Indonesian Journal of Electrical Engineering and Computer Science
Vol. 19, No. 1, July 2020, pp. 325~335
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v19.i1.pp325-335

A comparative review on deep learning models for text classification

Muhammad Zulqarnain, Rozaida Ghazali, Yana Mazwin Mohmad Hassim, Muhammad Rehan
Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia (UTHM), Malaysia

Article Info

Article history:

Received May 26, 2019
Revised Oct 23, 2019
Accepted Dec 11, 2019

Keywords:

CNN
DBN
Deep learning
RNN
Text classification

ABSTRACT

Text classification is a fundamental task in several areas of natural language processing (NLP), including words semantic classification, sentiment analysis, question answering, or dialog management. This paper investigates three basic architectures of deep learning models for the tasks of text classification: Deep Belief Neural (DBN), Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN), these three main types of deep learning architectures, are largely explored to handle various classification tasks. DBN have excellent learning capabilities to extracts highly distinguishable features and good for general purpose. CNN have supposed to be better at extracting the position of various related features while RNN is modeling in sequential of long-term dependencies. This paper work shows the systematic comparison of DBN, CNN, and RNN on text classification tasks. Finally, we show the results of deep models by research experiment. The aim of this paper to provide basic guidance about the deep learning models that which models is best for the task of text classification.

Corresponding Author:

Rozaida Ghazali,
Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia,
86400" Parit Raja, Batu Pahat, Johor, Malaysia.
Email: rozaida@uthm.edu.my

1. INTRODUCTION

Text classification (TC) is one of the important tasks of machine learning and has been extensively used in the several areas of Natural Language Processing (NLP). It's objective to designed appropriate algorithms to allow computers to extract features and classify texts automatically. Deep learning was developed from artificial neural networks and now become papolar area of machine larning, that efforts to extract high-level abstraction in data by using hierarchical mechnism. It is a developing technique and has been extensively applied in the several of areas included, pattern recognition, semantic parsing, speech recognition, computer vision and natural language processing. DNNs have become very intersting research areas in the last few years [1]. To build traditional neural networks (NNs), it is important to employ neurons to develop real-valued activations by fitting the weights. However, Backpropagation is an effective gradient descent method which has performed an essential role in ANNs since 1982. Text classification has been greatly benefited by the revival of the deep neural network (DNNs) due to their remarkable achievement with less essential of engineering features.

Deep leaning models usually take hierarchitcal architctures to combine their layers. The output of a lower layer can be considered as the input of a higher layer through simply linear or nonlinear connections. These models can process low-level word vectors features of the data into high-level abstract features vector. Based on the characteristics, deep learning techniques are more powerfull than mechine learning approaches in features representations. The performance of existing machine learning approaches commonly depend on

the users knowledge, however deep learning approaches depend on the datasets. Therefore, we identified that deep learning models have minimized the needs for users and rapidly improve the performance in the fields of computer visions.

In this paper, we have investigaed three basic and mostly useable DNNs architectures nemaly are; convolution neural network (CNN), deep belief network (DBN) and recurrent neural network (RNN) [2]. The CNNs are very complex and widely used deep architecture that perform extremely better in domains areas with big amounts of training datasets, and had untimely successes in digit classification tasks. While DBNs are a generative probabilistic architecture with composed one visible layer and few hidden layers of the deep architecture [3]. In the last five years, the RNNs have been obtained good results in various machine learning applications, and are an extension of conventional feedforward NN, which is capable to manage a variable-length sequential input. Gating mechanisms have been developed to reduce approximately gaps of essential two succeed RNNs models types such as: Long Short Term Memory introduced in 1997 (LSTM) and Gated recurrent unit (GRU) 2014 [4]. In the other words generally, we can say CNNs are especially for hierarchical, DBN general purpose and are RNN are sequential architectures.

Recently, the deep learning models have been achieved outstanding results in the various areas of NLP such as text classification [5]. Now the question is that how should we choose among them which is best for text classification tasks. Based on the previous results and characterization of these models are hierarchical model (CNN) vs. general purpose model (DBN) and vs. sequential model (RNN), the choose of CNN for challenging NLP classification tasks such a text classification. While sentiment analysis classification since this task of sentiment analysis is usually determined by key phrase; recently CNN have been showed outperformance results with gated mechanism LSTM on classification and sequence language modeling tasks. On the other hand, selection of DBN model perform the similar tasks of NLP such a text classification, it has an ability that can learns multiplex features with hidden layers and acquire more compound functions to demonstrate data. Each hidden layer unit learn a statistical connection among the units in the lower layer, the higher layer representative tends to become more complicated [6]. While RNN model chooses for the sequence to sequence sequential modeling tasks such as language modeling, and its required flexible sequential modeling of context dependencies. For example, RNN model performs very well on many sequential of tasks such as NLP, text classification, web classification, spam filtering, document-level sentiment classification and any audio datasets [7].

