Implementation of Recurrent Neural Network with Sequence to Sequence Model to Translate Language Based on TensorFlow

Symphorien Karl Yoki Donzia[1], Haeng Kon Kim[2]

***Abstract*—Recently, many recurrent neural network based LM, a type of deep neural network for dealing with sequential data, have been proposed and achieved remarkable results. Most deep learning frame works, support the GPU to form fast models; in particular the execution of these models on several GPUs. In this work, an automatic learning algorithm will developed and proposed to address a deep neural network with a convolutional layer and a connected layer. The proposed algorithm is an extension of Ensemble's approach and uses a multilayer perceptron for data points, and a multilayer perceptron to combine the experts and predict the end result. In order to solve this problem, in this paper, we propose a CNN based language model that deals with textual data regarding a multi-dimensional data with respect to the input of the network. To train this dimensional input to Long-Short Term Memory (LSTM), we use a convolutional neural network (CNN) for dimensionality reduction of input data to avoid the vanishing gradient problem by decreasing the time step between input words. The dataset to train and test this model was taken from the META data set database with low velocity than MNIST dataset which is the object of our future work. Our implementations can be used for fast exhaustive search in Recurrent Neural Network, which can find a textual data regarding a multi-dimensional data by using Python 3.6. to get better performance.**

***Index Terms*—Language Modeling, Neural Language Modeling, Deep Learning, Neural Network, GPU**

I. Introduction

In recent years, deep learning has played a vital role in artificial intelligence and has also been successfully applied in many fields. For example, AlphaGo [1], developed by Google DeepMind, has achieved significant success in the game Go to beat the best human game Go players. In general, machine learning models are classified into two groups: supervised learning and unsupervised learning. A supervised learning model involves learning a function derived from the labeled learning data. The labeled learning data consists of a set of learning examples, and each example has an input value and an output value, also called a label. The learned function is used to correctly determine class labels for unknown data. In contrast to supervised learning approaches, unsupervised machine learning approaches are used to uncover unlabeled learning data patterns[2]. Deep learning involves comprise manifold platform of performance which helps to comprehend data like images, audio, and text. The idea of deep learning comes from the study of the artificial neural network, Multilayer Perceptron which includes more hidden layers which is a deep learning structure. [3].

Currently, graphics processing units (GPUs) have evolved from fixed function representation devices to programmable and parallel processors. Market demand for real-time high-definition 3D graphics is pushing GPUs to become multicore, highly parallel, multithreaded processors with huge computing power and high bandwidth memory. As a result, the GPU architecture is designed so that more transistors are dedicated to data processing than to caching data.[2] GPU-accelerated LSMs may be more computationally efficient than CPU-based LSMs. In addition, it is a major problem to make the LSM algorithms in the GPU optimized for the best efficiency. One of the main problems of metaheuristics is to rethink the existing parallel models and the programming paradigms to enable their implementation in GPU accelerators.[2] Deep Neural Network (or Deep Learning) is one of the machine learning algorithms that uses a cascade of multi-layers composed of a number of neurons and non-linear functionality units for prediction, classification, feature extraction, and pattern recognition [4]. Recently, deep neural network has achieved remarkable results in computer vision, natural language processing, speech recognition, and language modeling. Especially, Long-Short Term Memory (LSTM) [5], a type of recurrent neural network, is designed to process sequential data by memorizing previous input of the network, and LSTM is more robust to the vanishing and exploding gradient problem [6] than transitional recurrent neural network.

Fig.1. Generating Image Descriptions

[1] Dept. of Computer and Information Engineering Daegu Catholic University, Hayang-up, Gyeongsansi Gyeongsangbuk-do, Korea, yoki10@cu.ac.kr
[2] Dept. School of IT Engineering, Daegu Catholic University, Hayang-up, Gyeonsan-si Gyeongsangbuk do, Korea, hangkon@cu.ac.krl

ISBN: 978-988-14048-1-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

With convolutional neural networks, RNNs have been used as part of a model to generate descriptions of untagged images. It's pretty amazing how it seems to work. The combined model aligns even the words generated with the characteristics found in the images.

