Using Machine Learning to Determine Identical Internet Questions

Miles A. Curry
CS 434 - Machine Learning and Data Mining
School of Computer Science
Oregon State University
Corvallis, Or 97331
currymi@oregonstate.edu

Christian Armatas
Oregon State University
Corvallis, Or 97331
armatasc@oregonstate.edu

Abstract

Internet questions answer sites like Stack Overflow, Quora, Yahoo Answers, are a popular destination of the web. Determining duplicate or identical questions increases site usability and performance. This paper looked at determining question duplication by machine learning. It used data provided by Quora through Kaggle.com and used a siamese neural network to determine question equivalence. We resulted in about 54% training accuracy. The current algorithm needs improvements including data processing optimization, validation sets, and further tinkering off loss and distance functions.

1 Introduction

How can you determine if two or more questions are similar? Stated another way, if two questions are duplicates and are similarly worded, how can we reduce the time readers spend searching for the best answer to their question, and the time writers spend providing answers to multiple versions of the same question. This problem was provided to us via kaggle.com's Prediction Competition "Quora Question Pairs."

In this document, we will discuss our approach to the *question pairing* problem, including data processing, encoding, our neural network design and final accuracy results, and any problems and lessons learned from the project.

2 Approach

2.1 Data Processing and Encoding

Question data was encoding using a bag-of-words technique. We used each individual question sets as their on bag-of-words rather then the entire corpus question bag-of-words. Our thinking behind this is that two questions are determined equal by looking only at each question in the pair and not by looking at other question pairs.

Furthermore we analyzed questions to determine their parts of speech using the NLTP and Gensim Python modules. Our hope was to further encode this information for the neural network to characterize. We believed that the more information that the neural network has, the better. We also believe that some parts of speech may hold better indicators to determining question relatedness. For instance take the following unrelated question set:

`What is the step by step guide to invest in share market in India?`

and

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

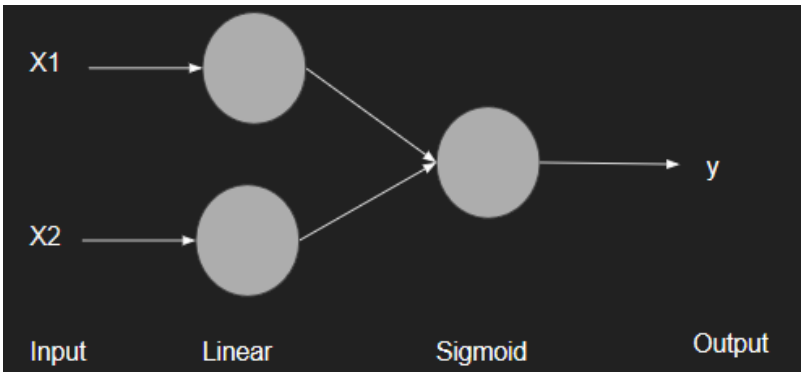

Figure 1: The design of our Siamese neural network.

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

Almost identical questions, the difference difference comes is the single noun `India`. With more testing and analysis, we would like to see if encoding parts of speech different presents better accuracy.

2.2 Neural Network Design

Based off of the design done by Homme et. al.[3], we created a two-layer Siamese neural network 1. We choose to do a Siamese neural network for several reasons. First, we needed to separate the two inputs and do feature recognition. For this we choose a linear regression activation function. We assumed linear regression would be capable of determining important features of each question.

For the next layer, we used a sigmoid activation function to make a prediction on whether the two inputs were identical or not.

To calculate loss, we used a contrastive loss function as found in [3] and [2]. We're not sure this loss function is the best loss function for us, so we are considering using other loss functions. We implemented in terms of time.

To calculate distance, we used a euclidean distance function. With euclidean distance we are achieving 50% - 54% training accuracy depending on number of epochs and question set. We then tested the training data using a cosine difference and we achieve 17% accuracy. Though at the time of writing, the author forgot that inverting our predicted solutions would result in a 83% accuracy.

A dropout rate of 0.1 was used.

2.3 Neural Network as Code

To code our neural network we used the Keras deep learning library for python and we based our code from their siamese neural network example [1].

3 Results

At the time of writing we only were able to produced training results for our neural network. We were not able to do any testing or validation sets and were not able to produce guesses for the testing data provided on kaggle.

Using euclidean distance over all inputs (n = 395,548. 8000 entries were removed for encoding issues) and 20 epochs we were able to achieve 54.84% training accuracy. With n = 10,000 and 20 epochs we achieved 52.81% training accuracy.

4 Problems, Lessons Learned and Future Improvements

One of the biggest current problems is that data processing and encoding can be incredibly complex and not optimized; the current run time is worse than cubic and consumes the majority of the time for processing. First, we must find a way to improve the processing of our network in order to help speed up testing and tinkering.

One of the lessons we learned was that taking the dot product of different sized 1D arrays isn't possible. To handle this, we padded the data with zeros onto the end of the data and truncated long data, however we realize this may have had an impact on our results.

We still need to add a lot of more features and do more iterations of testing and training. The following we still needs to be done:

- Validation sets - n-fold
- Try different distance functions
  - See if cosine distance works better then euclidean
- Try different loss functions
- See if there's a way to encode parts of speech
  - play with parts of speech weights to get better accuracy
- Validation, validation, validation!

5 Conclusion

Our solution is a good start, but definitely needs more testing and validation. Further more tinkering needs to be done over iterations and we need to run our. We're happy with our results so far, and believe that applied to testing data that after tinkering that they could produce decent results. To reiterate we used a two layer siamese neural network over encoded question sets to predict if two question were ultimately related. Future research needs to be done in validation, and determine a more accurate distance and loss calculations.

Individual Contribution Levels

A Individual Contribution Levels

- Research
  - Miles - 95
  - Christian - 5
- Paper
  - Miles - 50
  - Christian - 50
- Design
  - Miles - 100
  - Christian - 0
- Code
  - Miles - 100
  - Christian - 0
- Presentation
  - Miles - 90
  - Christian - 10

References

[1] Keras source code.

[2] Yann LeCun Raia Hadsell, Summit Chopra. Dimensionality reduction by learning an invariant mapping. 9 2005.

[3] Christopher Yeh Yushi Homma, Stuart Sy. Detecting duplicate quesitons with deep learning. 2017.