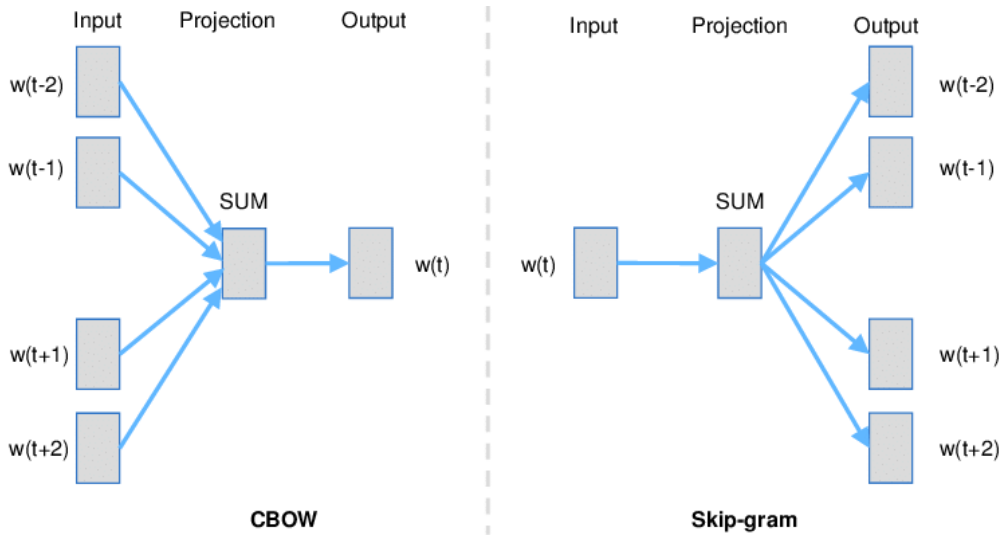
First of all, on the concept of computational graphs. TensorFlow is a computational graph model, that is, a model that expresses the operation process in the form of a graph. NLP is the abbreviation of Natural Language Processing, a technology for studying the computer processing of human language. At present, TensorFlow is the most active among all open source projects on the open-source community Github. From its launch to the present, it has undergone several versions of evolution. A large number of practical problems can already be solved efficiently and flexibly. This article will try to explain the simple application of TensorFlow in the field of NLP. In the field of NLP, the language model is the most basic link. This article will cover deep-learning-related models such as word2vec, RNN, LSTM.

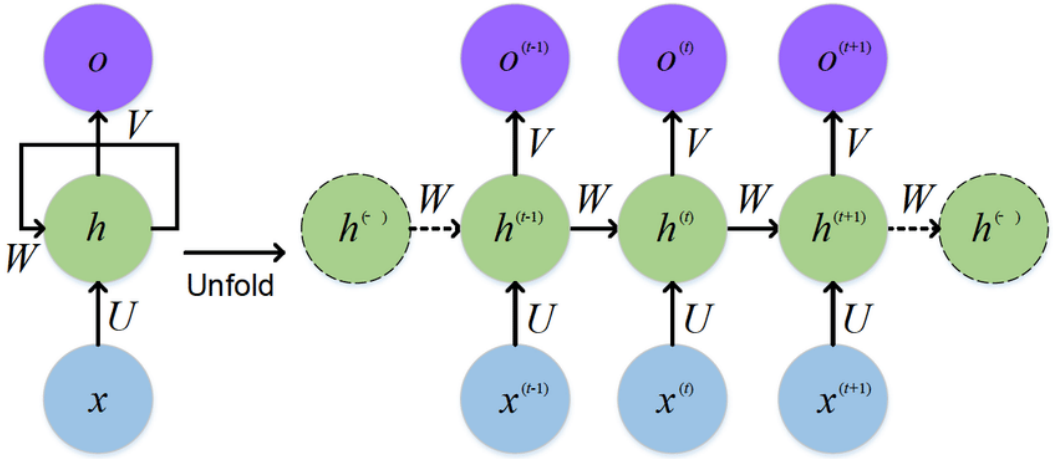
Word2vec, two models are used in Word2Vec, the CBOW model (Continuous Bag-of-Words model) and the Skip-gram model (Continuous Skip-gram Model). An example of the model is as follows, which is a neural network model with a three-layer structure, including an input layer, a projection layer, and an output layer.



(reference 1)

The CBOW model predicts the current word based on the context of the word, while the Skip-Gram model, on the contrary, predicts the context based on the current word. Given a corpus as the training set, the vector representation of each word can be trained through the above model.

RNN, Recurrent Neural Network. The traditional neural network model is from the input layer to the hidden layer and then to the output layer. The nodes between each layer are disconnected. This ordinary neural network does not have the memory function, and the recurrent neural network RNN is to solve this kind of problem. , It is memorable and is usually used to deal with time series problems. For example, if you want to predict what the next word of a sentence will be, you generally need to use the previous word, because the preceding and following words in a sentence are not independent. The reason why RNN is called a recurrent neural network is that the current output of a sequence is also related to the previous output. The specific form of expression is that the network will memorize the previous information and apply it to the calculation of the current output, that is, the nodes between the hidden layers are no longer unconnected but connected, and the input of the hidden layer not only includes the output of the input layer It also includes the output of the hidden layer at the previous moment.



(Reference 2)

However, RNN has a shortcoming. Although it can connect the previous information to the current input, if the time span between the current input and the previous information is large, due to gradient attenuation and other reasons, the ability of the RNN to learn such far information will decrease. This problem is called the Long-Term Dependencies problem.

LSTM is a kind of RNN with selective memory function, it can effectively solve the problem of long-term dependence, and can learn the key information before. RNN has a problem. When backpropagating, the gradient will also decay exponentially. As a result, the gradient after many

stages of propagation tends to disappear, and it cannot handle the problem of long-term dependence. Although RNN can theoretically handle sequences of any length, it is difficult for RNNs to handle sequences longer than 10 in practice applications. In order to solve the problem of the disappearance of the RNN gradient, the Long Short-Term Memory module is proposed, which realizes the memory function on the sequence through the switch of the gate. When the error propagates back from the output layer, it can be recorded using the memory element of the module. So LSTM can remember information over a relatively long period of time.

When using TensorFlow to deal with deep learning-related issues, we don't need to pay too much attention to its internal implementation details. We only need to focus on the construction of the model and use the abstract unit structure already provided by TensorFlow to build a flexible model. It is precise because of the high level of abstraction of TensorFlow that sometimes it is quite difficult for people to understand. So in the process of using TensorFlow, do not define the problem too deeply. All data can be regarded as Tensor. Use Tensor's operators to operate on it. Don't think about the details of the operation in your mind.

References:

1. https://www.researchgate.net/figure/Continuous-Bag-of-words-CBOW-CB-and-Skip-gram-SG-training-model-illustrations_fig1_326588219
2. https://www.researchgate.net/figure/The-standard-RNN-and-unfolded-RNN_fig1_318332317
3. http://www.tensorflow.org
4. https://www.tensorflow.org/versions/r0.8/tutorials/recurrent/index.html#recurrent-neural-networks
5. http://colah.github.io/posts/2015-08-Understanding-LSTMs/