the attention module should generate alignments concurrently for each output word. The key point to make it happen is that the generated alignments should have no time dependency and not be related to each other. With these changes, the decoder can generate output sequence simultaneously as well and a loop-free encoder-decoder model is obtained. Because the model is kind of a feedforward network with no recurrent connections, we don't need to use gradient-based back-propagation through time (BPTT) technique when training the model, which makes the training process faster than RNN based encoder-decoder model. Details and equations about the model with single memory block are presented in section II-C.

C. Architecture of FSMN based Encoder-Decoder Model

The hidden states for both encoder and decoder are obtained from FSMN. Equations with single memory block are given below.

$$\mathbf{h}_\mathbf{t}^\mathbf{1} = f(W_h \mathbf{x_t} + b_h) \quad (3)$$

$$\mathbf{v}_\mathbf{t}^\mathbf{1} = f(W_v \mathbf{y_t} + b_v) \quad (4)$$

The memory block for encoder and decoder is calculated as:

$$\tilde{\mathbf{h}}_\mathbf{t}^\mathbf{1} = f\left(\sum_{i=0}^{N} a_i \cdot \mathbf{h}_{\mathbf{t-i}}^\mathbf{1}\right) \quad (5)$$

$$\mathbf{h}_\mathbf{t}^\mathbf{2} = f\left(W\mathbf{h}_\mathbf{t}^\mathbf{1} + \tilde{W}\tilde{\mathbf{h}}_\mathbf{t}^\mathbf{1} + b\right) \quad (6)$$

$$\tilde{\mathbf{v}}_\mathbf{t}^\mathbf{1} = f\left(\sum_{i=0}^{N} b_i \cdot \mathbf{v}_{\mathbf{t-i}}^\mathbf{1}\right) \quad (7)$$

$$\mathbf{v}_\mathbf{t}^\mathbf{2} = f\left(W_{v'}\mathbf{v}_\mathbf{t}^\mathbf{1} + \tilde{W}_v\tilde{\mathbf{v}}_\mathbf{t}^\mathbf{1} + W_H\tilde{\mathbf{H}}_\mathbf{t} + b_v\right) \quad (8)$$

The encoder context $\tilde{\mathbf{H}}_\mathbf{t}$ in (8) is ignored if there is no attention module in this decoder layer. The attention weight for time t in the decoder is calculated as:

$$e_{i,t} = f_{\text{score}}(\tilde{\mathbf{v}}_\mathbf{t}^\mathbf{1}, \mathbf{v}_\mathbf{t}^\mathbf{1}, \mathbf{y_{t-1}}, \mathbf{h_i^2}) \quad (9)$$

$$\alpha_{i,t} = \frac{\exp(e_{i,t})}{\sum_{j=1}^{T}\exp(e_{j,t})} \quad (10)$$

where the score function $f_{\text{score}}()$ maps its input to a scalar value. We employ a multiplicative [2] attention mechanism and the score function is given below:

$$e_{i,t} = \tilde{\mathbf{v}}_\mathbf{t}^\mathbf{1} U_e^T \mathbf{h_i^2} + \mathbf{v}_\mathbf{t}^\mathbf{1} W_e^T \mathbf{h_i^2} \quad (11)$$

The reason we use both $\tilde{\mathbf{v}}_\mathbf{t}^\mathbf{1}$ and $\mathbf{v}_\mathbf{t}^\mathbf{1}$ to calculate the attention is that $\tilde{\mathbf{v}}_\mathbf{t}^\mathbf{1}$ contains little information of the current word based on the FSMN equation in (8) and $\mathbf{v}_\mathbf{t}^\mathbf{1}$ must be used to emphasize the current input. The encoder context for time t is the weighted average of all the representation $\mathbf{h_i^2}$:

$$\tilde{\mathbf{H}}_\mathbf{t} = \sum_{i=1}^{T} \alpha_{i,t} \mathbf{h_i^2} \quad (12)$$

As equations shown above, both encoder and decoder including attention module can be computed simultaneously which can speed up the training of proposed encoder-decoder model.

III. Experiments

We conduct machine translation in WMT'14 English-to-French translation task, which contains 12M parallel sentence pairs. The datasets and output vocabulary size are the same with [1]. We also train the proposed models with sentence length up to 30 and 50 respectively (named as FSMNsearch-30 and FSMNsearch-50). We compare our model with the conventional GRU based encoder-decoder model in [1] and focus on the generic attention models with FSMN component.