This study compared between CNNs, DBNs, and two very useable types of RNNs such that LSTM and GRU, systematically on classifications tasks. In this study, we found that two main finding by our research experiment: (1) CNNs and RNNs provides complementarily information for text classification tasks. Which architecture performs better depends on how important it is to semantically understand the entire sequence. However, based on our research experiment we found that some deficiencies of standard RNN are the gradient vanishing and exploding issues. It makes the training of RNN difficult, in two ways: (i) it cannot process very long sequences if using hyperbolic tanh activation function; (ii) it is very unstable if using rectified linear unit (ReLU) as an activation function. RNNs types such as LSTM and GRU manage to overcome this issue. (2) Learning rate changes performance comparatively smooth, while the batch size and hidden layers size represents large variations in results.

2. RELATED WORK

There is various deep learning models have been applied in the different areas of Natural Language Processing like a text classification and language modeling model. To the good of our knowledge, there are various systematic comparisons of these deep learning models like CNN, DBN, and RNN. According [7] to investigate hierarchical traditional CNN, general DBN and simple RNN ("i.e., no gating mechanism") relation classifications. However, Several various approaches have been developed for text classification, such as using Naïve Bayes (NB), Support Vector Machines (SVMs) with rule-based features [8], combining SVMs with naive Bayes, and building dependency trees with Conditional Random Fields [9].

The CNN extracts of the most relevant informative n-grams for the similarity and only considers their resulting activations. Neural network has multiple hidden layers to capture long-term dependencies and performed time series forecasting [10]. DBN have better extraction learning abilities and can extracted extremely recognizable features from the high-dimensional actual features area [11]. In engage the RNN to build the language models [12]. For conservation proceeds, to proposed a novel higher order RNN for temperature time series prediction [13].

In [3] the DBN jointly perform with SVM to achieved better results of Chinese text classification algorithm for labeling of semantic role presents CNN in [14]. For classification of long term sentences both are [15, 16]. One of the alternates of the traditional CNN is Network In Network (NIN) proposed by [17], where the 1*1 Conv-filter used is a Multi-Layer Perceptron (MLP) instead of the conventional linear filters

and the fully connected layers are replaced by a global average-pooling layer. In addition, the CNN joint with LSTM to achieve excellent results of an attention-based LSTM for an answer selection. In contrast, [18] comparison word2vec [19], CNN, GRU and LSTM in sentiment classification of Russian tweets, and find GRU model better classification performance then LSTM and CNN. In experimental evaluation, both [4] and [20] identified there is no clear winner among GRU and LSTM. In various multiple classification tasks, they produce similar performing and tuning hyper-parameters such as batch and layer size is frequently most important than picking the paradigm architecture.

3. DEEP MODELS DESCRIPTION

In this section describes a briefly presentation of Convolutional NN, DBN, GRUs, and LSTM.

3.1. Convolution neural network (CNN)

CNN is the extensively used in the deep learning framework and have become a very popular tool in recent years, especially in the image processing community.

a. Input layer

This layer of x consists n entries. Each entry is denoted by a d-dimensional dense vector; thus the input x is described as a feature map of dimensionality d × n. Figure 1 (a) shows the input layer as the lower rectangle with multiple columns.

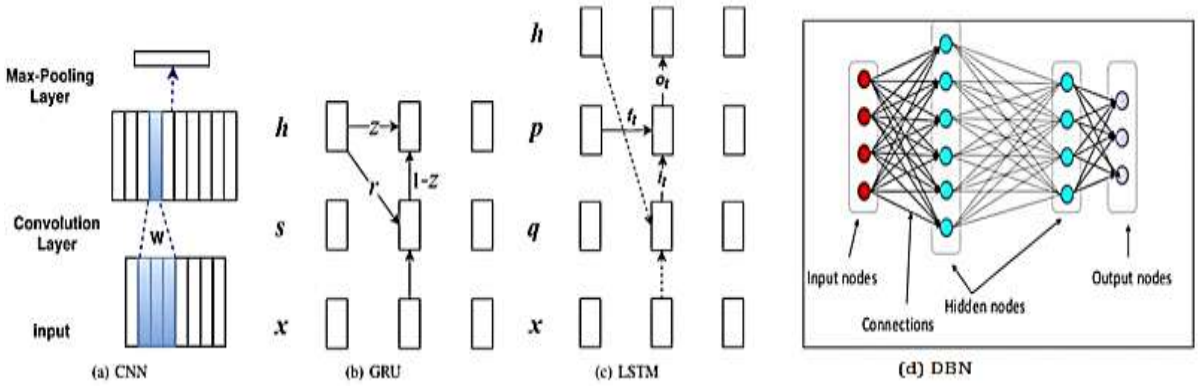

Figure 1. Four typical DNN architectures

b. Convolutional layer

This layer most important and a fundamental layer of a CNN and involved most of the computation process. Conv.layer extract set of related features maps manage neurons in it. This layer contain set of learn-able filters or kernels and these filters maps and produced two-dimensional activation when stacked composed along the depth dimension, generate the output volume. It is used to represent learn from sliding w-grams. For an input sequence with n entries: $x_1, x_2 \ldots\ldots x_n$, let vector $c_i \in R^{wd}$ be the conbined embedding of w entries $x_i - \hat{w}+1 \ldots, xi$ where w is the filter width and $0 < i < s + \hat{w}$. Embedding for x_i, $i < 1$ or $i > n$, are zero padded. We produce the illustration of model $p_i \in R^d$ for the w-gram $xi - \hat{w}+1 \ldots$. x_i used the convolution associative weights $\hat{W} \in R^{d \times wd}$.

