AI Quantification of Language Puzzle to Language Learning Generalization

A Project Report

Presented To

Dr. Chris Pollett

Department of Computer Science

San Jose State University

In Partial Fulfillment

Of the Requirements for the

Class CS 297

By

Harita Shroff

December 2019

Abstract

Online language learning applications provide users multiple ways/games to learn a new language. Some of the ways include rearranging words in the foreign language sentence, filling in the blanks, providing flashcards and many more. Primarily this research focuses on quantifying the effectiveness of these games in learning a new language. Secondarily my goal for this project is to measure the effectiveness of exercises for transfer learning in machine translation. As part of CS297, I worked on projects involving different artificial intelligence topics. Mainly the focus was on different types of neural networks and their implementations. Language modeling and sequence-to-sequence learning were topics providing possible insight into the end goal of the project.

TABLE OF CONTENTS

1. Introduction

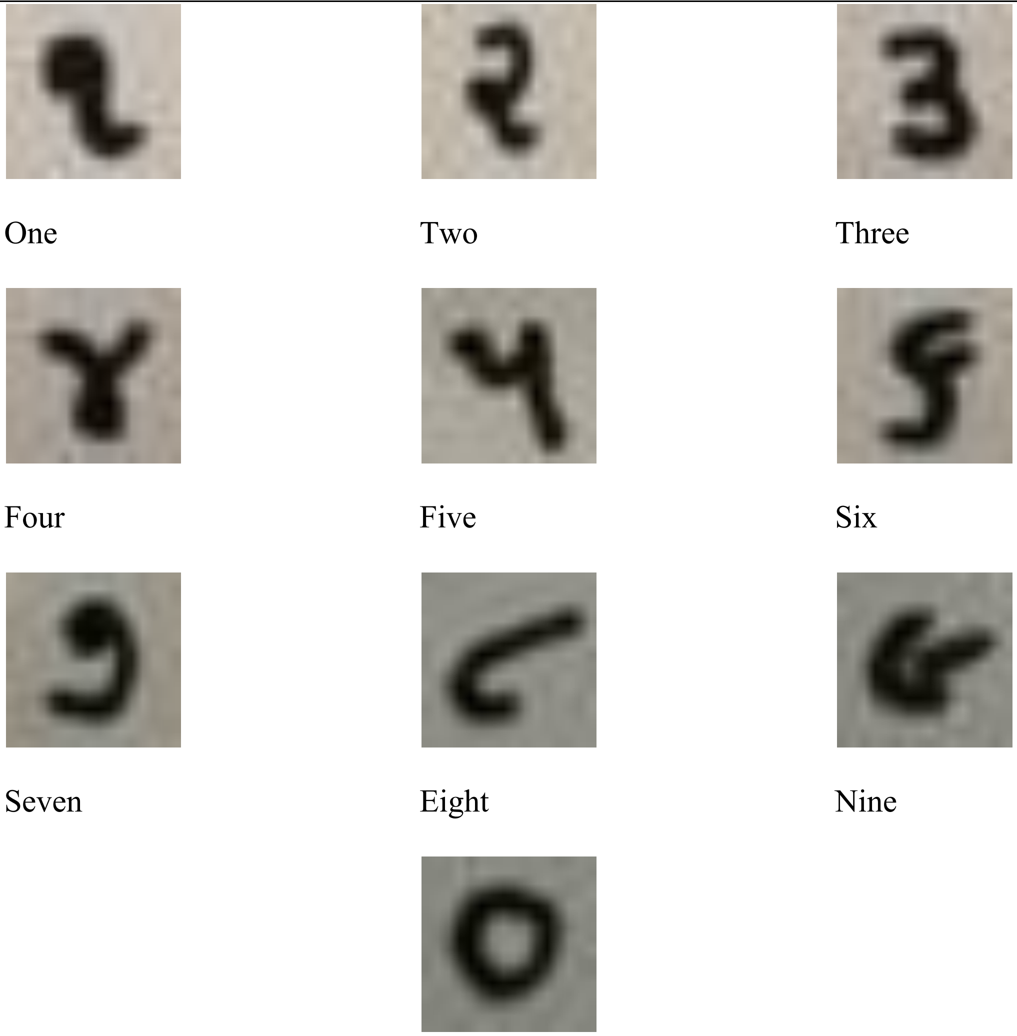
Learning a foreign language is always a challenging task. The spread of the internet not only connected different parts of the world but also provided some powerful tools. Online language learning is one such tool people find appealing nowadays. There are many tools available on the internet which can help a user learn a new language at their own pace. According to [5], very little research has been done to study the effectiveness of these platforms. These platforms use a different set of activities including language puzzles to help its users learn a new language. The goal of this research is to quantify the effectiveness of these methods using artificial intelligence technology.

