Analysis on the Effect of Dropout as a Regularization Technique in Deep Averaging Network

S. Lovelyn Rose[1], M. Rashmi[2]
{ slr.cse@psgtech.ac.in[1], rashmi.manavalan@gmail.com[2]}

Professor (Department of CSE), PSG College of Technology, Coimbatore, India[1], Student (Department of CSE), PSG College of Technology, Coimbatore, India[2]

Abstract. Deep neural networks are powerful machine learning systems and many deep learning models for natural language processing tasks focus on learning the compositionality of their inputs. DAN model relies on both simple vector operations and neural network-based models for learning the compositionality. The depth of the model allows it to capture subtle variations in the input even though the composition is unordered. However, overfitting is a serious problem in any deep neural network. Dropout is a technique for addressing overfitting in large neural networks. The idea is to randomly drop neurons and their connections from the neural network during training phase. This prevents neurons from co-adapting. DAN includes a variant of dropout where individual words are dropped rather than individual neurons of the feed forward network.But since this technique has the potential to drop critical words it may have significant impact on the performance of the model in text classification tasks. This paper deliberates on this drawback and the impact of dropping individual neurons rather than word-level dropout.

Keywords: DAN; Dropout; Word Embeddings; Sentiment Analysis.

1 Introduction

Word embeddings are numerical representation of input text in vector space using a composition function. A composition function is used to compose word embeddings into vectors that effectively capture the meaning of the underlying text. There are two types of composition functions: Unordered and Syntactic. Unordered functions take input as Bag of Words (count based) representation while syntactic functions take word order and sentence structure into account. Syntactic functions are known to outperform unordered functions in several tasks [1]. Syntactic functions require much more training time and is computationally more expensive when it comes to larger datasets. Unordered composition functions are computationally faster in training while syntactic functions are more accurate.

One example of unordered composition functions is the neural bag of words.The neural bag of words is a fully connected network which maps an input text into one of ‘k’ output labels[3].The hidden layer representation of the input sequence of text is obtained as an average of the input vectors which is then passed to a fully connected softmax layer to estimate probabilities for the output labels .

ICCAP 2021, December 07-08, Chennai, India
Copyright © 2021 EAI
DOI 10.4108/eai.7-12-2021.2314492

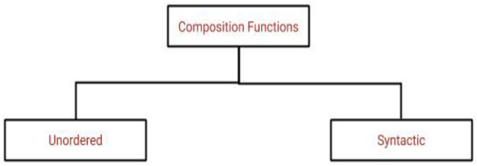

Fig 1: Classification of Composition functions

An example of syntactic composition functions would be Recursive Neural Networks (RecNN).It is a variant of neural network obtained by using weights recursively by traversing the input structure in topological order.

2 Deep Averaging Network

Deep Averaging Network (DAN) is a variant of neural bag of words that add in multiple layers of non-linearity. It tries to leverage the speed of unordered composition functions with the accuracy of syntactic functions. DAN constructs multiple neural layers upon the average of the word vectors. The working of DAN can be summarized as follows: averaging the embeddings associated with input sequence, passing the average through one or more linear/non-linear layers and a linear classification in the final layer.

A novel-variant of dropout regularizer is applied where each training instance randomly drops some of the tokens before the average operation is performed. Deep Averaging Networks can also be applied to data with high level of syntactic variance. The model works by the principle that more discriminative features can be extracted by deepening the layers of the neural network. The Deep Averaging Networks can perform nearly as well as complex composition functions-based models. This model is comparatively fast because most of the operations are element wise or matrix multiplications.

The main idea behind a deep feed forward neural network is that each layer learns a more abstract form of input than the previous layers.The first step is to compute the corresponding vector representation of the input text by taking an average of the individual word vectors as shown in equation (1) where X represents the input sequence of text, g represents the composition function and v_w is the word embedding associated with each word w ∈ X.

$$z = g(w \in \mathrm{X}) = \frac{1}{|X|}\sum_{w \in \mathrm{X}} v_w \; (1)$$

Unlike NBOW where this vector z is passed to a logistic regression function, this vector is further transformed through several feed forward layers, the output at each layer can be computed as follows where W represents the weight matrix and b represents the bias associated with each layer.

$$z_i = g(z_{i-1})$$
$$`z_i = f(W_i . z_{i-1} + b_i)$$

The output of the final feed forward layer is sent to a softmax layer. The training time of DAN is similar to that of NBOW and the addition of several feed forward layers enables the model to detect slight variations in input compared to NBOW model. It is comparatively less complex to ReccNN and the complexity of DAN increases only with the number of layers and not the number of nodes in a parse tree in ReccNN[3].

