DeepGuidance: Using Neural Networks to Suggest Areas of Study

Elena Montes, *elenamontes@ou.edu* | Dr. Christan Grant, *cgrant@ou.edu*

The University of Oklahoma, Norman, OK

INTRODUCTION

Many university students enter as passionate students with undecided majors. As they matriculate, many will switch their major at least once before graduation.

To guide the students' selection of a major, we use a neural network model trained on a corpus with 3 billion running words to suggest an area of study (Google, 2013). The model takes as input natural language terms representing a students' interests or skills and gives as output the net's prediction for the most relevant area of study. After training, the top performing model predicts correct results with between 60% and 90% accuracy.

Since there were some anomalies in training, future work involves smoothing the model to create more predictable accuracy and using a different data set for more accuracy. The high ambiguity, subtlety, and variability embedded in the structure of human language makes deciphering natural language difficult for a computer (Goldberg, 2016). Thus, using a neural network model with distributed vector representations lets the computer help us to uncover hidden connections between related terms. This can assist in making better suggestions than a human may, given seemingly unrelated interests.

STAGE 1: METHODOLOGY

During the first phase, we constructed 12 different net architectures to compare on three variables, in order to choose the one with the best performance:

(1) activation function used in hidden layers (from a choice of three)
(2) depth of the neural net (2 or 5 layers)
(3) inclusion of a dropout layer to set a random half of inputs to zero for each sample

The table below shows the varying properties for each neural net.

Name	Activation	Deep	Dropout
NN01	ReLU	No	No
NN02	ReLU	No	Yes
NN03	ReLU	Yes	No
NN04	ReLU	Yes	Yes
NN05	tanh	No	No
NN06	tanh	No	Yes
NN07	tanh	Yes	No
NN08	tanh	Yes	Yes
NN09	sigmoid	No	No
NN10	sigmoid	No	Yes
NN11	sigmoid	Yes	No
NN12	sigmoid	Yes	Yes

Other activation functions exist; we chose these due to historical context and popularity.

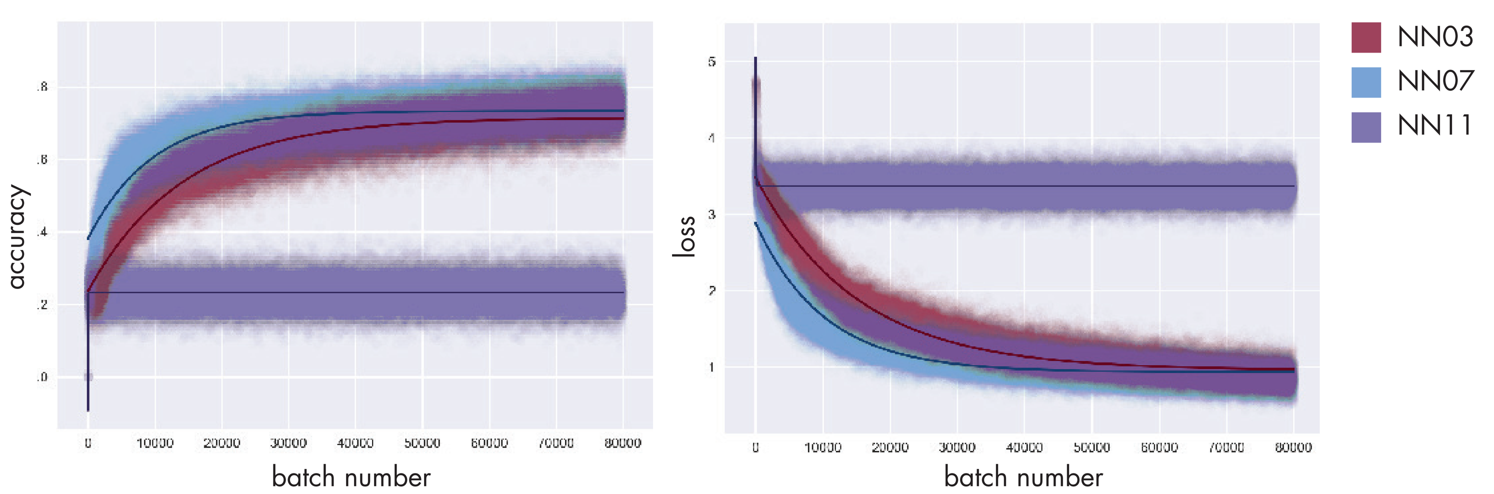
The experiment was conducted in **Python** using the **Keras** library for neural network construction and training, the **gensim** library and **Google News** pre-trained word embeddings for NLP samples, and the **pandas** and **Matplotlib** libraries for data collection and graphing.

