**CS 482/682 Final Project Report: Text Categorization**

Cong Gao, Zhaohao Fu, Liujiang Yan

**Abstract:** Text categorization is a fundamental task in area of natural language processing. In this report, we introduce several deep neural networks based approaches, utilizing different representation for text, such as word-embedding based recurrent convolutional neural network and character level convolutional neural network . We evaluate the performance on public datasets of different sizes and types. Some variants of architecture that utilizes small kernel size and skip layer connection are also evaluated.

1. **Problem Statement**

Text categorization is an essential task in natural language processing with various applications such as information filtering, sentiment analysis, and text-based searching. It is a typical classification problem similar as image classification task in computer vision. A key difference, which restricts the direct deployment from those well successful image classification methods, is that the representation of text is character based but not numerical value based. Previous works have been done to find a good feature and representation for text that we could utilize for machine learning method. In this project, we try two different representations and corresponding approaches, one of which is word based representation and the other is character based representation.

To demonstrate and compare the effectiveness, performance, and efficiency of the proposed methods, we perform the experiments suing the following datasets: Twitter Comments, UCI news, and AG news. See the following table for detailed information about each of the datasets.

**Twitter Comments:** This dataset contains comments from Hillary Clinton and Donald Trump. We will use this dataset to see if the proposed method is sufficient for this task.

**UCI News:** This dataset contains English news abstract of multiple major topics. We will use this dataset for tuning the hyperparameters and architectures.

**AG News:** This dataset contains title and description from news articles from more than two thousand news sources. We will use it to evaluate the performance at final step.

1. **Technical Approach**
2. Recurrent Convolutional Neural Networks (RCNN)

**Word Embedding** The representation the method utilizes word embedding for representation, which is a real valued vector per word and enables us to measure the semantics relatedness by the distance between vectors. In our implementation, we use pre-trained model by Standard NLP group.

**Architecture** Figure 1. Shows the architecture of the RCNN, which consists of several components. The first one is a bidirectional recurrent network, where the sequence input is the sequence of word embeddings given by the text, and the hidden variables stand for left and right contextual information correspondingly. Then we stack the corresponding left and right contextual hidden variables with the input and form a feature vector. Then a fully connected layer connect the feature vector to a hidden layer and a max pooling layer is performed to capture the max responses at each position. The output is same size with the classes’ number and a log softmax activation function is performed to get the final output. Since it is a classification problem and the final layer is log softmax, we use negative log likelihood as our loss function.