A variation of DAN, termed the ADAN (Adversarial Deep Averaging Network)[10] was proposed to tackle text classification problems in languages without adequate annotated data. The basic proposal was to transfer the learnings from a resource rich language to a low-resource language. ADAN encompasses both a sentiment classifier and a language discriminator both of which takes the input from a DAN (Deep Averaging Network) which acts as a feature extractor.

3 Dropout

Dropout is a stochastic regularization technique for reducing overfitting and thereby improving the generalization of the model. It is also argued that it reduces neuron co-adaptation and improves the sparseness in feature representation. Deep neural networks contain multiple hidden layers that areable to learn complicated relationships between the input and the output.

The key idea is to randomly drop units (along with their connections) from the neural network during training. Application of dropout to a network produces a “thinned” network from it. The thinned network is basically composed of nodes that survived the dropout[2]. A neural network composed of ‘n’ nodes can have a maximum of 2^n thinned networks. During each training phase, a new thinned network is produced and trained.Thereforetraining a neural network using dropout as a regularizer equals training 2^n thinned networks that share weights.Since the networks share weight the total number of parameters required is still in the order of n^2 or lesser [2]. During testing phase, it is infeasible to explicitly to perform average on the predictions from many thinned networks.At test time, it is easy to approximate the effect of averaging the predictions of all these thinned networks by simply using a single unthinned network that has smaller weights.[2]This reduces overfitting and makes the model more generalized.

Dropout prevents co-adaptation by making the presence of other hidden units unreliable. Therefore, a hidden unit cannot rely on other specific units to correct its mistakes. It must perform well in a wide variety of different contexts provided by the other hidden units [1].

For a neural network with ‘N’ hidden layers, let i ∈ {1, . . , N} denote the position of the hidden layers in the network and $W^{(i)}$ and $b^{(i)}$ represent the weights and bias at each of these layers ,$z^{(i)}$ represents the input and $y^{(i)}$ represent the output at each layer. The output from each layer is multiplied by a vector of Bernoulli random variable ($r^{(i)}$)each with a probability p of being 1,to generate thinned output $x^{(i)}$. This thinned output is passed as the input to the next layer.

$\rho_{\varphi}{}^{(\iota)} \sim \Beta\varepsilon\rho\nu o\upsilon\lambda\lambda\iota(\pi)$ (3)

$\xi^{(\iota)}= \rho^{(\iota)}*\psi^{(\iota)}$ (4)

The feed forward layer of a simple neural network can be given as below:

$\zeta\varphi(\iota+1) = \omega(\iota)*\psi(\iota)+ \beta\varphi(\iota+1)$ (5)

$\psi\varphi(\iota+1) =\phi(\zeta\varphi(\iota+1))$ (6)

Here, f denotes the activation function. After applying dropout the feed forward operation now becomes,

$$\zeta\varphi(\iota+1) = \omega(\iota)*\xi(\iota)+\beta\varphi(\iota+1) \quad (7)$$

$$\psi\varphi(\iota+1) = \phi(\zeta\varphi(\iota+1)) \quad (8)$$

These operations are applied at each hidden layer which can also be summarized as sampling a larger network into smaller sub-networks and the derivatives of the loss function are back propagated through these sub networks. Bernoulli dropout is one type of a regularization technique that approximates the variational distribution for uncertainty estimates. The dropout applied here is only during the training phase. The dropout as a Bayesian approximation[9] proposes an even more quantifiable approach to model uncertainty. The idea is to apply dropout during training and testing as approximate Bayesian interference in deep Gaussian process. This reduces the problem of representing a models uncertainty without compromising the accuracy and the computational cost.

4 Proposed Architecture

DAN's use a variant of dropout that removes entire words from the input[1]. This has the potential to remove important tokens (e.g. "bad" in "This experience was bad"). Though the probability of removing important tokens might be relatively low, the impact these words have on the final classification will be relatively high.

Instead of dropping out individual words, the effect of dropping out individual units in the feedforward layer is analyzed. The effect ofword dropout rate at a moderate level improves model performance. Better results for DAN in sentiment classification tasks was obtained with a dropout rate of 0.3. Hence individual units in the feed forward layer were dropped out with a probability of 30%.