$$p_i = tanh(\hat{W} \cdot c_i + b) \quad (1)$$

where bias $b \in R^d$.

c. Max-pooling

The main CNNs model have alternates convolutional-layers and pooling layers, the aim of these layers extract to higher level features and reduce the spatial dimension of the activation maps (without loss of information) and the number of parameters in the model minimize the computational complexity and control the overfitting issues. The pooling layers perform some of the basic computational operations are, max-pooling, average pooling, stochastic pooling [21], spatial pyramid pooling [22], spectral pooling [23], and multi-scale order less pooling [24]. However, the max-pooling layer works on data to compresses and makes

smooth the data. Max-layer selects the maximum value of the receptive field and Make data invariant to small translational changes. Figure 2 indicates the basic operation of max-pooling layer.

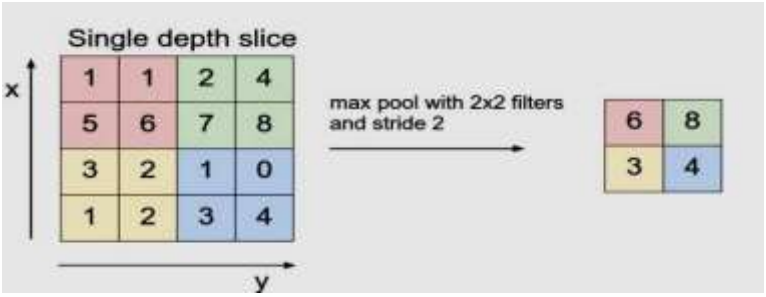

Figure 2. Max pooling process

d. Fully connected layer

It is final layer of convolutiona networks to produce the output. In this layer, all neurons are fully connected to each other in the forward and previously layer, as a systematic NN. The neurons have not especially organized (1 dimensional) so there cannot be a conv layer after a fully connected layer. In the recent, there are few architectures have been their fully connecting layer replace, as in "Network In Network" by [17], where FC replace in a global average-pooling layer.

$$f(x) = max(0, x) \quad (2)$$

3.2. Deep belief network

DBN is a deep architecture of feed-forward neural networks with one visible input-output layer and many hidden layers and also consists of several Restricted Boltzmann's Machine (RBM). The essential architecture of DBN is shown in Figure 3.

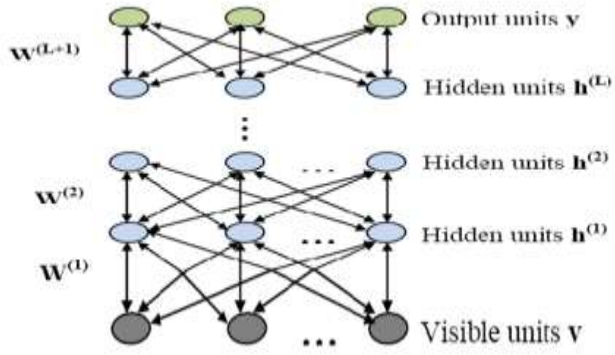

Figure 3. DBN basic architecture with L hidden layers

Let v and y show the states of visible layer nodes and hidden nodes $h_{1+2+....L,}$ respectively. For binary state nodes, that is v and y ε {0, 1}, the state of h is set to1with possibilities.

$$p_h = p(h_k = 1 \mid x) = \sigma\,(b_k + \sum_k w_{jk} x_k) \quad (3)$$

where σ(x) is the logistic sigmoid function 1/(1+exp (-y)), b_k is the bias of *h*, and *x* is the binary state. w_{jk} is the weight between *x* and *h*. In DBN these neurons of the hidden layers show a fully connected connection and a visible layers node has 0 and 1 states. When it is active, the value is 1 and node is used, and when the value is 0 the status is not activated and a node is not used.

3.3. Gated recurrent unit (GRU)

GRU was introduced by Cho et al 2014 [4] and have been used for sequence modeling. It is the latest type of recurrent neural network to adaptively capturing long-term dependencies of various time scales. GRU basic mechanism consist of two gates one is reset gate ṙ and other is update gate z, moreover, it has no separate memory and modulates information flow inner the unit. GRU Similarly to the LSTM but it easy to train as compare to LSTM. However, the GRU don't have any other alternate mechanism to handle the degree to which state is exposed. GRU has shown the following equations:

$$z_{t = sigm} (W^{(z)}x_t + U^{(z)}h_{t-1} + b_z) \quad (4)$$

$$\dot{r}_{t = sigm} (W^{(r)}x_t + U^{(r)}h_{t-1} + b_r) \quad (5)$$

$$\hat{h}_t = \tanh (Wx_t + \dot{r}_t * Uh_{t-1}) \quad (6)$$

$$h_t = z_t \circ h_{t-1} + (1- z_t) * \hat{h}_t \quad (7)$$

where, z, ṙ is the update date and reset date, $x_t \varepsilon \dot{R}^d$ represent token x at time step t, $h^t \varepsilon \dot{R}^h$ is the hidden state at time t, * is multiplication and σ, tanh is the activation functions. All $U \varepsilon \dot{R}^{d*h}$ and $W \varepsilon \dot{R}^{h*h}$ are weights parameters.

