Machine Learning Engineer Nanodegree

Capstone Proposal

Zitong Guo
August 21st, 2017

Proposal

Domain Background

Given an integer sequence: 1, 2, 3, 4, 5, ? So what is the next number?

If your answer is 7, You read that correctly. That's the start to a real integer sequence, the powers of primes[1]. Continuously, how about the next number in 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ? If you answered 89, you may enjoy this challenge. In this project, a machine learning solution will be demonstated to predict the next number of a given integer sequence.

A number in a sequence is equivalent to a word. Based on this observation, we try to predict the last number based on the preceding numbers. Hence, the problem can be treated as the Natural Language Processing (NLP)[2] domain.

NLP is the computerized approach to analyzing text that is based on both a set of theories and a set of technologies. Definitionally speaking, NLP is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications.

The history of NLP generally started in the 1950s, although work can be found from earlier periods. During the 1970s, many programmers began to write "conceptual ontologies", which structured real-world information into computer-understandable data. Up to the 1980s, most NLP systems were based on complex sets of hand-written rules.

Recent research has increasingly focused on unsupervised and semi-supervised learning algorithms. Such algorithms are able to learn from data that has not been hand-annotated with the desired answers, or using a combination of annotated and non-annotated data.

In recent years, there has been a flurry of results showing deep learning techniques achieving state-of-the-art results in many natural language tasks, for example in language modeling, parsing and many others, among which, sequence prediction is different from traditional classification and regression problems. It is required to take the order of observations into account and apply models that have memory and that can

learn any temporal dependence between observations.

Problem Statement

This problem at hand is defined by Kaggle[3] team's competition named Integer Sequence Learning[4]. It challenges you create a machine learning algorithm capable of guessing the next number in an integer sequence. While this sounds like pattern recognition in its most basic form, a quick look at the data will convince you this is anything but basic!

Datasets and Inputs

The dataset[5] of this project contains the majority of the integer sequences from the On-Line Encyclopedia of Integer Sequences® (OEIS®)[6]. It is split into a training set, where you are given the full sequence, and a test set, where we have removed the last number from the sequence. The task is to predict this removed integer.

The input dataset of this project is two CSV files for `train` and `test`. It's known that for the training set, we are given the entire sequence and for the test set the final element has been removed, which is the target we are trying to predict.

We explore the training set to understand the characteristics of the datasets using R.

```
train <- read.csv("train.csv")
str(train)
```

```
## 'data.frame':    113845 obs. of  2 variables:
##  $ Id      : int  3 7 8 11 13 15 16 18 20 21 ...
##  $ Sequence: Factor w/ 112880 levels "-1,-1,-1,-1,-1,-1,-1,-1,8,-1,-10,-19,-28,-3
```

```
head(train)
```

```
##   Id
## 1  3
## 2  7
## 3  8
## 4 11
## 5 13
## 6 15
##   Sequence
## 1  1,3,13,87,1053,28576,2141733,508147108,402135275365,1073376057490373,970038548
## 2  1,2,1,5,5,1,11,16,7,1,23,44,30,9,1,47,112,104,48,11,1,95,272,320,200,70,13,1,1
## 3  1,2,4,5,8,10,16,20,32,40,64,80,128,160,256,320,512,640,1024,1280,2048,2560,409
## 4  1,8,25,83,274,2275,132224,1060067,3312425,10997342,36304451,301432950,17519415
## 5  1,111,12211,1343211,147753211,16252853211,1787813853211,196659523853211,216325
## 6  1,1,1,1,1,1,1,1,5,1,1,1,1,5,5,1,1,1,1,11,5,5,11,5,1,1,1,1,5,23,5,23,5,5,1,1,
```

It's obersed that each row of the data contains `Id` and `Sequence`. There are totally 113,845 sequences indicated by `Id`.

Solution Statement

The Recurrent Neural Networks[7] approach – usually just called "RNNs" - can be applied to solve this problem. This task particularly interests me as it's analogous to word prediction. Hence integers are treated as words in the solution.

Benchmark Model

The `Mode` methodology is used as the benchmark model for the last number prediction in a certain sequence. For this, we simply find the mode in a given sequence, and that will be our guess for the last term in the sequence. The Mode Benchmark (implemented in R) seen on the competiton leaderboard[8] has an accuracy of `0.05746`.

```
Mode <- function(x) {
    ux <- unique(x)
    ux[which.max(tabulate(match(x, ux)))]
}
```

The top 20 accuracy scores of the competition leaderboard range between 0.20 - 0.59 (excluding an outlier in first place with a score of 0.98).

Evaluation Metrics

The evaluation metric for this problem is straightforward and simple. It is based on the accuracy of the predictions (the percentage of sequences where the next number is predicted correctly).

Project Design

In this final section, summarize a theoretical workflow for approaching a solution given the problem. Provide thorough discussion for what strategies you may consider employing, what analysis of the data might be required before being used, or which algorithms will be considered for your implementation. The workflow and discussion that you provide should align with the qualities of the previous sections. Additionally, you are encouraged to include small visualizations, pseudocode, or diagrams to aid in describing the project design, but it is not required. The discussion should clearly outline your intended workflow of the capstone project.

Data Preprocessing

Reading the provided CSV file produces a data frame of two variables, Id and Sequence. The Id variables are integers, and are exactly how we want them. The Sequence variable is in strings, so we will need to convert that to a list of numbers. Relative methods to deal with this kind of scenarios will be applied.

Build and Train Model

As mentioned above, a number in a sequence is equivalent to a word. Based on this observation, we try to predict the last number based on the preceding numbers. Hence, the problem can be treated as the Natural Language Processing (NLP) domain.

It's known that RNNs can use their internal memory to process arbitrary sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition. Hence, the RNNs approach will be applied in the project to handle the NLP domain problem.

Model Evaluation

The prdiction accuracy (i.e., the percentage of sequences where the next number is predicted correctly) can be applied to evaluate the designed model.

The big question of this investigation, is whether this model can be used to predict the last term of a given sequence accurately. As the nature of the challenge was a contest, the predictions created by the model will be submitted online on Kaggle for a blind evaluation and then returned an accuracy score.

---

1. "Powers of primes. Alternatively, 1 and the prime powers (p^k, p prime, k >= 1). (Formerly M0517 N0185)", *https://oeis.org/A000961* ↩
2. Natural language processing from Wikipedia, the free encyclopedia, *https://en.wikipedia.org/wiki/Natural*language*processing* ↩
3. Kaggle Official Website, *https://www.kaggle.com* ↩
4. Kaggle Integer Sequence Learning Competition, *https://www.kaggle.com/c/integer-sequence-learning* ↩

5. Kaggle Integer Sequence Learning Dataset, *https://www.kaggle.com/c/integer-sequence-learning/data* ↩

6. On-Line Encyclopedia of Integer Sequences® (OEIS®) Official Website, *https://oeis.org* ↩

7. Recurrent Neural Networks from Wikipedia, the free encyclopedia, *https://en.wikipedia.org/wiki/Recurrent*neural*network* ↩

8. Kaggle Integer Sequence Learning Competition Leaderboard, *https://www.kaggle.com/c/integer-sequence-learning/leaderboard* ↩