Deep Learning - Final Project

Team members：綦家志、何庭昀

Title：Using RNN to solve sentence similarity problem

Abstract:

In kaggle data analysis competition, there is a topic we are interested in, Quora Question pairs. The main purpose of this problem is to predict whether two sentences have the same meaning. After this project, we can construct a method and a code to solve sentence similarity problem.

Keyword: sentence similarity, natural language processing

Introduction:

Quora is a place to gain and share knowledge about anything. It’s a platform to ask questions and connect with people who contribute unique insights and quality answers. Over 100 million people visit Quora every month, so it's no surprise that many people ask similarly worded questions. Multiple questions with the same intent can cause seekers to spend more time finding the best answer to their question, and make writers feel they need to answer multiple versions of the same question.

Recently, Quora uses a Random Forest model to identify duplicate questions. In this problem, we are challenged to tackle this natural language processing problem by applying advanced techniques to classify whether question pairs are duplicates or not. Doing so will make it easier to find high quality answers to questions resulting in an improved experience for Quora writers, seekers, and readers.

Purpose:

1. Construct sentences similarity model.
2. Predict whether question pairs are duplicates or not.

Problem Formulation:

The goal of this problem is to predict which of the provided pairs of questions contain two questions with the same meaning. The training data similarity labels have been supplied by human, which are inherently subjective, as the true meaning of sentences can never be known with certainty. Human labeling is also a 'noisy' process, and reasonable people will disagree. As a result, the similarity labels on this dataset should be taken to be 'informed' but not 100% accurate, and may include incorrect labeling. Data given in this problem is (1.) a training data with 404290 pairs of sentences and (2.) a testing data with 2345796 pairs of sentences.

For example, in training data, we have

1. “What is the step by step guide to invest in share market in India?”

   “What is the step by step guide to invest in share market?”

   “0”

2. “Astrology: I am a Capricorn Sun Cap moon and cap rising... what does that say about me?”

   “I'm a triple Capricorn (Sun, Moon and ascendant in Capricorn) What does this say about me?”

   “1”

The labels “0” means previous two sentences are not duplicates, and “1” means previous two sentences are duplicates.

In testing data, we have

1. “How does the Surface Pro himself 4 compare with iPad Pro?”

   “Why did Microsoft choose core m3 and not core i3 home Surface Pro 4?”

2. “Should I have a hair transplant at age 24? How much would it cost?”

   “How much cost does hair transplant require?”

Our final purpose is to construct a method and a code to predict whether question pairs are duplicates or not.

Method:

We decide to use google library, word2vec, to transform every word in sentences to a distributed represent vector, which means we can evaluate correlation of two words by their vectors. And then, we train LSTM RNN by sending each pair of vector lists (two sentences) to LSTM RNN to get two final RNN outputs, evaluate distance between these outputs, compare with the training data labels, and update LSTM RNN weights and biases. After finishing the training, we send sentence pairs from testing data to trained LSTM RNN, and evaluate similarity of each sentence pair.

Reference:

[1] Jonas Mueller and Aditya Thyagarajan, “Siamese Recurrent Architectures for Learning Sentence Similarity,” Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)