We developed the TensorFlow system to experiment with new models, train them in large datasets and put them into production. We have establishedTensorFlow on many time of undergo with our first generation system, DistBelief [7], which generalizes to permit researchers to examine a large assortment of concept with relative comfort. TensorFlow favor great-scale training and implication: it efficiently uses hundreds of potent servers (GPUs) for rapid training and implement production implication models on distinctive platform ,varies from large clusters issued through multiple programming. A data center that runs locally on mobile devices. In the same time, it is supple to mainstay experimentation and the review for new machine learning models and system amelioration.

A. *Background & motivation*

Machine translation is similar to linguistic modeling since our input is a sequence of words in our source language (English). We want to generate a sequence of words in our target language (Korean). In this study, we explore, evaluate and analyze the influence of RNN architectures, the characteristics of data sets. The formation of an RNN is similar to the formation of a traditional neural network. We also use the backpropagation algorithm, but with a small twist. As the parameters are mutual by all the time level of the connection, the gradient in each outflow cpmfode not even on the calculations of the current time level, but on the precedent time level. For example, to calculate the gradient at t = 4, we would need to go back three steps and summarize the gradients. This is called Backpropagation Through Time (BPTT). If that does not make sense yet, do not worry, we'll have a full article on the bloody details. There are certain mechanisms to deal with these problems, and some types of RNNs (such as LSTM) have been specifically designed to avoid them. The learning algorithm can read a batch of input data and current parameter values and rewrite the gradients in the parameter server. This model works for the formation of simple feed-forward neural networks, but fails for more advanced models, such as recurrent neural networks, which contain loops [8]; Adverse networks, in which two related networks are formed alternately [9]; and reinforcement learning models, where the loss function is calculated by an agent in a separate system, such as a video game emulator [10]. In addition, there are many other machine learning algorithms, such as maximizing expectations, learning decision forests and latent Dirichlet allocation, which do not fit the same mold as neural network training.

B. *Review of the Related Literature*

The network was made by Specht in 1991 [11], which showed that neural networks allow smooth transitions from one observed value to another, and therefore, may provide better results than conventional regression. Pal et al [8] further investigated the use of Perceptions Multi-Layer (MLP) for fuzzy classification. The additional literature implements multilayer perceptions for the purpose of predicting.

C. *Neural Network-based Language Model*

Language models based on neural networks worked better than transient language models based on numbers. One of the important techniques that leads to the successful application of deep learning in LM is the incorporation of words. The incorporation of words teaches the representation of words according to their context, which is the word that surrounds the target word and converts the text into real vectors. Since each neural network model only takes the number as input, the inclusion of words for each neural network model based on NLP is required. Word2Vec [12] is a well-known word integration technique that uses a continuous word bag (CBOW) or a skip-gram. In theory, RNNs can use information in arbitrarily long sequences, but in practice they are limited to only a few steps. [13]This is what a typical RNN looks like:

