Capstone Proposal

Machine Learning Engineer Nanodegree
Brian Burns
December 16, 2016

Word Prediction using Recurrent Neural Networks

Domain Background

Word prediction is the task of predicting the most likely words following the preceding text. This has many applications, such as suggesting the next word as text is entered, as an aid in speech and handwriting recognition, or generating text to help fight spam.

The generation of a likely word given prior words goes back to Claude Shannon's paper on Information Theory (Shannon 1948) which was based on Markov models introduced by Andrei Markov (Markov 1913). These n-grams formed the basis of commercial word prediction software in the 1980's, eventually supplemented with similar syntax and part of speech predictions (Carlberger 1997). More recently, distributed representations of words have been used in recurrent neural networks (RNNs), which can better handle data sparsity and allow for more of the context to affect the prediction (Bengio 2003).

My interest in the task stems from the desire to learn more about machine learning with Natural Language Processing (NLP), Markov models, the Python Natural Language Toolkit (NLTK), distributed word representations, word2vec, RNNs, LSTM, and Deep Learning with Keras and TensorFlow.

Problem Statement

Problem: Given a sequence of n words, what are the k most likely words that might follow?

For instance, given the sequence "Alice went to the", the program should return a list of words like "forest", "library", and "mountains", depending on the text the program was trained on.

Datasets and Inputs

Any text can be used, so long as it has words separated by spaces. For this project texts from the Gutenberg project will be used, as an arbitrary number of words can be easily obtained without copyright or licensing issues (texts before 1922 are out of copyright). More modern corpora could also be used, such as Twitter posts or chat logs, if the purpose were to predict more contemporary writing.

The models will be trained with up to approximately a million words, using the following texts:

Author	Year	Title	Gutenberg #	Words
George MacDonald	1858	Phantastes	325	72,334
Victor Hugo	1862	Les Miserables	135	568,690

Author	Year	Title	Gutenberg #	Words
Lewis Carroll	1865	Alice in Wonderland	28885	30,355
Robert Louis Stevenson	1883	Treasure Island	120	71,611
Henry James	1898	The Turn of the Screw	209	45,294
M R James	1905	Ghost Stories of an Antiquary	8486	47,891
Arthur Machen	1907	The Hill of Dreams	13969	68,849
Kenneth Graham	1908	The Wind in the Willows	289	61,461
P G Woodhouse	1919	My Man Jeeves	8164	53,858
M R James	1920	A Thin Ghost and Others	20387	34,293
Total				1,054,636

The texts can be found at gutenberg.org, for example http://www.gutenberg.org/etext/28885.

Solution Statement

A Recurrent Neural Network (RNN) will be used to predict the next word in a sequence. In such networks, a sequence of words is encoded as a set of word vectors in a high-dimensional space (e.g. 300) – these will be obtained from the precalculated word2vec vectors (Mikolov 2013). Then the network will be trained until the output is within a certain distance of the actual word – on testing, a sequence of words will be fed into the network and the output used to search the vector space for the closest k words.

Benchmark Model

For the benchmark model a simple n-gram model will be used – this is a standard approach for next word prediction. A nested dictionary is created based on the training data, which counts occurrences of n-tuples of words. These are then normalized to get a probability distribution, which can be used to predict the most likely words following a sequence.

Evaluation Metrics

For evaluation of both approaches, the accuracy score will be reported, for increasing training set sizes. The dataset will be split into training and test sets – after training, a sequence of words from the test set will be chosen at random, then fed to the predictor, and the most likely k words compared to the actual following word.

Accuracy: # correct predictions / # total predictions

A prediction will be considered *correct* if the actual word is in the list of k most likely words - this is relevant to the task of presenting the user with a list of most likely next words as they are entering text.

Project Design

The texts will first be preprocessed to remove punctuation and converted to lowercase.

For the training step, the baseline trigram predictor will be fed all word triples, which will be accumulated in the nested dictionaries and converted to probabilities. For the RNN predictor, all word sequences or skip-grams will be fed to the network and trained until its output is close to the correct final word.

For the testing step, the baseline predictor will be fed random tuples of words, and the top k predicted words will be compared against the actual word, and an accuracy score tallied. For the RNN predictor, the same process will be used.

Training sets of increasing sizes will be used - 1k, 10k, 100k, 1 million words, and the results recorded for comparison. Timing and memory information will also be recorded for all processes for later analysis.

References

Bengio, Yoshua, et al. "A neural probabilistic language model." Journal of Machine Learning Research. (2003).

Carlberger, Alice, et al. "Profet, a new generation of word prediction: An evaluation study." Proceedings, ACL Workshop on Natural Language Processing for Communication Aids (1997).

Markov, Andrei, "An example of statistical investigation of the text Eugene Onegin concerning the connection of samples in chains." Bulletin of the Imperial Academy of Sciences of St. Petersburg, Vol 7 No 3. (1913). English translation by Nitussov, Alexander et al., Science in Context, Vol 19 No 4. 2006

Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).

Project Gutenberg. (n.d.). Retrieved December 16, 2016, from www.gutenberg.org.

Shannon, Claude, "A Mathematical Theory of Communication." The Bell System Technical Journal, Vol. 27. (1948).