Word Embeddings for Next Word Prediction

521 Project - Spring 2022

Benjamin Houghton, Chevy Robertson

GEORGETOWN UNIVERSITY

Introduction & Inspiration

The task Next Word Prediction (NWP) is observed quite often in our daily lives, when performing a Google search, typing a text message and even in Word processing software. One of the most fundamental steps in performing NWP is transforming the sentence or sequence into some numerical form that the computer can understand. This can be something as simple as a One Hot Encoded Vector (OHE) or as complex as Word Embeddings which were trained using a Neural Network Architecture.

Some of the notable inspiration for this project is the research referenced in the References section. The authors of the first reference trained LSTM models on 40-character-length sequences and tried to predict the next N characters. They were able to achieve 56% accuracy on the testing data. Similarly, the author of the third reference trained an LSTM model for the task of text generation. In addition to word embeddings and OHE vectors, he used vectors that described the context of the input data. The author reported that incorporating context as a feature led to better results.

Penn Treebank Dataset & Embeddings

The Penn Treebank (PTB) is a corpus that derives from a multitude of sources, including archives of stories produced by the Wall Street Journal, abstracts from government departments, radio transcripts, and many others. PTB is used for various tasks in NLP, but it is most commonly used for part-of-speech tagging. In its entirety, PTB consists of six text files and a README file. Half of the text files contain information at the word-level, and the other half contain information at the character-level. However, for the purposes of this project, only the word-level version of PTB was used. In its entirety, the latter version consists of 49,199 sentences that were preprocessed to exclude capitalization, numbers, and punctuation. Of these 49,199 data samples, 42,068 are included in the training text file, 3,370 are included in the development text file, and the remaining 3,761 sentences form the testing text file. The combination of all three partitions results in a vocabulary size of 10,000 words.

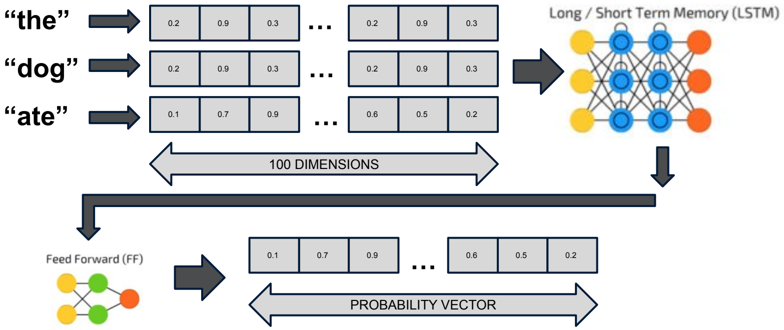
Word2Vec Embeddings: In the context of NLP, "Word2Vec" refers to a model that converts each word in a corpus to a vector. In this project, such a model from Gensim, an NLP library, was trained on the vocabulary of PTB. This resulted in the creation of a Word2Vec embedding space that consisted of each word represented as a dense, 100-dimensional vector.

Trained Embeddings: The trained embedding space also consisted of each word represented as a dense, 100-dimensional vector. However, unlike Word2Vec, the space was created using a Keras Embedding layer that updated the embedding weights during training.

OHE Vector Inputs: The OHE vector inputs consisted of each word represented as a sparse, one-hot-encoded vector with a dimensionality equal to 10,000, or, the vocabulary size of PTB.

Methodology

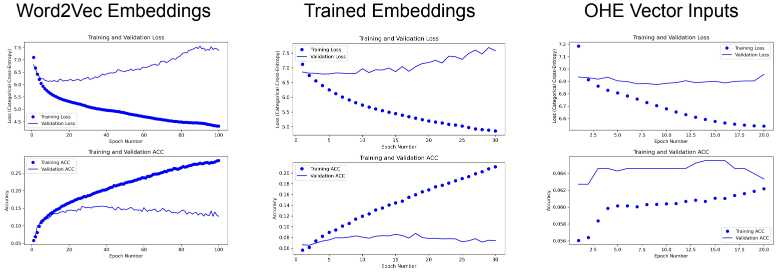
Each of the 3 models used a different input or embedding space. Two of the models used a 100 dimensional word embedding space, one of which is the Word2Vec pre-learned embedding space.



Output has the same dimensionality as vocabulary, where each element in the probability vector represents each word's probability of being the next word in the sentence

Training Results

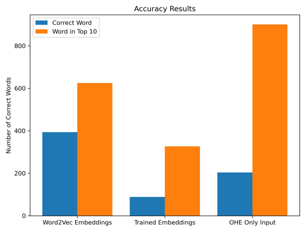
From the training results it is observed that the LSTM models which used Word2Vec embeddings were able to generalize better than the other models. The other models quickly overfit to the training data even with dropout and regularization. Another advantage of the Word2Vec model is the training speed since the embedding weights were frozen.



Results & Conclusions

Model	Correct NWP	NWP Accuraccy	Prediction in Top 10	Top 10 Accuracy
Word2Vec	394	0.111	625	0.177
Trained Embeddings	89	0.025	327	0.093
OHE Vector Input	204	0.058	901	0.255

On the right hand side the accuracy of all 3 models are shown. It is noted that the model using Word2Vec embeddings had the highest number of correct prediction. However, the model which used OHE vectors as inputs most often predicted a word that was similar to the target word.



To further investigate the output of the models, specially the model which used Word2Vec embeddings and the OHE input model several made up sentences were created and the models were used to predict the next word. As long as in the input sentence vocabulary exists within the corpus any sentence can be input into any of the 3 models for next word prediction.

Fabricated Sentences:	**Word2Vec Prediction:**	**OHE Prediction:**
i really love a baseball	**and**	**the**
what time are you going	**to**	**the**

Not surprisingly is seems that the models tend to predict very common words. Another major finding of this experiment was that pre-trained embedding spaces such as Word2Vec and others like it can be leveraged to speed up the training time of a model while delivering accurate results.

References

Ambulgekar, Sourabh, et al. "Next Words Prediction Using Recurrent NeuralNetworks." ITM Web of Conferences. Vol. 40. EDP Sciences, 2021

Sundermeyer, Martin, Ralf Schlüter, and Hermann Ney. "LSTM neural networks for language modeling." Thirteenth annual conference of the international speech communication association. 2012

Santhanam, S. "Context based Text-generation using LSTM networks. arXiv 2020." arXiv preprint cs.CL/2005.00048.

Ghosh, Shalini, et al. "Contextual lstm (clstm) models for large scale nlp tasks." arXiv preprint arXiv:1602.06291 (2016).