Artificial intelligence will be used to measure the effectiveness of these platforms. One of the key techniques for this project will be machine learning language models. According to [1], instruments that want to translate sentences from one language to the other need to have the capability to differentiate between sentences. The language model is the formalization of this idea. As mentioned in [2], translating a sentence from one language to the other is easy if the structure of the languages is similar. In a case where structures are different, and sentences are taken as a sequence, sequence-to-sequence technique can be applied for the translation.

This semester, the goal was to understand the application of machine learning technology in the field of language learning. I learned different types of neural networks which include feed-forward, convolution, and recurrent neural network. Along with these fundamentals, language models and sequence-to-sequence learning were also explored. At last, having a good dataset is critical for any machine learning project so various dataset collections are also discussed in this project.

2. Deliverable 1: Feed-Forward Neural Network

As stated in [1], it is standard to get started with a deep learning introduction with image processing. The first deliverable for this project was to identify Gujarati digit images using the simple feed-forward neural network. Here is the sample dataset used for this exercise:



One

Two

Three

Four

Five

Six

Seven

Eight

Nine

Zero

Fig 1. Sample Dataset for Deliverable 1

Figure 1 contains 16x16 images of Gujarati digits starting from 0 to 9. These were the captured images of handwritten digits. Using the PIL python library, the images were converted to black and white images and 256-pixel values were extracted. Each pixel value ranged from 0 to 255 where 0 indicating a total black pixel while 255 indicates a white pixel. Any number between 0 to 255 indicates a variation of a grey pixel. These values were fed into the computational layer whose output was then fed into a softmax layer to convert images to the probability distribution. A loss function in a machine learning model provides a comparative measure of how good or bad a model is. The goal is to minimize the loss. The process of going from input pixels to the loss function is called the forward pass. The values calculated here will be used in the backward pass which used the stochastic gradient descent method. The result of this deliverable was computed using the number of correct answers model produced. The accuracy was used to validate the model.

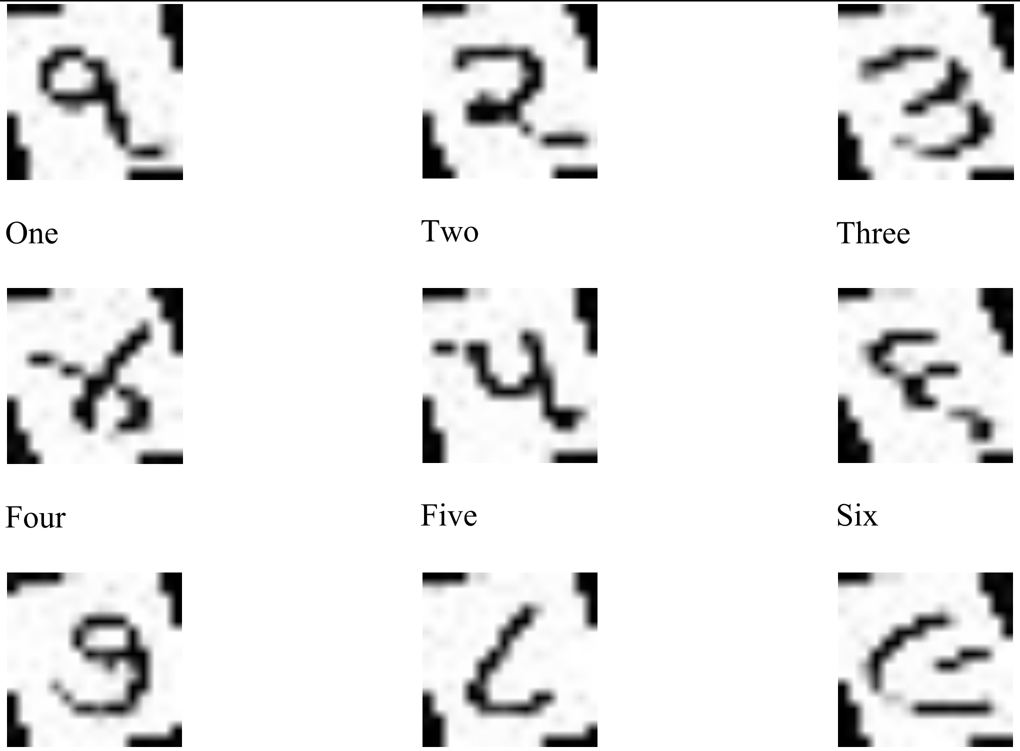
This deliverable's implementation was done in python using the Tensorflow library. Training and testing data were loaded with an independent python program written to mimic the existing implementation of the MNIST data loader. Though the result of this deliverable did not meet the expectation, mainly due to the low volume of the dataset, it was still a very important step in getting started with machine learning. Figure 2 is the result for this deliverable.

```
> python tensorflow_ffnn.py

2019-12-02 18:09:09.983627: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU
supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA

2019-12-02 18:09:10.004569: I tensorflow/compiler/xla/service/service.cc:168] XLA service
0x7fa0775955e0 executing computations on platform Host. Devices:

2019-12-02 18:09:10.004618: I tensorflow/compiler/xla/service/service.cc:175]   StreamExecutor
device (0): Host, Default Version

Test Accuracy: 0.10000000149011612
```