3.4. Long short term memory

The Long Short Term Memory (LSTM) unit was introduced by [25]. With the minor modification of the original unit of LSTM has been made. The RNN unit which simply computes the weighted sum of the inputs signals and applies a non-linear function. However, the LSTM manage the memory at time *t*.

$$i_t = sigm (x_tU^i + W_i h_{t-1} + b_i) \quad (8)$$

$$f_t = sigm (x_tU^f + W_f h_{t-1} + b_f) \quad (9)$$

$$o_t = sigm (x_tU^o + W_0 h_{t-1} + b_o) \quad (10)$$

$$q_t = \tanh (x_tU^q + h_{t-1} W_q + b_q) \quad (11)$$

$$p_t = f_t * p_{t-1} + i_t * q_t \quad (12)$$

$$h_t = o_t * \tanh(p_t) \quad (13)$$

LSTM has three gates: where i_t, f_t, o_t is the input gate, forget gate and output gate. Sigm is sigmoid activation function have been generated in all gates to ensembles of input x_t and the previous hidden state h_{t-1}. In order to create the hidden state at current time step *t*, it first creates a temporarily result q_t by a tanh non-linearity over the composite of input x_t and the preceding hidden state h_{t-1}, then combine this temporary result q_t with history p_{t-1} by input gate i_t and forget gate f_t accordingly to get an updated history p_t, finally the output gate o_t over this updated history p_t to become the final hidden state h_t.

4. EXPERIMENTS SETUP

4.1. Datasets

4.1.1. Sentiment text classification (SentiTC)

The dataset of Stanford Sentiment Treebank (SST) [26, 27]. The sentiments prediction of this datasets is ("positive or negative") of the movie review. In this study, we use and divide the datasets into three parts: 6911 for training, 880 for validation (*val*) and 1822 testing sentences. As in [28] have to treated label phrases that happen as a subpart of training sentences as individualistic training occasion.

4.1.2. CNAE-9

On Sem-Eval 2012 tasks 7 [29]. It contains numbers of 1080 free texts business documents that descriptive of Brazilian companies categorized into 9 subsets, it was maintained only letters and then it have removed preposition of the texts. The CNAE-9 dataset split 756 documents for training and 324 documents for testing and there is no validation set.

4.1.3. Textual entailment (TE)

On Stanford Natural Language Inference (SNLI) [30]. SNLI consist pairs of premise-hypothesis labels with a relative ("entailment, contradiction, neutral"). After that remove the unlabeled pairs, end up having 549,359 pairs for training, 09,812 for validation (*val)* and 9,857 for testing.

4.1.4. Health news in twitter (HNT)

The health news is phase on the open source of datasets taken from UCI data repository. The data was collected by twitter API, and consist of health news from more than 15 main health news agencies such as BBC, CNN, and NYT. After processing the data we use and divide into two parts 70% in training and 30% is testing.

4.1.5. 20 newsgroups (20NG)

Taken from UCI data repository in raw form and have to used [31]. This dataset is balanced and it has 20 large classes. This dataset consists of 20,000 messages and taken from 20 newsgroups. In this study, we use and divide into 14000 messages for training, 2500 sentences for validation (*val*) and 3500 sentences for testing.

4.1.6. Reuters-21578 (R-21578)

Taken from the UCI data repository and which has been used in various previously experimental research studies [32]. From Reuters-21578 dataset, 15 classes have been to used that are skewed in size. The statistics summery of all datasets are presented in Table 1. We manage data in 2 categories. (i) Text classification TextC, including SentiC, CNAE-9, and TextC1 including 20NG, and R-21578. (ii) SemMatch including TE and HNT. By evaluate these two categories, we objective to find out some fundamental techniques used in CNNs / RNNs / DBN.

Table 1. Datasets description

Datasets	No. of Instances	No. of Attributes	No. of Web Hits	Missing Values	Area	Associated Tasks
SentiTC	3000	N/A	100816	N/A	N/A	Classification
CNAE-9	1080	857	50866	N/A	Business	Classification
TE	569028	21000	63121	N/A	N/A	SemMatch
HNT	580000	25000	25174	N/A	Computer	Classification
20NG	20000	N/A	80915	No	N/A	Classification/Clustering
R-21578	21578	05	139119	N/A	N/A	Classification

4.2. Implementation setting

To objectively study the encoding ability of various traditional DNNs, we use 6 different kinds of datasets in this experiment. Data preprocessing and manipulate have performed in Python 3.6, basis on the sklearn, numpy and pandas packages. Deep learning GRU networks and traditional DNNs are executed with TensorFlow, an open source software library for numerically computations using data flows graph. In this study, the experiment consists of the following design. (1) Always train from a scratch, no extra information use e.g., no pertained word embeddings. (2) Always training use by fundamental setup without complecated tricks such as batch normalization. (3) Define the relative hyper-parameters for respectively task and each model individually. Completely simulations were implemented on Intel Core i7-3770XPU @3.40 GHz, and 4GB of RAM machine, the descriptions of all experimental parameters shown in Table 2.