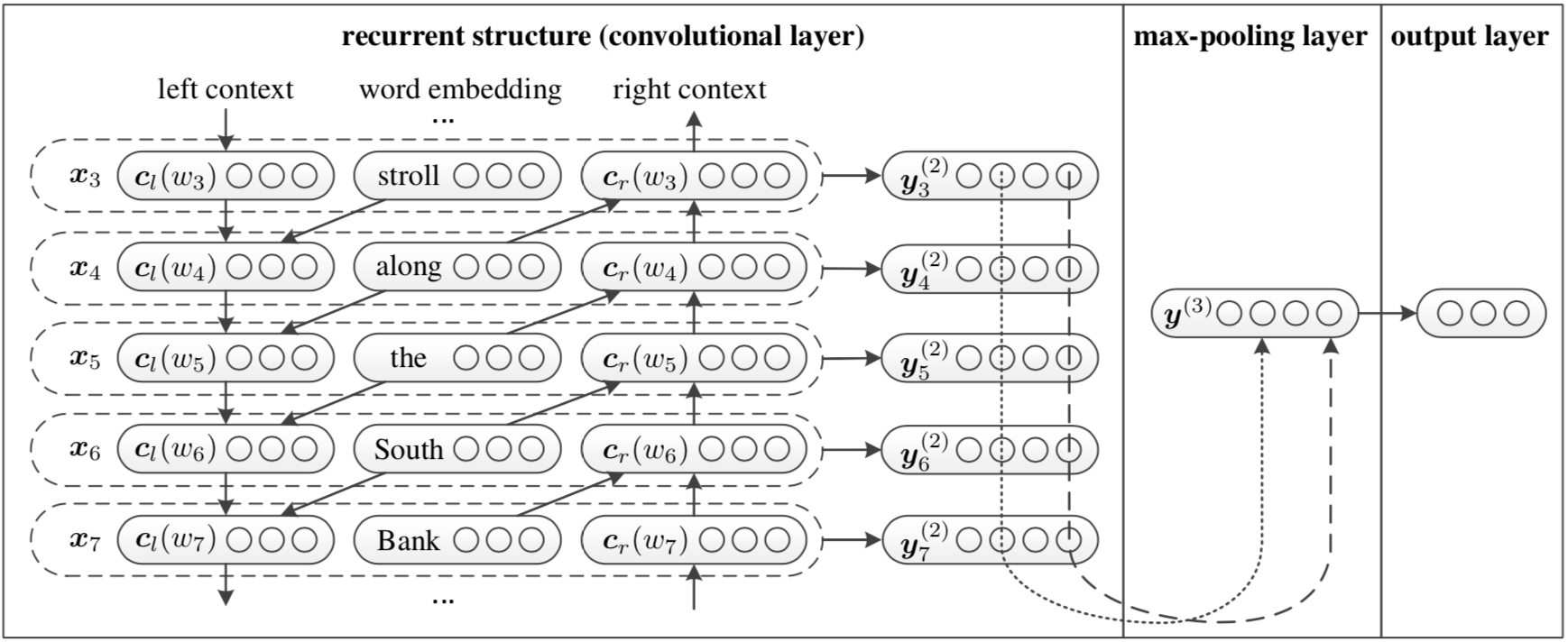


Fig 1. Architecture of the RCNN

**Variants**

1. **Zero-Padding**: The number of words per sentence is various therefore the length of representations are variant, leading to difficulty in using mini-batch training. We introduce a variant in representation by adding zero padding in the end to a fixed length. Noting that we are performing max operation at hidden layer, and the activation function from outputs of RNN to hidden layer is originally **tanh** and zero padding might change the max response. Therefore, we correspondingly change the activation function to ReLU.
2. **Deep Text Learning**: The original net connects the outputs of RNN to final prediction by a shallow connection and it could be enhanced to a deep connection by multiple 1d convolutional layers and max pooling layers.
3. Character-level Convolutional Neural Networks (Char CNN)

**Character quantization** The representation here is character level. For each given text, it forms a fixed size binary matrix that each column is one hot vector size 1-of- with one for the character’s index in a given alphabet and zero otherwise; and the total row number is fixed to , any character exceeding length is ignored. Thus, the sequence of characters is transformed to a sequence of such sized vectors with fixed length .

**Architecture** Figure. 2 presents the architecture of Character-level CNN, consisting of convolutional layers, max pooling layers and fully connected layers. As discussed above, the final layer is a log softmax for multiple class classification task, and we use negative log likelihood as our loss function.

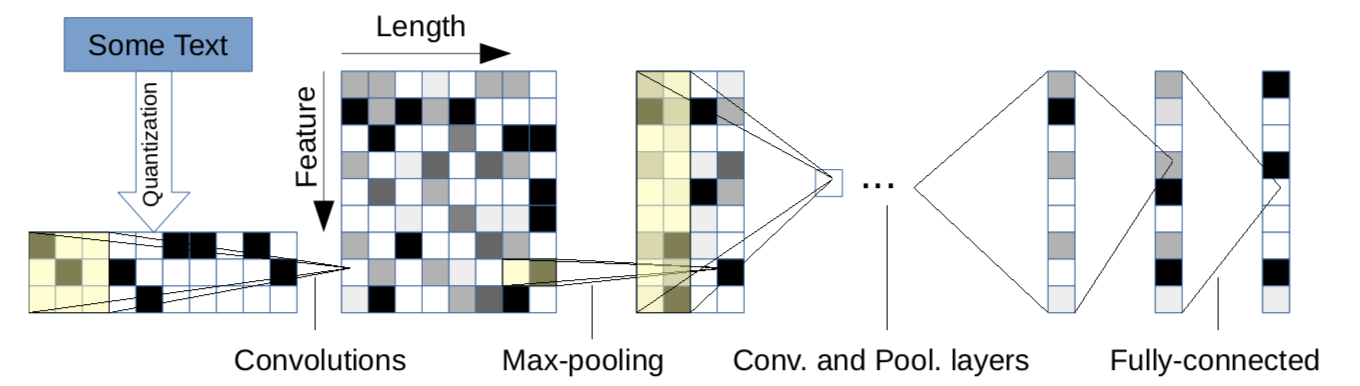


Fig 2. Architecture of the Character-level CNN

**Variants**

1. **Batch Normalization:** Some of our dataset have short length in input, and might lead to gradient vanishing given this deep network. Therefore we add batch normalization in each conv layer.
2. **Kernel Size**: Inspired by VGG net, a composition of two conv layers of kernel size as 3 have same reception field of one conv layer of kernel size 7, but has more nonlinearity. Therefore, we replace the large kernel conv layer with composition of small kernel conv layers, but keep the pooling layers.
3. **Skip Layer Connection**: The other approach to apply deep net for simple / small data is to perform deep residual learning. We introduce the skip layer connection as basic block, and we also remove the fully connected layer to reduce the number of parameters.
4. **Results**

The experiments are performed on a local machine, MacBook Pro (15-inch, Early 2013, quad core 3th Intel Core CPU). The whole pipeline is implemented on PyTorch and we decouple the code into components: configuration, dataset, models, etc.