Fig 2. Feed-Forward Neural Network Output

3. Deliverable 2: Convolution Neural Network

The aim of this deliverable was to get introduced to the concept of a convolution neural network. The objective was the same as Deliverable1 to identify Gujarati digits from their images. The only difference between datasets compared to deliverable-1 was the dataset volume. There was a total of 7200 images used for this deliverable. The Google Image website was used as a source instead of using handwritten images. Only 10 test images and 10 train images for 10 Gujarati digits were downloaded from Google Image. Using the pillow (PIL) library, these 20 images were converted to 7200 images by rotating each image 360 degrees. Figure 3 is the sample dataset containing rotated images.



One Two Three

Four Five Six

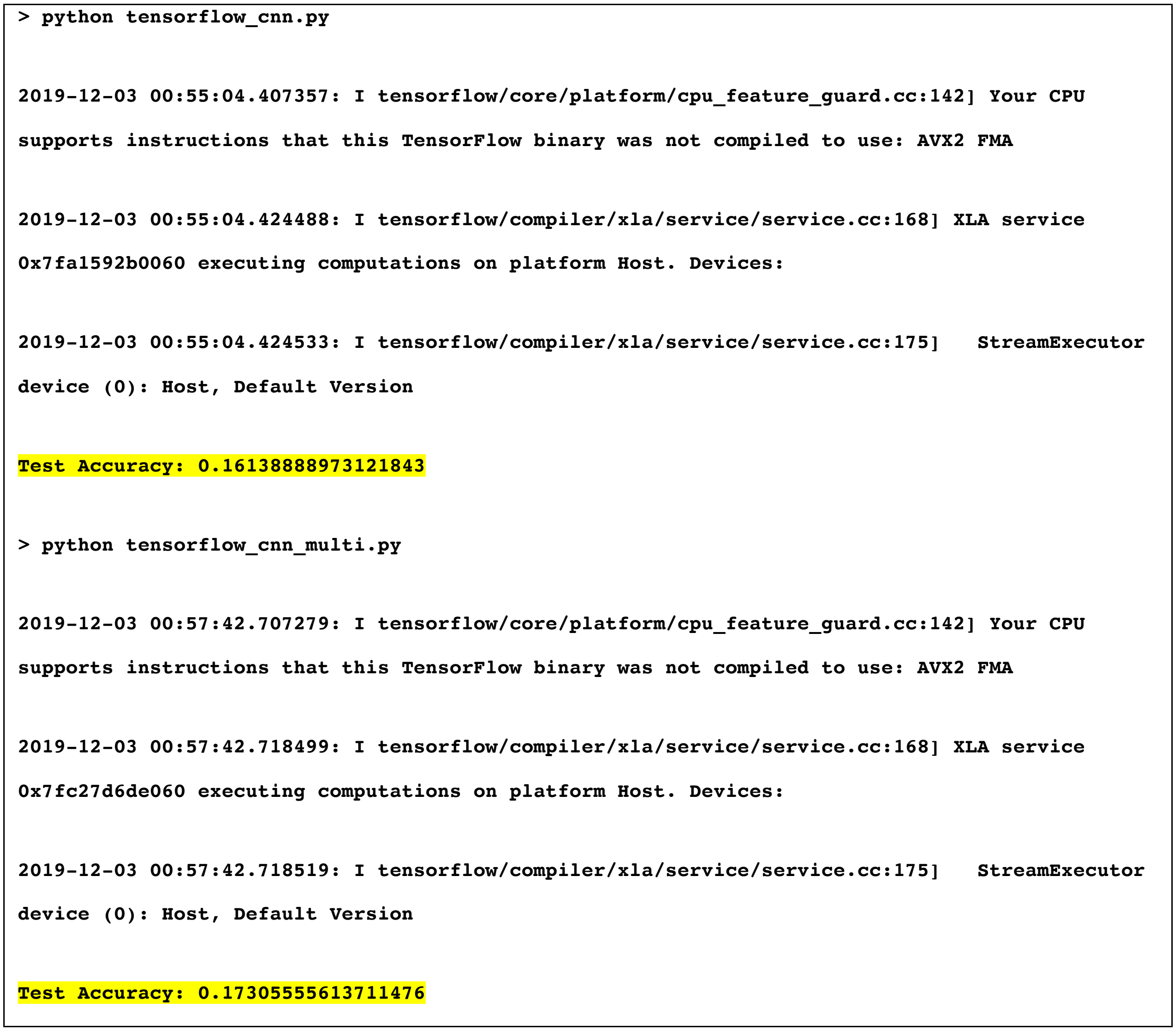
Seven Eight Nine

Zero

Fig 3. Sample dataset for deliverable 2.

Convolution neural networks operate on the concept of filters. Feed-forward neural networks are fully connected where each output from a layer corresponds to each input of the next layer. Convolution neural networks work on a concept of partially connected layers. Partial connectivity can be compared to data features requested through filters. Each image pixel is processed with a predefined filter to extract certain features from the image. Convolution filters are defined using their size, and its channel. Using the concept of stride and padding, filters can be applied to the images. Stride decides which image pixel rows and columns to process. Padding covers the image boundaries.

For this deliverable, a single layer as well multi-layers convolution neural network was implemented. The rectified linear unit (relu) was used as an activation function. The filter for single-level convolution was of 4 x 4 in size. A total of 4 such filters were used. Both the horizontal and vertical strides were set to 2. “Same” padding was used which would have padded