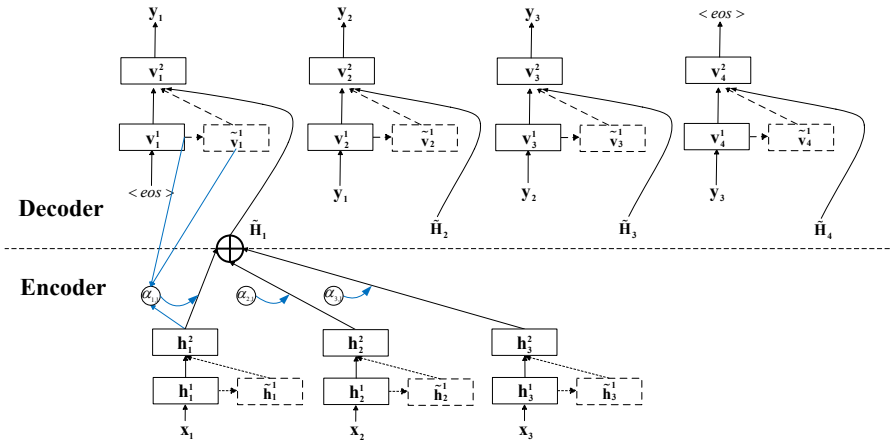

Fig. 2. FSMN based encoder-decoder model with single memory block. The FSMN based encoder $\tilde{\mathbf{h}}_t^1$ and $\mathbf{h}_t^2$ can be calculated simultaneously as usual. During training, the memory blocks of FSMN based decoder $\tilde{\mathbf{v}}_t^1$ for each time can also be obtained at once. As there is no time dependency for each time in decoder, the attention vector as well as the output word distribution for each time are obtained simultaneously.

So the recent progress on aggregating multiple models or enlarging the vocabulary [3] are not considered here.

For the FSMN based model, the encoder contains 4 hidden layers and is equipped with two memory blocks in the second and third layer respectively. All the hidden units adopt the rectified linear (ReLU) activation function. Inspired by the bidirectional LSTM, we have two encoder networks for the two directional encoding. The input of first encoder network is $\{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_T}\}$ with natural sequence order while another encoder network with $\{\mathbf{x_T}, \mathbf{x_{T-1}}, \ldots, \mathbf{x_1}\}$ as input. The two directional representations are concatenated together as the final encoder representation for decoding. The filter order N is set to 10 for FSMNsearch-30 models and 25 for FSMNsearch-50 models. Bidirectional FSMN is introduced in [16][17][18], but we don't use it in this work. As FSMN is a feedforward neural network, we can use more than one words as input which is similar to feedforward neural network based language model. In this work, the input window contains one or two words.

We use SGD to train the FSMN based model. The learning rate for all the parameters is 0.1 except that the learning rate for the FSMN filter is 0.001. Momentum and dropout are not used. All the experiments are conducted on single K40 GPU and the models are implemented with theano[19] based on the open source code[1].

The results are given in table I and the training speed is given in table II. From the result in table I we can see that FSMN based model can handle the machine translation task and yield comparable results to RNN based model. The input window with two words (named as FSMNsearch-30-2word) yields better BLEU result than one word (named as FSMNsearch-30-1word). What's more, the training speed is about 2 times faster (see table II) than RNN based model in our implementation benefiting from the concurrency of FSMN.

[1]https://github.com/nyu-dl/dl4mt-tutorial

We also noticed that in [14] CNN is much faster than RNN in sequence to sequence model. And we consider it is mainly due to our implementation efficiency that the FSMN based model yields marginal acceleration ratio.

TABLE I
BLEU SCORES ON THE TEST SET FOR DIFFERENT MODELS

Model	Test BLEU
RNNsearch-30 [1]	21.50
RNNsearch-50 [1]	28.45
FSMNsearch-30-1word	23.04
FSMNsearch-30-2word	23.47
FSMNsearch-50-2word	28.74

TABLE II
TRAINING TIME FOR DIFFERENT MODELS

Model	hour/epoch
RNNsearch-30	23.9
RNNsearch-50	52.5
FSMNsearch-30-2word	11.7
FSMNsearch-50-2word	33.1

We also visualize the FSMN encoding vectors $\mathbf{a}, \mathbf{b}$ in the last memory block (see (5),(7)) in encoder and decoder layer in Fig. 3. The learned FSMN filters are similar to [16][17][18], which means that the further a word is away from the current central word, the less important the word is for current central word representation and translation. The first value of the vector $\mathbf{b}$ is small which explains why we use both $\tilde{\mathbf{v}}_t^1$ and $\mathbf{v}_t^1$ to calculate the attention.

The attention of FSMN encoder is given in Fig. 4. The FSMN based model with input window containing one word (see the top figure in Fig. 4) can generate similar alignments with RNN based model, which proves the effectiveness of the FSMN based model. As for the model with input window containing two words (see the bottom figure in Fig. 4), it is reasonable that the alignments are one-word mismatch.

