Week 6-3: Paper Summaries

CE-510 Seminar: Social Media Mining

Student Name: Jiaqi Guo NetID: JGR9647

■ Character–level Convolutional Networks for Text Classification

On the one hand, the current text classification technology mainly considers words or combinations of words; On the other hand, research shows that convolutional neural networks are very useful in extracting information from original signals. In this paper, the author treats the character-level text as the original signal and uses a one-dimensional convolutional neural network to process it. Research shows that word embedding representation can be directly used in convolutional neural networks without considering the grammar or semantic structure of language.

This paper, using only characters, applies to the convolutional neural networks. The authors find that when training large data sets, deep convolutional neural networks do not require word-level meaning (including grammar and semantics of language). This is a very exciting simplification of engineering because no matter what language it is, it is made up of characters, so it is crucial for building systems across languages. As a bonus, the model can still handle unusual character compositions (such as spelling errors) and emoticons.

1. The Character-level Convolution Networks

Suppose we have a discrete input function, $g(x)$, whose values range from the real numbers in $[1, L]$. There is a discrete kernel function $f(x)$ whose values range from the real numbers in $[1, k]$. The convolution $h(x)$ of $f(x)$ and $g(x)$ with step d can be expressed as:

$$h(y) = \sum_{x=1}^{k} f(x) \cdot g(y \cdot d - x + c) \qquad h(y) \in [1, \lfloor \frac{l - k + 1}{d} \rfloor] \to \mathbb{R}$$

Where $c = k - d + 1$ is an offset constant. Just like the traditional convolutional neural network in computer vision, the model is parameterized by a series of kernels, which we denote as $f_{ij}(x)$ when the input is $g_i(x)$ and the output is $H_j(y), (I = 1, 2, \ldots, m,$ and $j = 1, 2 \ldots, n)$. We call each g_i an input feature, each h_j an output feature, m an input feature size, and n an output feature size.

In order to train a deeper network, the author proposes the concept of maximum pooling operation, which can

be expressed as:

$$h(y) = \max_{x=1}^{k} g(y \cdot d - x + c)$$

Possible Improvement Directions:

1. The length of character-level text is extremely long, which is not conducive to handling long text categories
2. Only character-level information is used, so the model uses less semantic information

■ Bag of Tricks for Efficient Text Classification

Fasttext is a simple and effective method for sentiment classification and word vector representation. It is comparable in accuracy to some deep learning classifiers, and many orders of magnitude faster in training and testing than deep learning models.

1. Hierarchical softmax

   When there are many target categories, the computation of linear classifier is very large, k is the categories, d is the dimension of hidden layer, and the time complexity is $O(kd)$. In order to improve the time efficiency, this paper adopts hierarchical Softmax method based on Huffman coding tree, and the time complexity is reduced to $O(d\log_2(k))$. In a tree structure, the target is the leaf node.

2. N-gram features

   The word bag model ignores word order but considering word order increases time complexity. In this paper, n-gram mechanism is used as an additional feature to obtain some local information about word order. This works just as well in practice as using word order directly.

   By using the trick of hash, this paper realizes a fast n-gram space mapping with high space utilization. When n is 2, only $10M$ is needed, and in other cases $100M$.

Possible Improvement Directions:

1. The model structure is simple, so it's not the optimal model at the moment.
2. Semantic information acquisition is limited because of the word bag idea.