image boundaries with 0s. For a multi-level convolution, a total of three filters were used of size 4 x 4, 2 x 2 and 1 x 1. Each had a different stride extracting different features of the image. The same model accuracy was used to compare the results of Deliverable1 and Deliverable2. Multiple epochs and different filter sizes and strides improved the result of image recognition compared to Deliverable1.



```
> python tensorflow_cnn.py

2019-12-03 00:55:04.407357: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU
supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA

2019-12-03 00:55:04.424488: I tensorflow/compiler/xla/service/service.cc:168] XLA service
0x7fa1592b0060 executing computations on platform Host. Devices:

2019-12-03 00:55:04.424533: I tensorflow/compiler/xla/service/service.cc:175]   StreamExecutor
device (0): Host, Default Version

Test Accuracy: 0.16138888973121843

> python tensorflow_cnn_multi.py

2019-12-03 00:57:42.707279: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU
supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA

2019-12-03 00:57:42.718499: I tensorflow/compiler/xla/service/service.cc:168] XLA service
0x7fc27d6de060 executing computations on platform Host. Devices:

2019-12-03 00:57:42.718519: I tensorflow/compiler/xla/service/service.cc:175]   StreamExecutor
device (0): Host, Default Version

Test Accuracy: 0.17305555613711476
```

Fig 4. Convolution Neural Network and Multilevel Convolution Neural Network Output

4. Deliverable 3: Word Embeddings

The aim of this deliverable was to understand language modeling through deep learning techniques. According to Eugene Charniak [1], a language model is a probability distribution over all strings in a language. A program must differentiate between sentences whiles translating from one language to the other. Similar language models will be used as part of the CS 298 project to understand the current implementation of online language learning platforms.

