International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-5, June 2019

Clustered Capsule Network Architecture for Text Classification

Madhuram M, Mayukh Dasgupta, Aqib Muhammed Ashik BT, Surya M

Abstract : In this paper we show that capsule network with some changes in its architecture and with the help of dynamic routing can mimic the speech processing section of the brain to some extent. The results obtained are state of the art and it also challenges some aspects of the capsule network architecture proposed by [Hinton et al., 2017]. This paper also makes a few changes in the selection procedure of the N-gram model proposed by [Wei Zhao et al., 2018]. The paper proposes the idea of mimicking the brain architecture for speech recognition using capsule network by clustering the final capsules into groups of similar lengths of vectors which may represent a specific section of the brain to understand properties of a text. As a result the instantiation properties of text are not lost.

Keywords :Capsule Network, Dynamic Routing, Machine Learning, Natural Language Processing, N-grams Model

I. INTRODUCTION

The Human Brain is specifically divided into two hemispheres (left and right). The left hemispheres consist of a perisylvian area which in turn consists of Broca's area and Wernicke's area. These parts of the brain help in speech recognition. Ocular signal is transmitted across the neurons in these areas. The transmission of ocular signals from one neuron to another is no act of probability, but a pre-decided path of hyperpolarized ionic system. Assume, a neuron A is connected simultaneously with neurons B and C. An ocular signal is supposed to be transmitted to B from A, then the path from A to B is hyperpolarized as per the output of B.

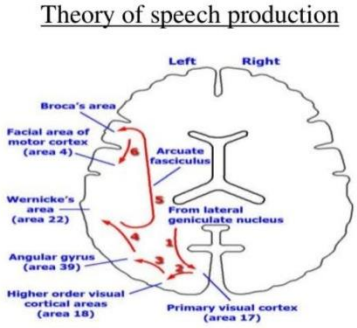

Figure 1: Theory of Speech Production

Manuscript published on 30 June 2019.
***** Correspondence Author (s)
Madhuram M*, Department of Computer Science, SRM Institute of Science and Technology, Chennai, India.
Mayukh Dasgupta, Department of Computer Science, SRM Institute of Science and Technology, Chennai, India.
Aqib Muhammed Ashik BT, Department of Computer Science, SRM Institute of Science and Technology, Chennai, India.
Surya M, Department of Computer Science, SRM Institute of Science and Technology, Chennai, India

Text classification computationally is a fundamental topic in natural language processing. To model text one needs to know the various compositions, hierarchies of language and their words. A very important aspect of natural language processing(NLP) is to understand the context in which a word is being spoken. Earlier efforts to text classification have been done and have reached SOTAs. Methods such as using a simple bag of words [Joachims, 1998; McCallum et al., 1998] classifier, implying understanding the meaning of each word or N-gram is a necessary step towards more complex models. [Mikolov et al., 2013] stated that more than understanding the meaning of a word it is necessary to understand "what" the word is trying to say. In simple words, there can be different interpretations of a word in different contexts. Keeping this in mind the texts were treated as a sequence and focused on their spatial patterns. These sequences could then be recognised by CNNs (Convolutional Neural Networks) and LSTMs (Long Short Term Memory Networks). These two helped reach text classification state of the art. But there was a problem with the networks. These networks produced scalar outputs and thus much of the spatial properties of the words were lost. It was quite a concern to Geoffrey E. Hinton that the excellent performance of these networks is actually a serious problem. These networks somehow learned to con themselves into believing the presence of context with every output. Hinton in his paper [Dynamic Routing Between Capsules., 2017] talks about changing this phenomenon of invariance with equivariance. He proposed the idea of capsules and changed the scalar output to vector outputs. These vectors when manipulated changed the input to the upper capsules thus leading to equivariance. He then used an algorithm named Dynamic Routing to route the output of lower layer capsules to the required upper layer capsules bypassing any other inputs. This paper follows a similar approach with little tweaks in the final architecture. We introduce Katz Backoff technique to select the type of N-gram (unigram, bigram, trigram, etc). The change in the architecture was done keeping in mind the actual orientation of how the brain works. The final capsule layer consists of clustered capsules with similar vector lengths. We show that not only the idea of probability disagrees with the actual brain working but also that clustering similar capsules gives us better results.

II. PROPOSED MODEL

Our capsule network, presented here is a variant of the capsule network presented in Wei Zhao et al., 2018. The proposed network consists of four layers: n-gram convolutional layer, primary capsule layer, clustered capsule layer and fully connected capsule layer.

A. N-gram Convolutional Layer

This layer is a basic convolutional layer which first calculates the Katz Backoff n-gram and then extracts n-gram features at every position possible in a sentence through various convolutional filters.