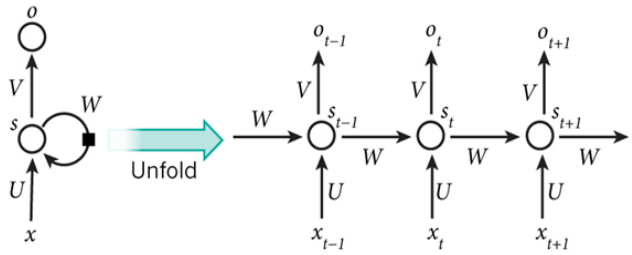

Fig.2. Forward Computation of RNN and Unfolding

The diagram above shows an RNN in a complete network. In the drop-down menu, we simply mean that we write the network for the complete sequence. For example, if the sequence of interest is a 5-word phrase, the network would unroll in a 5-layer neural network, one layer for each word. The formulas that govern the calculation in an RNN are the following: x_t is the entry to step t. For example, x_1 could be a one-to-one vector corresponding to the second word of a sentence. s_t is the hidden state in time step t. This is the "memory" of the network. s_t is calculated based on the previous hidden state and the input to the current step: s_t = f (Ux_t + Ws_ {t-1}). The function f is generally a non-linearity such as tanh or ReLU. s -1, which is required to calculate the first hidden state, is generally initialized to all zeros. o_t is the output in step t. For example, if we wanted to predict the next word in a sentence, it would be a vector of probabilities in our vocabulary. o_t = \ mathrm {softmax} (Vs_t).[14]

D. *Programming Languages*

R Language : Currently it is the most popular statistical programming language. The strength of the R language lies in obtaining sufficiently perceptible visualized data with simple coding. This helps shorten the development process. Python : It is the second most used language in the world after the R language, and helps to code the machine learning algorithms using the numpy library. Although it takes more time to code than the R language, it is used in several fields due to its portability, which is a typical language advantage.

ISBN: 978-988-14048-1-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

It can be used to calculate internal variables by adding a Scipy library or using Python to accelerate the algorithm. It is also easy to accelerate. Matlab [15]: Since mathematical accuracy is guaranteed to a certain extent, it is used mainly in the laboratory.

E. Divergence between Machine Learning and Deep Learning

Depth operation is a subtype of machine learning. When we use Machine Learning, manually extract the functions from the image. On the other hand, [15]it automatically provides the original image directly to the deep neural network that learns the functions. Deep Run often requires hundreds of thousands or millions of images to get better results. Deep execution requires a large amount of calculations and requires a high-performance graphics processor.

TABLE1
MACHINE LEARNING VS DEEP LEARNING

Machine Learning	Deep Learning
+ Small data sets can provide good results	- Large data set required
+ Model could be learned quickly	- Computationally intensive
- Multiple features and classifiers may try for the best results	+ Learn features and classifiers automatically
- Accuracy remains stable	+ Unlimited accuracy

F. TensorFlow Execution Model

TensorFlow uses a single data flow chart to represent all calculations and states in an automatic learning algorithm, which includes individual mathematical operations, parameters and their update rules, and preprocess inputs. The data flow chart explicitly expresses the communication between the subprocesses, which facilitates the execution of independent calculations in parallel and the division of the calculations between several devices. TensorFlow vary from unit data outflow systems in two [jase: a) The model backing multiple simultaneous performance in advanced imposed subgraphs of the global diagram. b) Individual vertices can have a mutable state that can be shared between different executions of the graph.The crucial study in the parameter server architecture is that the inconstant tange is crucial when traning very big models because it is possible perform elicit s to the site with very large parameters and propagate these updates to parallel learning stages as quickly as possible.[16].

II. IMPLEMENTATION

A. GRU-LSTM Network

According to the empirical evaluations during the demonstration in [16] of the empirical evaluation of synchronized recurrent neural networks in the modeling of sequences and the empirical exploration of recurrent network architectures, there is no clear winner. In many tasks, both architectures offer comparable performance, and adjustment hyperparameters such as the size of the layer are probably more important than choosing the ideal architecture. GRUs have fewer parameters and, therefore, can train a little faster or need less data to generalize. On the other hand, if you have enough data, the higher expression power LSTM can lead to better results.

.

B. Prosed Method (Sequence to Sequence Model)

The architecture of our proposed approach is applied to the LSTM network with an example sentence. In the model described above, each input must be encoded in a fixed-size status vector, since this is the only thing that is transmitted to the decoder. To allow the decoder to have more direct access to the input, a care mechanism has been introduced. Allows the decoder to take a look at the input in each decoding step. A multilayer network from sequence to sequence with LSTM cells and a mechanism of attention in the decoder resembles Fig.3.