Language models are designed by associating the probability of the next word given the previous words. As deep learning models cannot operate on the words, a floating-point vector is associated with each word. These vectors are called word embeddings. The deliverable was aimed to find the word embeddings for the set of words in the Gujarati language. If two different words are followed by the same words, then these two words get similar word embeddings. This similarity is measured through the cosine similarity index. The same set of words already had their word embeddings calculated for the English language. For the Gujarati language, Wikipedia articles containing these words were selected as a test corpus. The text was divided by the space and punctuations to create a word index. As Gujarati words cannot be processed as is, first each word was decoded using inbuilt python library. Tensorflow was used to generate the word embedding for a pair of words. Table 1 shows the achieved result from this deliverable.

Word	**Largest Cosine Similarity**
"ઉપર"	**0.49136004 ("મશીન")**
"નીચે"	**0.5024193 ("મશીન")**
"યાદ"	**0.500106 ("કમ્પ્યુટર")**
"કહે"	**0.5183961 ("કમ્પ્યુટર")**
"કમ્પ્યુટર"	**0.50508183 ("યાદ")**
"મશીન"	**0.5218789 ("નીચે")**

Table 1. Word embeddings for Gujarati words.

5. Deliverable 4: Dataset Collection

This project aims to understand the effectiveness of the applications employed by the online language learning platforms. The dataset required for this task has to be collected from one of the platforms. The ideal dataset would include user data where all conditions were recorded. Here conditions being the type of exercises, learning language, native language, total times word was encountered by the user, total time user was correct about the word and user identity to create a profile and understand the effectiveness. Duolingo has a dataset similar to this project's requirement. According to [3], the dataset contains 13 million Duolingo student learning traces. Each record consists of proportion of exercises indicating the exercise where the word was correctly used, activity timestamp, time since last activity included the same word, user identity, language user trying to learn, user's native language, word, total number of time user has seen the word, and total number of time user has correctly identified the word.

The second dataset contains information about 7 million words from 6000 plus users collected over a period of 30 days according to [4]. The dataset is divided between English, French and Spanish words. First-line in the dataset contains information about the user identity, user country, number of days from when a user has started learning this language, session type, activity format and time stamp. Each line followed by the first line contains unique id, the word, part of the speech in universal dependencies format, morphological features, dependency edge label, and edge head. Each unique id contains session information, index of activity within a given session and word position in the current activity.

Both datasets are sourced from a well-known language learning platform that validates their authenticity. The amount of data available is also large enough to divide data for any training and testing machine learning model. Datasets may require some modification depending on the research's focus while currently means to extract data is provided by sources in both cases.

6. Conclusion

As part of CS297, I learned different machine learning techniques. During the semester, as part of my work on this project, I explored feed-forward, convolution, and recurrent neural networks. I also studied different language model techniques and sequence-to-sequence mapping. I used TensorFlow for the coding of these machine learning techniques. Tensorflow was used as a third-party python library.

The first deliverable covered the Gujarati digit recognition using a feed-forward neural network. As the accuracy for the model was really low, convolution neural network was applied on the same dataset and accuracy improvement was noticed. While learning language models as part of deliverable 3, first a set of English words were converted to the Gujarati language. Wikipedia articles in Gujarati were used as test corpus for this deliverable.

The topics covered as part of CS297 will be used intensively during the CS298 project. The ultimate goal for the next phase is to quantify the effectiveness of language puzzles used in the online language learning platforms. Language puzzles use sentences from a user's native language and map it to a foreign language. Concepts like language models and sequence-to-sequence learning can be used to measure the effectiveness of language learning platforms. As mentioned in deliverable 4, datasets from well-known language learning platforms will be explored during CS298 work.

References

[1] E. Charniak, Introduction to Deep Learning, ISBN: 9780262039512192 pp. | 7 in x 9 in75 b&w illus. January 2019.

[2] S. Saini and V. Sahula, “A Survey of Machine Translation Techniques and Systems for Indian Languages,” in IEEE Int. Conf. on Comp. Int. & Comm. Tech., 2015.

[3] Settles and Burr, "Replication Data for: A Trainable Spaced Repetition Model for Language Learning", https://doi.org/10.7910/DVN/N8XJME, Harvard Dataverse, 2017.

[4] Settles and Burr, "Data for the 2018 Duolingo Shared Task on Second Language Acquisition Modeling (SLAM)", https://doi.org/10.7910/DVN/8SWHNO, Harvard Dataverse, 2018.

[5] R. Vesselinov and J. Grego, “Efficacy of New Language App”, 2015.