We denote $x \in \mathbb{R}^{L \times V}$ denotes the input sentence representation where L is the length of the input sentence and V is the size of the embedding word vector corresponding to the i-th word in the sentence [Wei Zhao et al., 2018.]. Let $x_i \in \mathbb{R}^{K_1 \times V}$ be the filter, where K_1 is the size of N-gram calculated from Katz Backoff equation:

$$P_{katz}(Wn|WW_{n-N+1}^{n-1}) = P^*(W_n|W_{n-N+1}^{n-1}) \quad \alpha \quad (W_{n-N+1}^{n-1}) \quad P_{katz}(W_n|W_{n-N+2}^{n-1})$$

Equation 2.1.1

P* denotes discounted probability.

Equation 2.1.1 shows that Katz Backoff probability for an N-gram just relies on the (discounted) probability P* if we've seen this N-gram before (i.e., if we have non-zero counts). Otherwise, we recursively back off to the Katz probability for the shorter-history (N-1)-gram. Here, α denotes normalizing factor. A filter W^a is convoluted with a word window $x_{i:i}$+L-K_1+1 at each possible position (stride 1) and it produces a column feature map $\mathbf{m}^a \in \mathbb{R}^{K_1+1}$, each element $m_i^a \in \mathbb{R}$ of the feature map is produced by

$$m_i^a = f(x_{i:i+K1-1} \circ W^a + b_0)$$

Equation 2.1.2

where ° is element wise multiplication by W^a+b_0, b_0 is a bias term and f is a nonlinear activation function (i.e., leaky ReLU).

B. Primary Capsule Layer

This is going to be the first capsule layer in which we preserve the instantiated parameters of a word such as the order of the word, the tag of the word according to POS. The idea is to replace the scalar output of CNNs with vector output.

Suppose, $p_i \in \mathbb{R}^d$ [Wei Zhao et al., 2018] denotes the instantiated parameters of a capsule, d denotes the dimension of the capsule. Let $W^b \in \mathbb{R}^{B \times d}$ be the filter for different sliding windows. We slide the window over each word vector (found by N-gram) for their matrix multiplication. The filter W^b multiplies each N-gram vector in $\{Mi\}_{i-1}^{L-K1+1}$ with stride of 1 to produce a column list of capsules.

The capsules are then calculated by:

$$p_i = W^b M_i + b_1$$

Equation 2.2.1

b_1 is the capsule bias term.

C. Relationship with Upper Layer Capsules

As mentioned in Hinton's paper, the lower level capsules learn about the nature of their parent capsules by probabilistic evaluations. It allows the lower level capsules to send their outputs to the appropriate parent capsules. This is done with the help of transformation matrices.

The prediction vector $\hat{u}_{j|i}$ is calculated by multiplying a weight matrix W_{ij} with the vector u_i from the lower capsules. The total input to a capsule s_j is the summation of all the prediction vectors $\hat{u}_{j|i}$.

$$s_j = \sum c_{ij} \hat{u}_{j|i} \quad , \quad \hat{u}_{j|i} = W_{ij} u_i$$

c_{ij} is the coupling coefficient.

Arguably, since brain signals do not travel from neuron to neuron using probabilities it is likely that we look for a more "real-life" approach.

Clustering approach is introduced at this point. Instead the length of the output vector denoting the probability of the presence of an entity, we can cluster those output vectors whose values are in close proximation with each other. In this way we can create a group of similar instantiated parameters. This group can then project a vector to one particular parent capsule which will be able to detect that particular instantiated feature.

Algorithm: Dynamic Routing Algorithm

1. **procedure** D_ROUTING($\hat{a}_{j|i}$, r, l)
2. $b_{j|i} \leftarrow 0$:
3. **for** r iterations **do**:
4. **for** all capsule i in layer l and j in layer l+1:
   $c_{j|i} = \hat{a}_{j|i} \times$ leaky_softmax($b_{j|i}$)
5. **for** all parent capsules v_j in layer l+1
6. **if**:
   $\| v_{j1}\|^2 \cong \| v_{j2}\| \cong \ldots\ldots \| v_{jn}\|^2$
7. $B_j = \prod_{i=1}^{n} vjn$
8. $C_j = g(B_j)$
9. **for** all capsules i in layer l and capsules j in layer l+1:
   $b_{j|i} = b_{j|i} + B_j \times v_j$
10. **return** B_j, C_j , v_j

We have eliminated the prediction process in the above algorithm and replaced it with a coupling process.

v_j is the parent-capsule. What we have done here is, collected all the parent capsules with similar lengths. Then we have multiplied them with each other to produce a joint cluster B_j. We then squash the results of this cluster using the squashing function proposed by Sabour et al. (C_j)

Once all the parent capsules are produced and coupled the coupling coefficient is updated by

$$b_{j|i} = b_{j|i} + B_j \times v_j$$

D. Fully Connected Capsule Layer

In this layer the capsules are flattened and fed into a fully connected capsule layer. The capsules are multiplied by transformation matrices. The final capsule $v_j \in \mathbb{R}^d$ is produced here.

