CS410 - Technology Review- Text Mining RNNLM

Reynold Chan

November 7, 2021

Introduction

A major aspect of text retrieval systems and text mining models consist of understanding the likelihood of the word's appearance. There are many methods as to the prediction of the appearance of a word within a sentence. An example of this need could be given that my word is "computer" what is the most likely word to appear next within the sentence ? Could it be keyboard, screen, laptop etc... Thus it is imperative that these methods are robust in their predictions.

Two prominent methods in the generation of these language model include using statistical language models which uses probabilistic rules and laws, and recurrent neural networks which is based off principles in machine learning and neural networks. This article will explore in particular the different statistical language model that are the most common, and the principles behind the recurrent neural network language models(RNNLM). It will conclude with an analysis on the benfits and drawbacks behind RNNLM.

Statistical Language Models

Statistical language models are the most popular form of language models to this date for the simplicity. Given a collection of documents , one can generate a statistical language model. There are several types of statistical language models. The following are some of the most common and a short description of each of these models. [1]

- Uni-gram Given a collection of words, a probability distribution can simply be drawn by the occurrences of the word in the collection.
- Bi-gram An improvement to the uni-gram model in where the uni-gram gives no consideration to any context to the word that occurs behind, the probability distribution is now drawn from the occurrence of a phrase of two words in the collection.
- N-gram An extension of the bi-gram model now extrapolated to N words before the word.
- Class LM Groups word by classes to generate a language model, for example thursday and friday would be grouped together.
- Topic LM Groups words by classes to generate a language model

Recurrent Neural Network Language Models

Neural network are the basis behind machine learning methods. They consist of an individual neuron which assign weights to the inputs to generate a predicted output. A typical neural network consists of many neurons that are connected in a feedforward manner within many layers. Refer to figure 1 for a representation of this.

Training a neural network requires labelled data in where the model algorithmic learns the weights associated with the training set and adjust the weights per gradient descent. This enables for a tuned model to the data set.

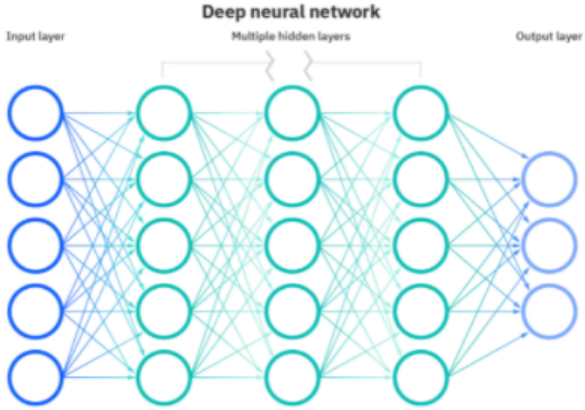

Figure 1: Neural Network Architecture [2]

The important aspect of a recurrent neural network is that the state of the previous layer is always passed as well to the next layer. This creates a memory component in where the layers are connected in such a way that enables for context to be represented within recurrent language models. Refer to figure 2 for the representation of this model. This is similar to the bi-gram model/N-gram model in where there is now context being kept track for the words but being trained through machine learning tactics

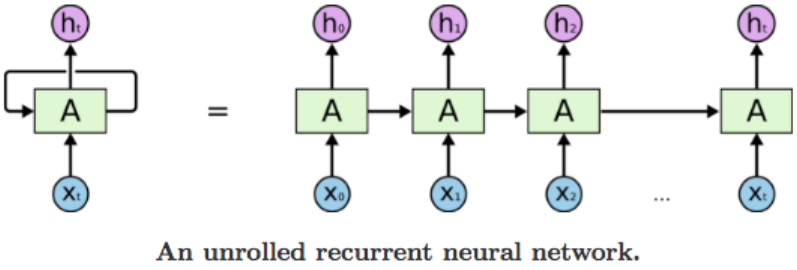

Figure 2: Recurrent Neural Network [3]

Benefits and Drawbacks

The benefits of using a RNNLM instead of a statistical model is that they are much better at predicting the next word as an incremental model. Around 50% reduction of perplexity was shown by Karfiat and Al[4] when compared to traditional statistical n-gram models. There are a lot of empirical evidence that RNNLMs just capture more complexity when it comes to the size of the sentence and words. In addition to outperforming them in context analysis , they are also able to capture longer sentences and context understanding for them. [1].

It is noted that the major drawback of RNNs is that they are still a neural network. And like any neural network, passing forward and training a neural network requires massive more amount of work as opposed to just generating a simple collection probability distribution from a bunch of words. As such even passing them through will be slower than the statistical language models.[1]

Conclusion

In the end, both methods of generating language models through both means are viable methods. Whether it is drawing probabilistic distributions of words or generating neural network models, they both

produce accurate results. The question that is the most important is the trade off between complexity and speed. Thus it would be a case-by-case basis in where one method of generating the language model might be better than the other.

References

[1] J. Dehdari, "A short overview of statistical language models." [Online]. Available: $https://jon.dehdari.org/tutorials/lm_overview.pdf$

[2] I. C. Education, "What are neural networks?" [Online]. Available: https://www.ibm.com/cloud/learn/neural-networks

[3] A. Mittal, "Understanding rnn and lstm," Aug 2021. [Online]. Available: https://aditi-mittal.medium.com/understanding-rnn-and-lstm-f7cdf6dfc14e

[4] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," vol. 2, 01 2010, pp. 1045–1048.