Base on the optimal hyper-parameters. (4) To Investigates the fundamental architectures and explanation of every model: CNN consist of a conv-layer and max-pooling layer; LSTM and GRU model the input from left to right and consistently use the last hidden state as the final representation of the input. Hyperparameters are tuned on hidden size, mini-batch size, learning rate, maximal sentences length, and ranking loss in HNT.

Table 2. Results of DBN, CNN, GRU, and LSTM in TC

Datasets/models		Lŕ	ĥidden	ƀatch	SentLen
SentiC (acc)	DBN	0.2	30	64	60
	CNN	0.2	30	32	60
	GRU	0.1	20	64	60
TextC	LSTM	0.2	20	64	60
	DBN	0.10	75	40	20
CNAE-9 (F1)	CNN	0.12	70	32	20
	GRU	0.12	80	128	20
	LSTM	0.1	80	32	20
TE (acc)	DBN	0.1	70	64	50
	CNN	0.1	70	50	50
	GRU	0.1	50	80	65
SemMatch	LSTM	0.1	80	32	50
HNT (MAP&MRR)	DBN	0.01	30	50	40
	CNN	0.01	30	60	40
	GRU	0.1	80	128	40
	LSTM	0.1	60	128	45
20NG (acc)	DBN	0.01	110	50	60
	CNN	0.01	100	32	60
	GRU	0.1	90	64	60
TextC1	LSTM	0.1	90	64	60
R-21578 (acc)	DBN	0.01	80	70	60
	CNN	0.01	80	50	60
	GRU	0.1	100	64	60
	LSTM	0.1	100	64	60

5. RESULTS AND ANALYSIS

In this section, we conducted the reseaech experimental for the tasks of text classification on several datasets with corresponding hyper-parameters. We evaluate the performance of all state-of-the-art deep learning approaches on the terms of accuracy (Acc) and Mean Reciprocal Rank (MRR). According to implementation setting and experiment basis, in text classification, every model has performed well on all datasets but GRU model show excellent performance on SentiC are evaluated with baseline deep learning models is such as DBN, CNN and LSTM as shown in Figure 4. In textC1, both GRU and LSTM are outperforming DBN and CNN. GRU show good result on 20NG datasets and LSTM show better result on R-21578 datasets.

In this study, we did conclude in experiment: the types of RNNs models such that GRU, LSTM are best and suitable for long-range context dependencies and text classification tasks. But in one other category, sentiment match, some unforeseen observations had shown. CNNs and DBN both are considered better at extract local and position-invariable features and have shown good performance on SentiMatch (dataset HNT), but in our experiments RNNs has shown excellent performance in contrast to CNN and DBN, especially in 20NG and SentiC, because RNNs which predicts and finally produce the relation output after processing the whole sentence.

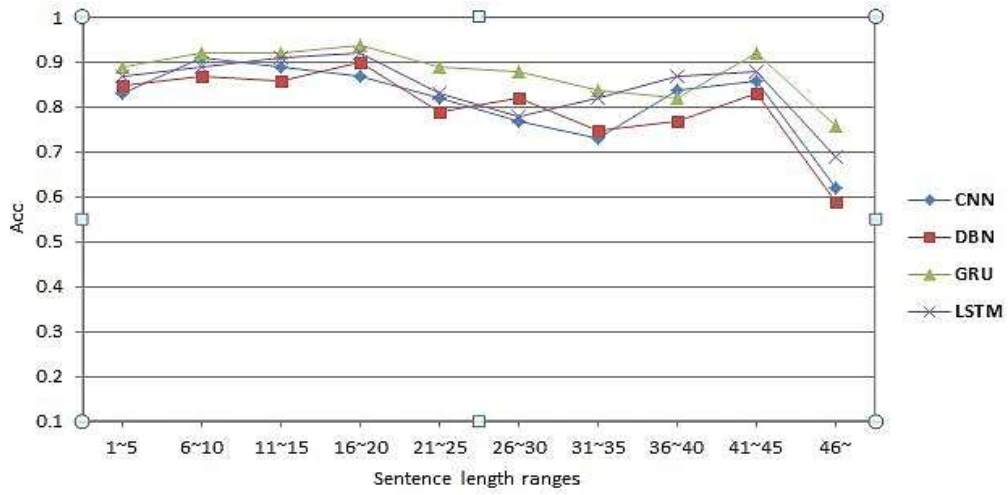

Figure 4. Distributions of various sentence lengths ranges and accuracy

5.1. Qualitative analysis

In this study, we show experiment that in which CNN prediction base is acceptably, although the GRU predicts falsely or vice versa. In the experiment, we have found out and show that GRU excellent performs on long-term range sentences dependencies. Studying accuracy vs sentence length can also support this. Figure 4 represent the accuracy w.r.t range of lengh. We observed that GRU and CNN are similar when the sentence length range are small, e.g., <12, then GRU increase the advantage over CNN when meet longer length of sentence. Error analysis presenting that longer sentence frequently contains of semantically paragraphs of inverse. This type of paragraph often includes a local robust measure for one sentiment polarity, but the effective classification relies on the understanding of the whole article. Consequently, which deep learning model performs better in text classification task depends on how often the conception of global/long-range semantics is required. This can also describe the occurrence in SemMatch – GRU/LSTM exceed CNN in TE while CNN predominates in HNT as text entailment relies on the comprehensive of the entire sentence [30].

