Stacked Convolutional Neural Network

Zeqiang Lai
Department of Computer Science
Beijing Institute of Technology
1120161865@bit.edu.cn

Jinxuan Jin
Department of Computer Science
Beijing Institute of Technology
1120161864@bit.edu.cn

Wenzhuo Liu
Department of Computer Science
Beijing Institute of Technology
1120161868@bit.edu.cn

Tian Huan
Department of Computer Science
Beijing Institute of Technology
1120161861@bit.edu.cn

Anteng Li
Department of Computer Science
Beijing Institute of Technology
1120161866@bit.edu.cn

Xueyan Guo
Department of Computer Science
Beijing Institute of Technology
1120162336@bit.edu.cn

Abstract

In this paper, we review the convolutional neural network in detail. In addition, we discuss several method to regularization. Based on the basic building blocks, we discuss and analyze a recent implementation called De-CNN for aspect extraction.

1 Introduction

Convolutional Neural Network(CNN) have proven to be powerful in the field of computer vision, especially in image recognition systems. Since the success in visual imagery, convolutional neural network and its variants have been widely used in many areas including natural language processing(NLP).

Recently, a novel approach[1] for aspect extraction was reported. In this model, the authors discard the traditional RNN architectures and only use several CNN layer to handle the features extraction. Inspired by this work, we consider it to be a kind of more general structures called stacked convolutional neural network(SCNN) which can be applied to capture spatial or sequential relationships.

2 Model

The stacked convolutional neural networks is a composition of a number of single convolutional layer. Each layer learns some features from previous layer and higher-level features are obtained with the increase of the number of layers.

The general structure of stacked convolutional neural network can be described as follow. See figure1

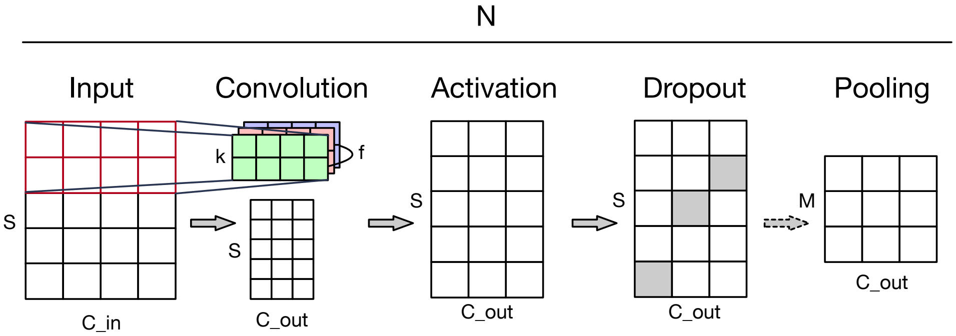

Figure 1: Overview of stacked convolutional neural network.

A basic unit of SCNN usually consists of a convolutional layer and a activation layer and optionally contains a pooling layer and a dropout layer. And one or many different types of the basic unit are stacked together to form the full network.

Convolutional layer: The fundamental element of convolutional layer is the filter. A filter is collection of weights which can be performed ”dot-products” with inputs to generate a new feature. The filters scan across all samples and integrate the information of samples that are close to each other. Filter weights are shared in each integration, which results in less parameters and faster calculation.

Activation layer adds the non-linearity to the networks to enhance the representation ability. There are various types of activation functions we can use, such as softmax, tanh, ReLU and variants of ReLU. The common choice is ReLU which results in faster training.

Dropout layer: Dropout is a kind of regularization approaches that is applied in training process to prevent overfitting. Here is the basic idea. Fisrt, randomly select part of neurons(often determined by a ratio ρ) and ignore their activations during training. Second, use all neurons but scale the activations by ρ during testing.

Pooling layer is often used in vision tasks (usually not in NLP tasks). It applies non-linearity downsampling on activation maps, discard or refine some information and produce a smaller feature map that can speed up training.

3 De-CNN

Dual embedding convolutional neural network(De-CNN) is proposed for the aspect extraction task. The overview of its structure can be seen in figure2. CNN layers part is an example of stacked convolutional neural network.

3.1 Analyze

The authors adopt two types of embeddings, domain embedding and general embedding to encode words in sentences. These embeddings are concatenated together and then processed by two CNN layer with the same number of filters (128) and different kernel size(5 and 3), respectively. The output of these CNN layers are concatenated again as the input of a stack of 3 identical convolutional layers with 256 filters and $kernel\_size = 5$. In order to keep the length of sequences, different paddings are used.

Assume we have a batch of sentence with shape $[batch\_size, seq\_len]$.

- After encoding and concatenation, the input becomes a 3D tensor with shape $[batch\_size, seq\_len, gen\_emb + domain\_emb]$.

- Two 1D convolutions are applied and the shape of output is $[batch\_size, seq\_len, 128 + 128]$.
- The shape of tensor remains the same in the latter convolutions.
- At the end of network, a fully-connected layer transform the input into a tensor with shape $[batch\_size, seq\_len, num\_label]$ to make predictions.

3.2 Parameters

The dimensions of general embedding and domain embedding are 300 and 100. Hence, the filter sizes of first two CNN layers are $(300 + 100) * 5$ and $(300 + 100) * 3$. Since there are 128 filters for each layer, the total number of parameters in first two CNN layers is

$$128 * (300 + 100) * 5 + 128 * (300 + 100) * 3 = 409600$$

For the same reason, the number of parameters in the latter layers(3 CNN) is

$$3 * (256 * (256 * 5)) = 983040$$

The total number of parameters in CNN layers is

$$409600 + 983040 = 1392640$$

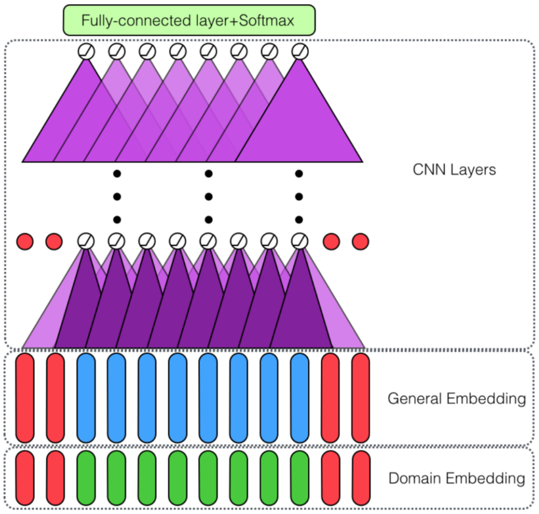

Figure 2: Overview of De-CNN

4 Experiment

In our experiment, we replicate the result of De-CNN for aspect extraction and try to apply a similar structure to named entity recognition task.

References

[1] Hu Xu et al. "Double Embeddings and CNN-based Sequence Labeling for Aspect Extraction". In: *arXiv preprint arXiv:1805.04601* (2018).