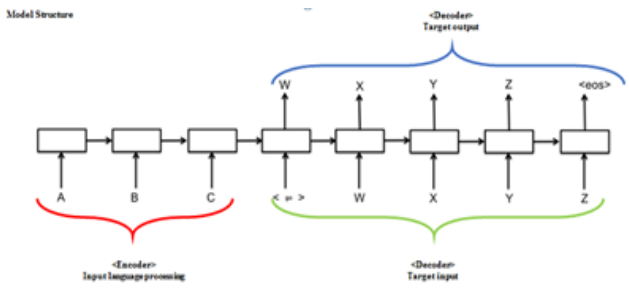

Fig.3. Design of Sequence to Sequence Model

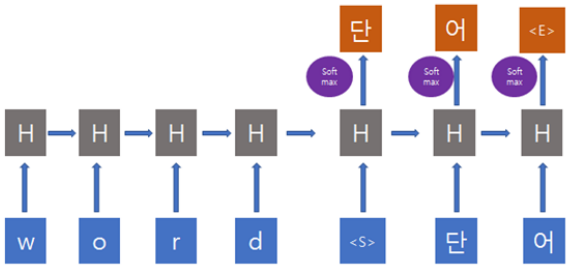

Fig.4. Proposal Architecture Sequence to Sequence Model

LSTM transform one word at a moment and calculates the probabilities of feasible values for the next word in the decision. Finally, the softmax layer is applied to the hidden representation of the LSTM to assign the probability distribution to the next word.

ISBN: 978-988-14048-1-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

C. *Sequence to Sequence Model Coding*

Purpose ; The main is to translate English word to Korean word. the parameters are distribute by all time phase in the network, the gradient at each output depends not only on the calculations of the current time step, but also the previous time steps our input is a sequence of words in our source language (English). We want to output a sequence of words in our target language (Korean). A essential difference is that our outflow just starts afteraward we have seen the finalyze input, because the former word of our translated phrase may require information receive from the complete input sequence.

WordDic;a)SEPabcdefghijklmnopqrstuvwxyz단어나무놀이소녀키스사랑 b)S : a symbol of input of decoding c)E : a symbol of output of decoding d)P : a empty sequence of word.

Traning data ; As word Dic proposal , the traning data also resume like a) ['word', '단어'], ['wood', '나무'], b) ['game', '놀이'], ['girl', '소녀'], c) ['kiss', '키스'], ['love', '사랑'], d) ['good', '좋아'], ['dead', '죽음'].

Learning rate is 0.01 and **Epoch time** is 100. We focus on testing the RNN with simple model, handling the tensiorFlow.\

III. Experimental Result

The graph shows that our model achieves relatively better performance with data and is more robust for overfitting than the base models

Fig.5. Grap of the Total cost

```
print('word ->', translate('word'))
print('wodr ->', translate('wodr'))
print('love ->', translate('love'))
print('loev ->', translate('loev'))
print('goal ->', translate('goal'))
print('dead ->', translate('dead'))
```

<Input>

```
=== 번역 테스트 ===
word -> 단어
wodr -> 나무
love -> 사랑
loev -> 사랑
goal -> 좋아
dead -> 죽음
```

<Result>

Fig.6. Traning Result

We training RNN with META data that result shows recall almost perfect But because of the META data set, it shows the low precise result (example of loev -> 사랑) But in the case when we train with public big data (such as, MINST), it will show the better result.

IV. Evaluation

In this section, we evaluate the performance of TensorFlow Unless otherwise stated, we run all experiments on a shared production cluster, and all figures plot median values with limit bars. In this paper we focus on system performance Sequence to Sequence Model Coding, rather than learning objectives like time to accuracy. TensorFlow is a system that allows machine learning practitioners and researchers to experiment with new techniques, and this evaluation demonstrates that the system (i) has little overhead, and (ii) can employ amounts of computation to accelerate real-world applications.