5.2. Sensitivity to hyperparameters

In the next step we check the performance of all deep learning models such that, CNN, DBN, GRU, and LSTM, how it performs stable performance when hyper-parameters values are different. Figure 5 present the performance of CNN, DBN, GRU, and LSTM on the term of various learning rates, hidden layers and batch size. All DNN models show comparatively smooth with respects to learning rate changes. In contrast, hidden size and batch size reason of large oscillation. However, we still observed about CNN curves are mainly below the lines of DBN, GRU and LSTM in SentiTC, and TextC but outperformance on HNT dataset in sentiment match task.

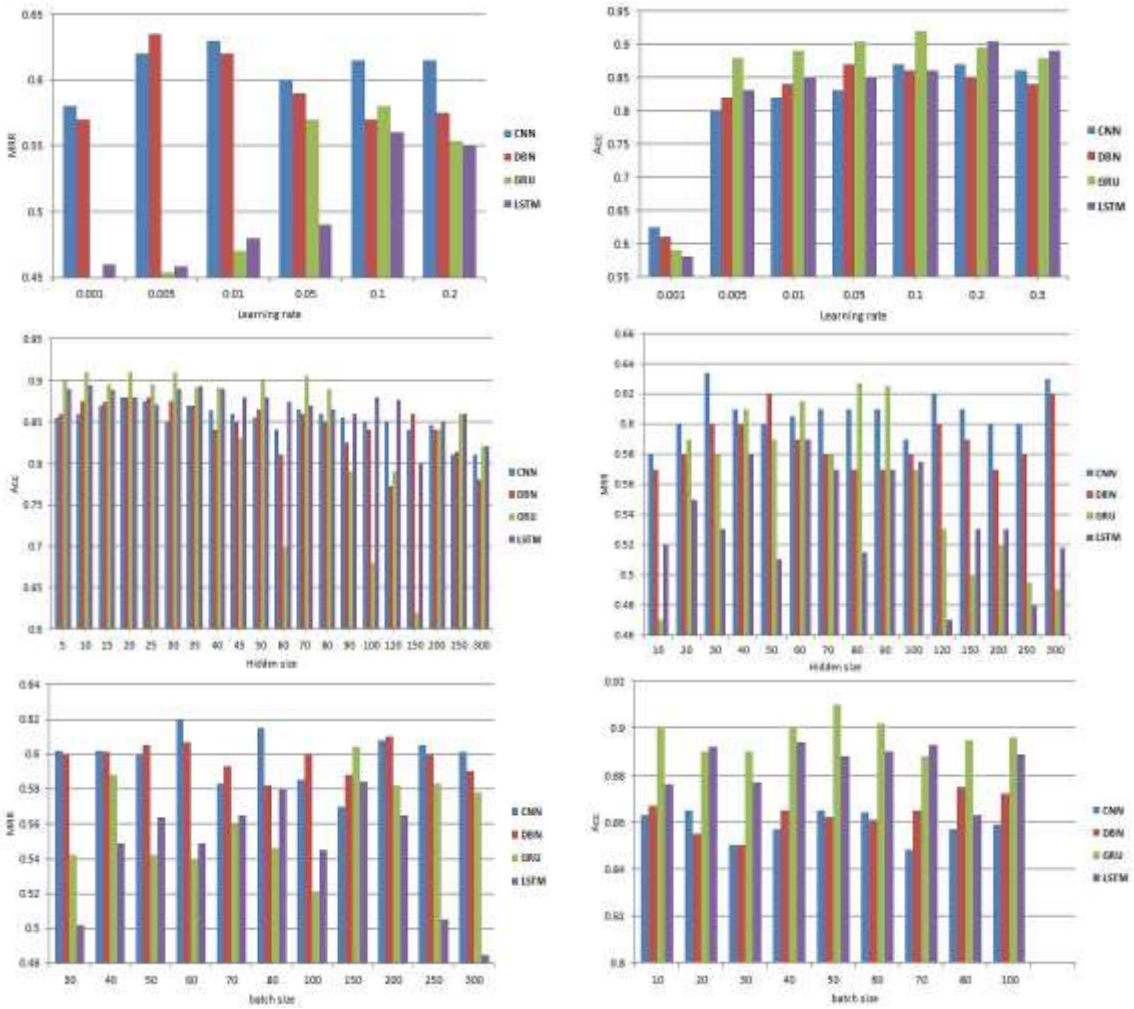

Figure 5. Accuracy for sentiment classification (left) and MRR (Mean Reciprocal Rank) for HNT (right) as a function of three hyperparameters: learning rate (top), hidden size (center), and batch size (bottom)