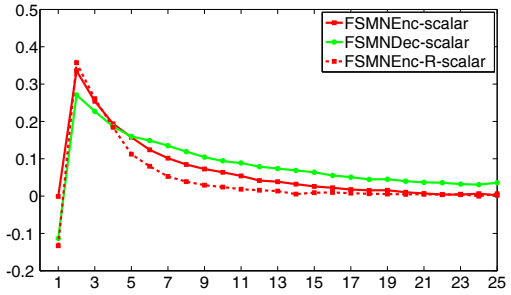

Fig. 3. Illustration of the learned filters in scalar FSMN for both encoder and decoder. 'FSMNEnc-scalar' means the forward encoder vector. 'FSMNEnc-R-scalar' means the reverse encoder vector. 'FSMNDec-scalar' means the decoder vector.

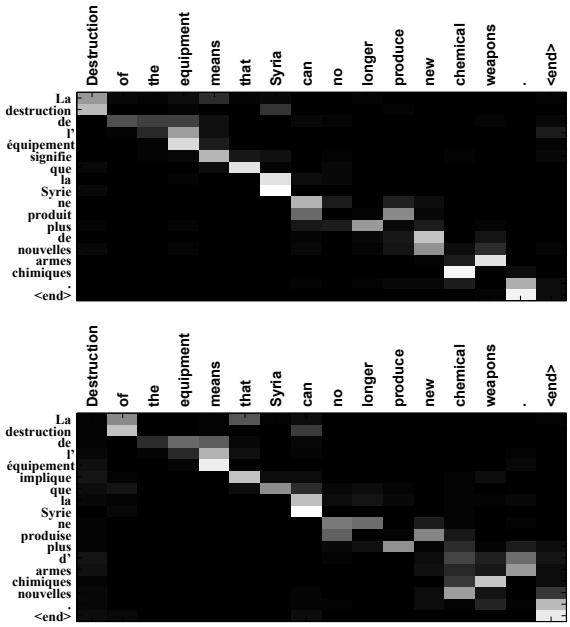

Fig. 4. top) Alignment found by FSMNSearch-30-1word with input window containing one word. bottom) Alignment found by FSMNSearch-50-2word with input window containing two words. The x-axis and y-axis of each plot correspond to the words in the source sentence (English) and the generated translation (French) respectively. The input sentence is same as Fig. 3(c) in [1] so as to compare the models.

IV. Conclusions and Future Work

In this paper, we replace all the RNN models used in encoder-decoder model with FSMN and apply it to machine translation task. The performance of the FSMN based model is comparable to the conventional attention based encoder-decoder model as FSMN contains memory block to model the long term dependency. Meanwhile, the training speed of the FSMN based model is faster because of the non-recurrent architecture. In our experiments, we prove that the RNNs in encoder-decoder model can be replaced by FSMN to some extent. In the future, we will consider more powerful FSMN models like vector FSMN and other technologies employed in RNN-based encoder-decoder model.

V. Acknowledgment

The authors would like to acknowledge the support of National Key Research and Development Program(Grant No: 2017YFB1002200).

References

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

[2] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015.

[3] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[4] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3156–3164.

[5] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention." in *ICML*, vol. 14, 2015, pp. 77–81.

[6] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in Neural Information Processing Systems*, 2015, pp. 577–585.

[7] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 4960–4964.

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

[9] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," *Neural Computation*, vol. 12, pp. 2451–2471, 1999.

[10] K. Cho, B. van Merriënboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Oct. 2014, pp. 1724–1734.

[11] F. Meng, Z. Lu, M. Wang, H. Li, W. Jiang, and Q. Liu, "Encoding source language with convolutional neural network for machine translation," *arXiv preprint arXiv:1503.01838*, 2015.

[12] J. Gehring, M. Auli, D. Grangier, and Y. N. Dauphin, "A convolutional encoder model for neural machine translation," *arXiv preprint arXiv:1611.02344*, 2016.

[13] N.-Q. Pham, G. Kruszewski, and G. Boleda, "Convolutional neural network language models," in *Proc. of EMNLP*, 2016.

[14] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," *arXiv preprint arXiv:1705.03122*, 2017.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017.

[16] S. Zhang, C. Liu, H. Jiang, S. Wei, L. Dai, and Y. Hu, "Feedforward sequential memory networks: A new structure to learn long-term dependency," *arXiv preprint arXiv:1512.08301*, 2015.

[17] S. Zhang, H. Jiang, S. Xiong, S. Wei, and L. Dai, "Compact feedforward sequential memory networks for large vocabulary continuous speech recognition," in *Proc. Interspeech*, 2016, pp. 3389–3393.

[18] S. Zhang, C. Liu, H. Jiang, S. Wei, L. Dai, and Y. Hu, "Nonrecurrent neural structure for long-term dependence," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 4, pp. 871–884, 2017.

[19] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: A cpu and gpu math compiler in python," in *Proc. 9th Python in Science Conf*, 2010, pp. 1–7.