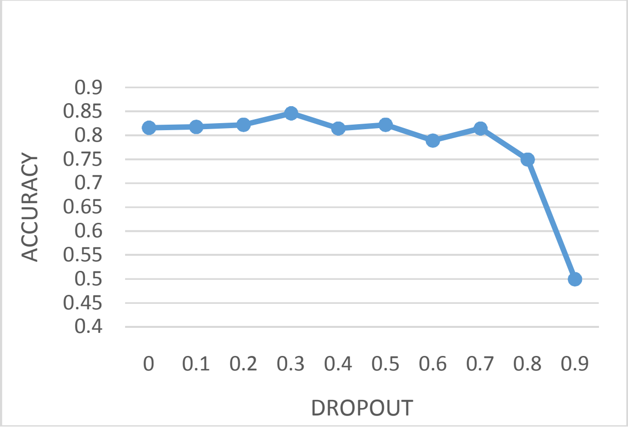

Fig 2: Effect of dropout in model accuracy

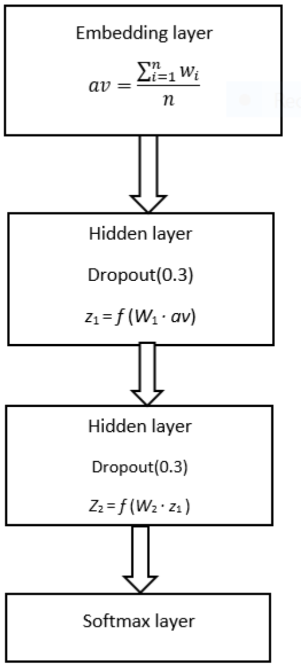

Fig 3: DAN with dropout at feed forward layers

The effective dropout rate of 0.3 is applied to the feed forward linear layers at each iteration .The softmax layernormalizes its input as a probability distribution of 'k' probabilities where k is the number of classes.

A modified architecture with one feed forward layer is also designed to analyze the impact of deepness of a network on the model's performance.

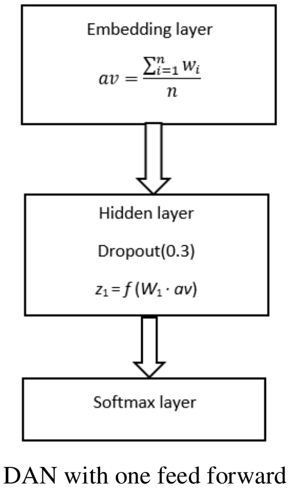

Fig 4: DAN with one feed forward layer

5 Experimental Setup

Sentiment analysis is the process of determining the emotion/sentiment over a series of words to aid in better understanding of the polarity of the context. Traditionally, one hot encoding is used to represent words which results in sparse vectors and also the semantics of the words aren't considered resulting in different representation of similar words.

The dataset used for evaluation is the 'Sentiment140', created by Stanford University, which comprises tweets collected from various sources and annotated with a polarity. The dataset comprises of the following fields, polarity of the tweet, tweet id, tweet date, query, username and the text.

The original DAN model used a variant of drop out, by dropping out units before the averaging layer[1]. The experiment aims to determine the best architecture by applying dropout not only at dimension-level as in the original architecture but also at word-level and its impact on the overall accuracy is analyzed.

It has been shown that pretrained word embeddings when input to DAN performed much better than randomly initialized embeddings [1].Sothe first task was to generate the corresponding word embeddings for the given dataset. Popular word embedding techniques like word2vec and GloVe were used to generate embeddings. The generated embeddings from both these techniques were compared using Logistic Regression and ANN as classifiers in a Sentiment analysis task. The following accuracies were obtained.

Table 1: Comparison of accuracy of word embedding models

Classifiers **Embeddings**	**Logistic Regression**	**ANN**
word2vec	74.78	80.56
GloVe	74.36	79.82

The word embeddings obtained from word2vec were slightly better than Glove embeddings and the word2vec embeddings were given as input to DAN for analysis. The original DAN architecture has showed that pretrained word embeddings performed better with DAN when compared to randomly initialized embeddings. Hence pretrained word embeddings from word2vec is passed as input to the DAN architecture.The experiments were conducted in a 64-bit, dual core laptop and the results of text classification are as follows/

Accuracy is the ratio of correctly predicted observations and the total number of observations. From Table 2, it can be seen that DAN with dropout in the hidden layer performed better than the original model for the given sentiment classification task.

Table 2: Comparison of accuracy of DAN variant models

MODEL	**ACCURACY**
DAN	80.5
DAN with Dropout	82.3
DAN with Dropout (Single feed forward layer)	81.7

F1 score is the harmonic mean of recall and precision. Precision can be defined as the measure of how correct the predicted values truly are while recall can be defined as the measure for how relevant the truly returned results are. As seen in Table 3, DAN with Dropout at the hidden layer has a slightly higher score compared to the original DAN model.

Table 3: Comparison of F1 score of DAN variant models

MODEL	F1 score
DAN	0.827
DAN with Dropout	0.832
DAN with Dropout (Single feed forward layer)	0.823

The AUC-ROC metric signifies the capability of the model to distinguish between classes.The ROC curve is plotted with TPR(True Positive Rate) in the Y-axis and the FPR(False Positive Rate) on the X axis. Higher the AUC metric,better the model can distinguish between classes.