6. CONCLUSION

In this paper, we have comparative reviewed of existing deep learning models. We concluded deep learning models are practically for us to solve many issues. This study invertigated and compared the four most extensively used deep neural networks namely the DBN, CNN, GRU, and LSTM for text classification. In this study, we found that the types of RNNs – GRU and LSTM networks perform well in sequential learning tasks and overcome the problems of vanishing and explosion of gradients in traditional RNNs when learning long-term dependencies. In addition, hidden size and batch size can construct DNN models performances vary dramatically. This suggestion that optimization of these two parameters is critical to the better performance of three models DBN, CNNs and RNNs. With the rapid development of hardware resources and computation technologies, we are hopeful that deep neural networks will obtain higher attention and find more extensive applications in the future.

ACKNOWLEDGMENTS

The authors would like to thanks Ministry of Education Malaysia, Universiti Tun Hussein Onn Malaysia and Research Management Center (RMC) for funding this research activity under the Fundumental Research Grant Scheme (FRGS), vote no.1641.

REFERENCES

[1] C. Science *et al*. "A Novel Approach for Efficient Training of Deep Neural Networks," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 11, no. 3, pp. 954–961, 2018.

[2] B. Shickel, P. J. Tighe, and A. Bihorac, "Deep EHR : A Survey of Recent Advances in Deep Learning Techniques for Electronic Health Record (EHR) Analysis," *IEEE J. Biomed. Heal. Informatics*, vol. 22, no. 5, pp. 1589–1604, 2018.

[3] G. Tzortzis and A. Likas, "Deep Belief Networks for Spam Filtering," *19th IEEE Int. Conf. Tools with Artif. Intell. 2007)*, pp. 306–309, 2007.

[4] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," pp. 1–9, 2014.

[5] L. Tan *et al.*, "SDF-NN: A Deep Neural Network with Semantic Dropping and Fusion for Natural Language Inference," *2017 IEEE 29th Int. Conf. Tools with Artif. Intell.*, pp. 72–79, 2017.

[6] L. Y. Ann, P. Ehkan, M. Y. Mashor, S. M. Sharun, and L. Y. Ann, "FPGA-based architecture of hybrid multilayered perceptron neural network," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 14, no. 2, pp. 949–956, 2019.

[7] N. T. Vu, H. Adel, P. Gupta, and H. Schütze, "Combining Recurrent and Convolutional Neural Networks for Relation Classification," *arXiv Prepr. arXiv1605.07333*, pp. 412–418, 2016.

[8] J. Silva, L. Coheur, A. C. Mendes, and A. Wichert, "From symbolic to sub-symbolic information in question classification," *Artif. Intell. Rev.*, vol. 35, no. 2, pp. 137–154, 2011.

[9] T. Nakagawa, K. Inui, and S. Kurohashi, "Dependency tree-based sentiment classification using CRFs with hidden variables," *Proceeding HLT '10 Hum. Lang. Technol. 2010 Annu. Conf. North Am. Chapter Assoc. Comput. Linguist.*, no. June, pp. 786–794, 2015.

[10] R. Ghazali, Z. A. Bakar, Y. Mazwin, and M. Hassim, "Functional Link Neural Network with Modified Cuckoo Search Training Algorithm," *Int. Conf. Intell. Comput.*, pp. 285–291, 2014.

[11] J. Song, S. Qin, and P. Zhang, "Chinese Text Categorization Based on Deep Belief Networks," *IEEE ICIS 2016*, no. June, pp. 1–5, 2016.

[12] P. Vincent, "A Neural Probabilistic Language Model," *A neural probabilistic Lang. Model. J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, 2003.

[13] R. Ghazali, N. A. Husaini, L. H. Ismail, T. Herawan, and Y. M. M. Hassim, "The performance of a Recurrent HONN for temperature time series prediction," *Proc. Int. Jt. Conf. Neural Networks*, no. July, pp. 518–524, 2014.

[14] R. Collobert and J. Weston, "A unified architecture for natural language processing," *Proc. 25th Int. Conf. Mach. Learn. - ICML '08*, pp. 160–167, 2008.

[15] M. Zulqarnain, R. Ghazali, M. G. Ghouse, and M. F. Mushtaq, "Efficient Processing of GRU Based on Word Embedding for Text Classification," *Int. J. Informatics Vis.*, vol. 3, no. 4, pp. 377–383, 2019.

[16] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent Convolutional Neural Networks for Text Classification," *Twenty-Ninth AAAI Conf. Artif. Intell.*, pp. 2267–2273, 2018.

[17] N. Aloysius and M. Geetha, "A review on deep convolutional neural networks," *2017 Int. Conf. Commun. Signal Process.*, pp. 0588–0592, 2017.

[18] A. K. arkhipenko, K. I. kozlov-ilya, T. J. integral, S. K. kirillskorniakov, G. A. gomzin, and T. D. turdakov, "Comparison of neural network arChiteCtures for sentiment analysis of russian tweets," *Dialogue Can. Philos. Assoc.*, 2016.