Published By:
Blue Eyes Intelligence Engineering &
Sciences Publication (BEIESP)

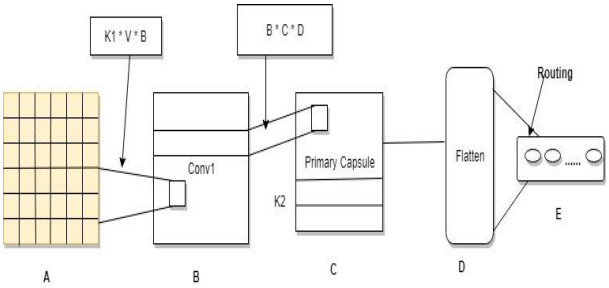

Figure 2: Architecture of Capsule Network for text classification. The working of all the layers are mentioned above

III.EXPERIMENTAL SETUP

A. Experimental Details

We evaluated our model with a series of datasets. The datasets are movie reviews (MR) (Pang and Lee, 2005), Stanford Sentiment Treebank extension (SST-2), customer review (CR). These datasets contains several bench marks such as movie reviews, sentiment analysis and customer reviews.

Table 1: Datasets used for Experimental Setup

Dataset	Train	Dev	Test	Classes	Classification Task
MR	8.8k	0.9k	1.3k	2	Review classification
SST-2	8.6k	0.9k	2.0k	2	Sentimental analysis
CR	3.5k	0.3k	0.6k	2	Review classification

B. Implementation Details

450-dimensional word2vec vectors were used to initialize embedding vectors. Mini-batch with size of 30 was conducted for the datasets. Adam optimization was used with a learning rate of 1.5 to train the model. 3 iterations of routing were used for all the datasets.

C. Neural Nets for Comparison

In the experiments, neural networks such as LSTM, Bi-LSTM, LR-LSTM(LSTM regularized by linguistic knowledge) and CNN-rand were used for comparison with the proposed network.

IV. RESULTS

In the experiments done it was recorded that the capsule network performed exceptionally well in all the datasets, better than all other neural networks applied on them. This shows the effectiveness of the proposed clustered capsule network architecture. The exceptional results achieved were the result of the implementation of Katzz Backoff for N-gram modelling which is a crucial part of natural language modelling. Implementation of the clustered capsule network has also helped in boosting the result. This can be observed by comparing the results of our paper and the results obtained in Wei Zhao et. al paper.

Table 2: Comparison of the Results

	MR	SST-2	CR
LSTM	77.1	81.2	75.7
Bi-LSTM	79.3	80.1	79.1
LR-LSTM	81.5	87.6	83.4
Capsule	84.3	88.4	89.1

V.CONCLUSION

In this paper, we successfully implemented the Katz Backoff algorithm before evaluating the N-gram model. We also changed the window size for the word vector rotation. This reduction in window size with stride of 1 helped in deeper evaluation of the instantiated parameters. Finally, we argued the usage of probabilities to find the outputs of the parent capsules and instead approached it with a cumulation of vectors of similar length and then projecting the final output v_j.

REFERENCES

1. Geoffrey E. Hinton, Sara Sabour, Nicholas Frosst. 2017 Dynamic Routing Between Capsules.
2. Wei Zhao, Jianbo Ye, Min Yang, Zeyang Lei, Soufei Zhang, Zhou Zhao. 2018 Investigating Capsule Networks with Dynamic Routing for Text Classification.
3. Jaeyoung Kim, Sion Jang, Sungchul Choi. 2018 Text Classification using Capsules
4. Edgar Xi, Selina Bing, Yang Jin. 2017 Capsule Network Performance on Complex Data
5. Speech Language Processing: Daniel Jurafsky, Steve H. Martin

AUTHORS PROFILE

Madhuram M SRM Institute of Science and Technology, B.Tech, Faculty of Computer Science Department.

Mayukh Dasgupta SRM Institute of Science and Technology, B.Tech Computer Science Third Year, Publications: 'Water Distribution using Machine Learning Techniques', IJETER Vol-6 Issue 10 October 2018.

Aqib Muhammed Ashik BT SRM Institute of Science and Technology, B.Tech Computer Science Third Year Publications: 'Finger Recognition and Gesture Based Augmented Keyboard' VOl- 4 Issue-5, 2018.

SuryaM SRM Institute of Science and Technology, B.Tech Computer Science Third Year, Publications: 'Finger Recognition and Gesture Based Augmented Keyboard' VOl- 4 Issue-5, 2018