To evaluate the proposed methods’ performance, we first shuffle the dataset and spilt each one to 80% as training and 20% as validation. We finally test the performance on AG news by splitting to 60% as training, 20% as validation and 20% as test. To evaluate the accuracy, we simple count the ratio of correctly classified samples.

**Tweets Comments**

As we introduce, this dataset is relatively small, and we use it to demonstrate whether the proposed architecture is sufficient for this task. For the following we show each method’s loss plot and accuracy plot.

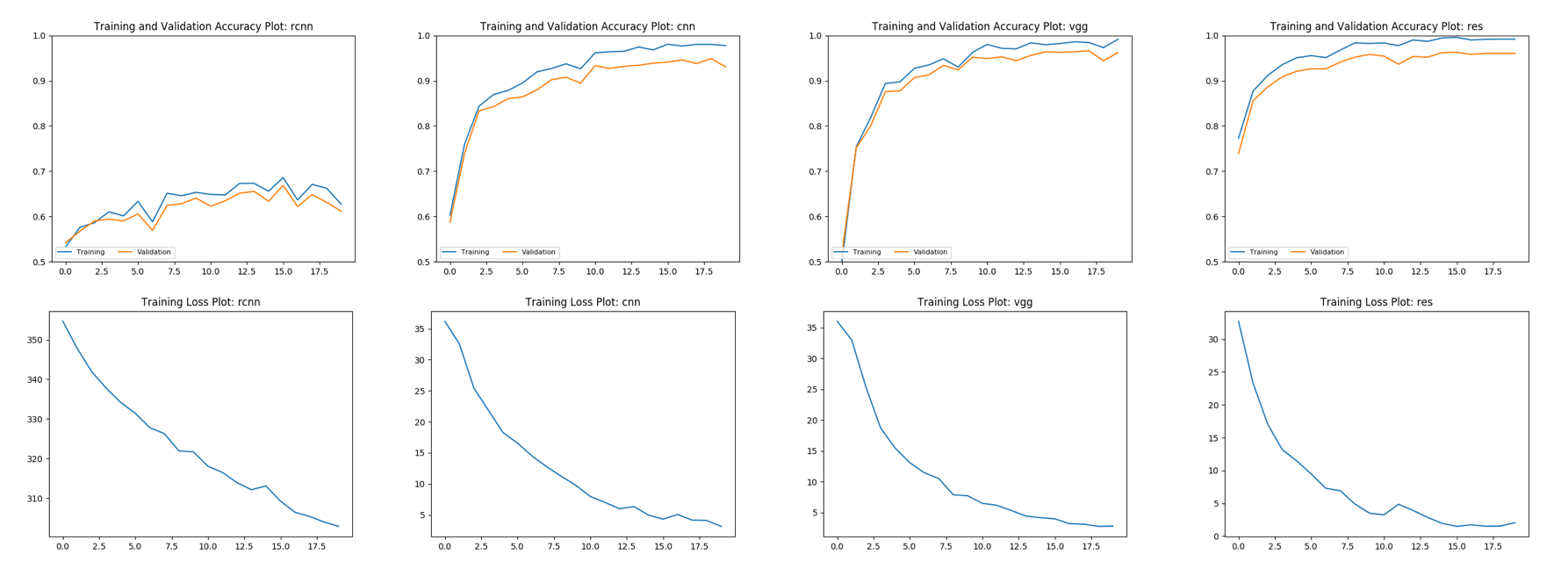


Figure 3 Accuracy and Loss Plot on Tweets Dataset

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | RCNN | Character CNN | VGG | ResNet |
| Accuracy | 0.65 | 0.93 | 0.96 | 0.96 |

Table 1 Accuracy on Tweets Dataset

As the results show, character level CNN and its variants all achieve more than 90% classification accuracy in this dataset, and the introduction of small kernel and skip layer connection leads to about 4% accuracy improvement. The original RCNN method takes too much time to train due to single sample batch, therefore we only perform experiment by zero padding and use fixed length of 10 words and batch size of 10. Through the plot we shall see that, this method does not achieve similar result as char CNN, part of the reason might be the introduced zero padding.