Table 4: Comparison of ROC score of DAN variant models

MODEL	AUC
DAN	0.827
DAN with Dropout	0.832
DAN with Dropout (Single feed forward layer)	0.823

Average precision computes the precision value with recall over a range of 0 to 1. This metric is predominantly used in binary classification and closer the value is to 1, better the model. The proposed model has a slightly higher score than DAN and justifies the improvement in classification by adding dropout at the feed forward layers.

Table 5: Comparison of average precision score of DAN variant models

MODEL	AVERAGE PRECISION
DAN	0.827
DAN with Dropout	0.832
DAN with Dropout (Single feed forward layer)	0.823

The training time each model took for 30 epochs is as given in Fig 6. The training time is also significantly reduced in DAN by adding dropout to the hidden layers.

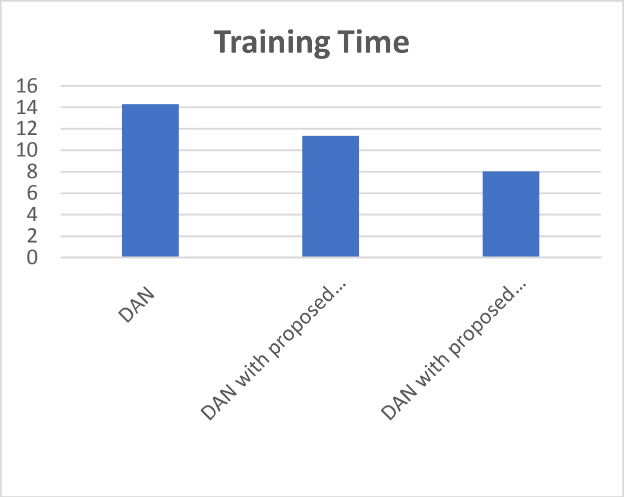

Fig 5: Comparison of training time

The effect of adding dropout to the feed forward layers has shown enhanced performance to the original DAN model using the given dataset

6 Conclusion

Though the architecture itself is trivial, Deep Averaging Networks challenge many of the conceptions in text classification. They are also a powerful and quick baseline for many text classification tasks. Deep Averaging Networks perform very well with low computational cost. An attempt to improve the DAN model by increasing the computation. By dropping out dimensions rather than individual words the performance of the model has been enhanced slightly. Since dropout reduces co-adaptation of neural units, better results have been obtained when compared to the original DAN model. Also, by increasing the depth computational complexity also increases, but with depth, variations in input are captured effectively by the network and thus are able to give results comparable to that of syntactic composition functions. Since the ordering and morphological features of the individual words isn't considered during composition, the performance of the model in texts with negation is not as expected.

One future prospect would be to selectively dropout words that are less discriminative to the experimental outcome. This can be achieved by using attention mechanisms that yields better sentence representations. One such attention mechanism is theCMA(Cascading Multiway Attention) which uses multiway attention mechanisms to generate attention to sentences to generate efficient representations for differentiating the polarity of different document-level or sentence-level representations [8].

References

[1] Mohit Iyyer,Varun Manjunatha,Jordan Boyd-Graber,HalDaumé" Deep Unordered Composition Rivals Syntactic Methods for Text Classification", Proceedings of the 53rd Annual Meeting of the

Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, July, 2016

[2] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever and Ruslan Salakhutdinov," Dropout: A Simple Way to Prevent Neural Networks from Overfitting",Journal of Machine Learning Research ,2014.

[3] I. Sheikh, I. Illina, D. Fohr, and G. Linarès, "Learning Word Importance with the Neural Bag-of-Words Model," Proceedings of the 1st Workshop on Representation Learning for NLP, 2016.

[4] Minh-Thang Luong ,Richard Socher, Christopher D. Manning," Better Word Representations with Recursive Neural Networks for Morphology", Proceedings of the Seventeenth Conference on Computational Natural Language Learning, pages 104–113,2013.

[5] Mikolov T., Sutskever S., Chen K.,Corrado G., and DeanJ., "Distributed representations of words and phrases and their compositionality", NIPS, pages 3111–3119,2013.

[6] Jeffrey Pennington , Richard Socher , Christoper Manning ,"Glove: Global Vectors for Word Representation", EMNLP,2014.

[7] Quoc Le, Tomas Mikolov,"Distributed representations of sentences and documents",International conference on machine learning, 1188-1196,2014.

[8] Dehong Ma, Sujian Li, Xiaodong Zhang, Houfeng Wang, Xu Sun, "Cascading Multiway Attentions for Document-level Sentiment Classification ",Proceedings of the the 8th International Joint Conference on Natural Language Processing, pages 634–643,2017.

[9] Yarin Gal, Zoubin Ghahramani," Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning",International Conference on Machine Learning, 2016.

[10] Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, Kilian Weinberger ,"Adversarial Deep Averaging Networks for Cross-Lingual Sentiment Classification",Transactions of the Association for Computational Linguistics, vol. 6, pp. 557–570, 2018.