V. Conclusion

Our model will contribute to further research on the use of RNN in the language translated for regression. We have proposed a language model based on GRU-CNN-LSTM designed to treat textual data as dimensional inputs to predict the probability of the next possible word based on its previous words. We apply our approach to several networks based on LSTM. We use Python 3.6 with TensorFlow 1.7 in the same environment modeling the sequence in sequence. And the backpropagation algorithm over time (BPTT) in more detail and demonstrates which called the waste gradient problem. These prompt our lift to RNN models like LSTMs which are the current shape of the ruse for umpteen NLP labor. As we noted in the experimental result after the data traning, we tested the RNN with the META model and administered the flow of the tensor. The result shows the low and accurate result due to the META data set. And that will be the object of our future work. In future work, we will train with large public data such as MINST, which will show the best result. As well known MNIST database is a wide database of handwritten digits popularly regular for the training of diverse image treatment systems. And the database is also widely used for training and testing in the field of machine learning. Then a new data set will include 28x28 grayscale images of more high-quality fashion products. And the learning set has many images and the test set will have huge images.The MNIST mode is intended to directly replace the original MNIST data set with machine learning analysis algorithms because it shares the same image size, the same data format, training ,testing on divisional structure.

Acknowledgment

This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2018-2013-1-00877) supervised by the IITP (Institute for Information & communications Technology Promotion).

Following are results of a study on the " Leaders in Industry-University Cooperation +" Project, supported by the Ministry of Education and National Research Foundation of Korea

REFERENCES

[1] AlphaGo, https://deepmind.com/research/alphago/.
[2] Che-Lun Hung #1, Yi-Yang Lin *2, Performance of Convolution Neural Network based on Multiple GPUs with Different Data CommunicationModels,H301AR2A10.978-1-5386-5889-5/18/$31.00 ©2018 IEEE SNPD 2018,
[3] Tianyi Liu, Shuangsang Fang,Implementation of Training Convolutional Neural Networks,University of Chinese Academy of Sciences, Beijing, China
[4] LeCun, Y., Bengio, Y., & Hinton, G. 2015. Deep learning. Nature, 521(7553), 436-444
[5] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.
[6] Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks, 5(2), 157-166.
[7] J. Dean, G. S. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Y. Ng. Large scale distributed deep networks. In Proceedings of NIPS, pages 1232–1240, 2012. research.google.com/archive/large deep networks nips2012.pdf.
[8] M. I. Jordan. Serial order: A parallel distributed processing approach. ICS report8608, Institute for Cognitive Science, UCSD, La Jolla,1986.cseweb.ucsd.edu/˜gary/PAPERSUGGESTIONS/Jordan-TR-8604.pdf
[9] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial nets. In Proceedings of NIPS, pages 2672– 2680, 2014. papers.nips.cc/paper/5423-generativeadversarial-nets.pdf.
[10] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis.Human-level control through deep reinforcement learning. Nature, 518(7540):529–533, 02 2015. dx.doi.org/10.1038/nature14236 ,
[11] J. Dean, G. S. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Y. Ng. Large scale distributed deep networks. In Proceedings of NIPS, pages 1232–1240, 2012. research.google.com/archive/large deep networks nips2012.pdf.
[12] T.Brants and A. Franz. Web 1T 5-gram version 1, 2006. catalog.ldc.upenn.edu/LDC2006T13 ,
[13] http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/ ,
[14] http://www.wildml.com/2015/10/recurrent-neural-network-tutorial-part-4-implementing-a-grulstm-rnn-with-python-and-theano/
[15] MathWorks MATLAB utilize of Deep Learning.Page10 KR_Deep_Learning
[16] Martín Abadi, Paul Barham, Jianmin Chen, TensorFlow: A System for Large-ScaleMachine Learning, November 2–4, 2016 • Savannah, GA, USA ISBN 978-1-931971-33-1