**UCI news**

The UCI news dataset is relatively large and has multiple classes, therefore we use it to demonstrate whether the proposed architecture can scale and achieve comparative accuracy.

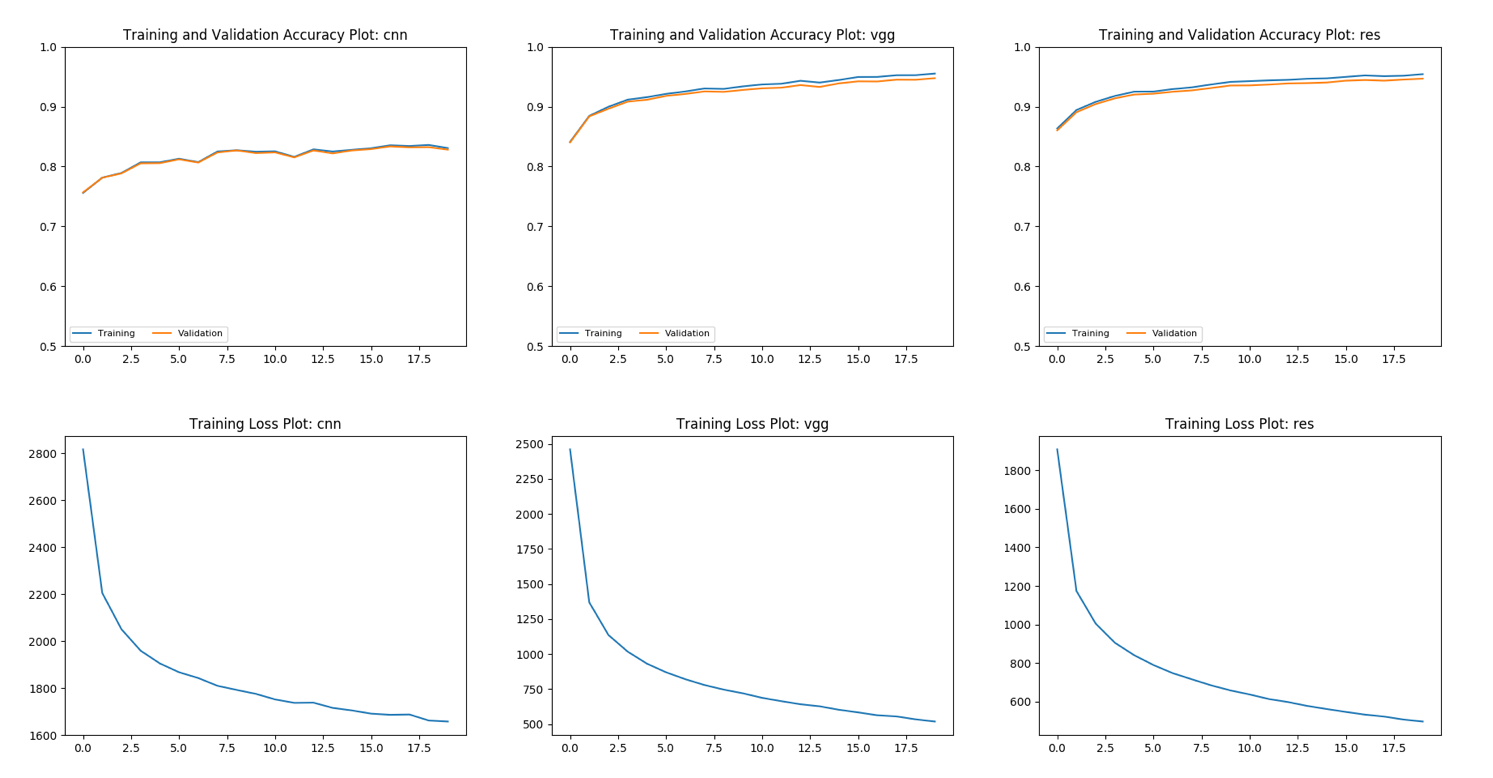
****

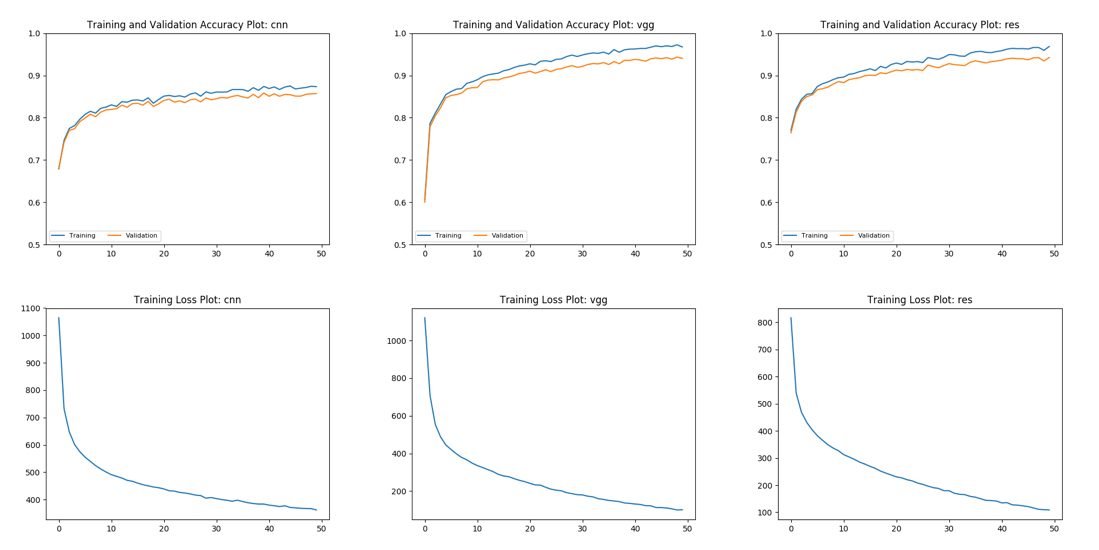
Figure Accuracy on UCI news dataset

|  |  |  |  |
| --- | --- | --- | --- |
|  | Character CNN | VGG | ResNet |
| Accuracy | 0.828 | 0.947 | 0.946 |

Table 1 Accuracy on Tweets Dataset

**AG news**

We use AG news dataset to compare each method’s performance, by using same training data and validation data, and we test on a separated dataset.



|  |  |  |  |
| --- | --- | --- | --- |
|  | Character CNN | VGG | ResNet |
| Validation Accuracy | 0.857 | 0.940 | 0.942 |
| Test Accuracy | 0.776 | 0.838 | 0.833 |

Table 1 Accuracy on AG news Dataset

**Summary**

Through the performance of the proposed methods we shall see, character level convolution neural networks and its variants outperform word-level recurrent convolution neural network, in aspects of both accuracy and performance. Also, by introducing small kernel, batch normalization, and skip layer connection, the modified architectures achieve significant improvement while keep similar training and inference efficiency.

1. **Discussions**

Though there exists significant difference between image classification task and text categorization task, we shall see that, through appropriate representation, some general network architectures and principles could seamlessly adapt the text categorization task. For the variants of character level convolutional neural networks, through replacing the large kernel conv layer with small kernel conv layer, we reduce the number of parameters while remain the same receptive field and introduce more non linearity. Through introducing batch normalization layer after convolution layers, and skip layer connection as residual learning, we alleviate