[19] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "5021-Distributed-Representations-of-Words-and-Phrases-and-Their-Compositionality," *Adv. Neural Inf. Process. Syst.*, vol. April, pp. 3111–3119, 2013.

[20] A. Dosovitskiy, J. T. Springenberg, and T. Brox, "An Empirical Exploration of Recurrent Network Architectures Rafal," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June, pp. 1538–1546, 2015.

[21] N. A. Muhammad, A. A. Nasir, Z. Ibrahim, and N. Sabri, "Evaluation of CNN , Alexnet and GoogleNet for Fruit Recognition," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 12, no. 2, pp. 468–475, 2018.
[22] A. Nguyen, J. Yosinski, and J. Clune, "Deep Neural Networks are Easily Fooled: High Confidence Predictionsfor Unrecognizable Images," *Comput. Vis. Pattern Recognit.*, 2015.
[23] O. Rippel, J. Snoek, and R. P. Adams, "Spectral Representations for Convolutional Neural Networks," pp. 1–9, 2015.
[24] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8695 LNCS, no. PART 7, pp. 392–407, 2014.
[25] S. Hochreiter, "Long Short Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1–32, 1997.
[26] R. Socher, A. Perelygin, and J. Wu, "Recursive deep models for semantic compositionality over a sentiment treebank," *Proc.* no. October, pp. 1631–1642, 2013.
[27] M. Zulqarnain, R. Ghazali, S. H. Khaleefah, and A. Rehan, "An Improved the Performance of GRU Model based on Batch Normalization for Sentence Classification," *Int. J. Comput. Sci. Netw. Secur.*, vol. 19, no. 9, pp. 176–186, 2019.
[28] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A Convolutional Neural Network for Modelling Sentences," *Proc. 52nd Annu. Meet. Assoc. Comput. Linguist. (Volume 1 Long Pap.*, pp. 655–665, 2014.
[29] I. Hendrickx *et al.*, "SemEval-2010 Task 8 : Multi-Way Classification of Semantic Relations Between Pairs of Nominals," *Comput. Linguist.*, no. June 2009, pp. 94–99, 2010.
[30] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," 2015.
[31] W. Sharif, N. A. Samsudin, M. M. Deris, and M. Aamir, "Improved relative discriminative criterion feature ranking technique for text classification," *Int. J. Artif. Intell.*, vol. 15, no. 2, pp. 61–78, 2017.
[32] J. Nam, J. Kim, E. Loza Mencía, I. Gurevych, and J. Fürnkranz, "Large-scale multi-label text classification - Revisiting neural networks," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8725 LNAI, no. PART 2, pp. 437–452, 2014.

BIOGRAPHIES OF AUTHORS

Muhammad Zulqarnain received his Bachelor and Master degree in Computer Science & Information Technology from The Islamia University of Bahawalpur (IUB), Pakistan. He received his M.Phil degree (Master of Philosophy) from National College of Business Administration & Economics, Lahore, Pakistan. He is currently pursuing Ph.D from University Tun Hussein Onn Malaysia. His research interest is Machine Learning and Deep learning for natural language processing and its application

Rozaida Ghazali is currently a Professor at the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia (UTHM). She graduated with a Ph.D. degree in Higher Order Neural Networks from the School of Computing and Mathematical Sciences at Liverpool John Moores University, United Kingdom in 2007. Earlier, in 2003 she completed her M.Sc. degree in Computer Science from Universiti Teknologi Malaysia (UTM). She received her B.Sc. (Hons) degree in Computer Science from Universiti Sains Malaysia (USM) in 1997. In 2001, Rozaida joined the academic staff in UTHM. Her research area includes neural networks, swarm intelligence, optimization, data mining, and time series prediction. She has successfully supervised a number of PhD and master students and published more than 100 articles in various international journals and conference proceedings. She acts as a reviewer for various journals and conferences, and as an editor in a few Springer conference proceedings. She has also served as a conference chair, and as a technical committee for numerous international conferences.

Yana Mazwin Mohmad Hassim is a senior lecturer at the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia (UTHM). She graduated with a PhD degree from Universiti Tun Hussein Onn Malaysia (UTHM) in 2016. Earlier, in 2006 she completed her Master's degree in Computer Science from Universiti of Malaya (UM). She received her Bachelor of Information Technology (Hons) degree majoring in Industrial Computing from Universiti Kebangsaan Malaysia (UKM) in 2001. In 2003, Yana Mazwin joined the academic staff in UTHM. Her research area includes neural networks, swarm intelligence, optimization and classification.

Muhammad Rehan is an Assistant professor at the Department of Computer Science and Information Technology, Baqai Medical University (BMU) Karachi, Pakistan. He received his M.Phil. Degree (Master of Philosophy) from the Faculty of Computer Science and Information Technology, Hamdard University core area is Computer Networks in 2016. Karachi, Pakistan. He received his Bachelor of Computer Science degree majoring in Software Engineering from Dadabhoy Institute of Higher Education (DIHE) Karachi, Pakistan in 2004. He is currently pursuing his Ph.D. from University Tun Hussein Onn Malaysia (UTHM). His research area is Data Mining and Machine Learning for Optimization and its application.