the gradient vanishing problem that the network could adapt to datasets of different scale without tuning the network. Also, through the similar performance of word level based method and character based method, deep neural networks again show the ability to capture and learn the representation.

There exists several limitations while the performance on these tasks seems promising.

Through the implementation of the project, we have the chance to apply deep learning to tasks other than vision. Also, since we build the whole pipeline from ground, we learn a lot about how to design the learning system and manipulate different components to work.

1. **Conclusion**

We have developed the pipeline for text categorization, and also implemented the word level recurrent neural networks and character level convolution neural networks as baseline method. We also introduce several variants inspired by well-performed convolution networks architecture for vision tasks. We evaluate the proposed methods’ performance on dataset of different scales and types, and it shows that our introduced variants have gained significant improvements than the baseline methods, in both performance and efficiency.

1. **Hyperparameters**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | RCNN | Character CNN | VGG | ResNet |
| Learning Rate | 0.01 | 0.05 | 0.05 | 0.05 |
| Momentum | 0.9 | 0.9 | 0.9 | 0.9 |
| Epoch | 50 | 50 | 50 | 50 |

Table Hyperparamters for each architecture

1. **Reference**

[1] Lai, S., Xu, L., Liu, K. and Zhao, J., 2015, January. Recurrent Convolutional Neural Networks for Text Classification. In AAAI (Vol. 333, pp. 2267-2273).

[2] Zhang, X., Zhao, J. and LeCun, Y., 2015. Character-level convolutional networks for text classification. In Advances in neural information processing systems (pp. 649-657).

[3] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

[4] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.