STAGE 1: RESULTS

We ran all 12 nets on 8 million samples, processed in batch sizes of 100.
NN07, the tanh-activated, deep net with no dropout, consistently performed the best.
NN11, the sigmoid-activated, deep net with no dropout, consistently performed the worst.

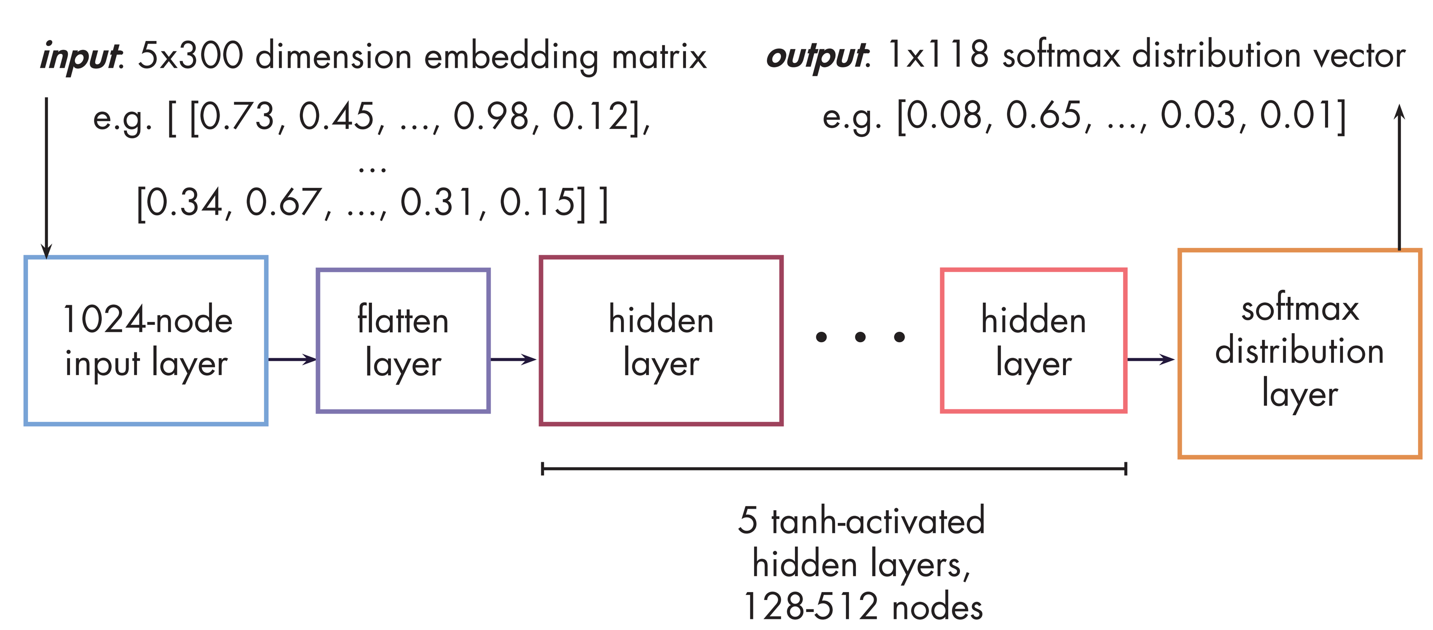
For all but two nets tested, accuracy and loss both consistently improved over time. However, though the general trend showed learning progress, the different network architectures also showed consistent oscillation which did not decrease over time.

The graph below shows the accuracy and loss through training of a selection of three nets (one for each activation function, as these groups performed similarly).



The sigmoid-activated nets in particular did not show any improvement over time.

The figure below shows the architecture of the neural network used in the area of study suggestion task. The final layer uses a softmax activation function to generate a normalized distribution over the 118 areas of study selected for the task.



STAGE 2: METHODOLOGY

During the second phase, we observed labeling trends over different output classes (majors). Classification was based on cosine distance between a random sample word vector and the vectors representing the different majors.

Word embeddings were extracted from Google News vectors (Mikolov et al., 2013).

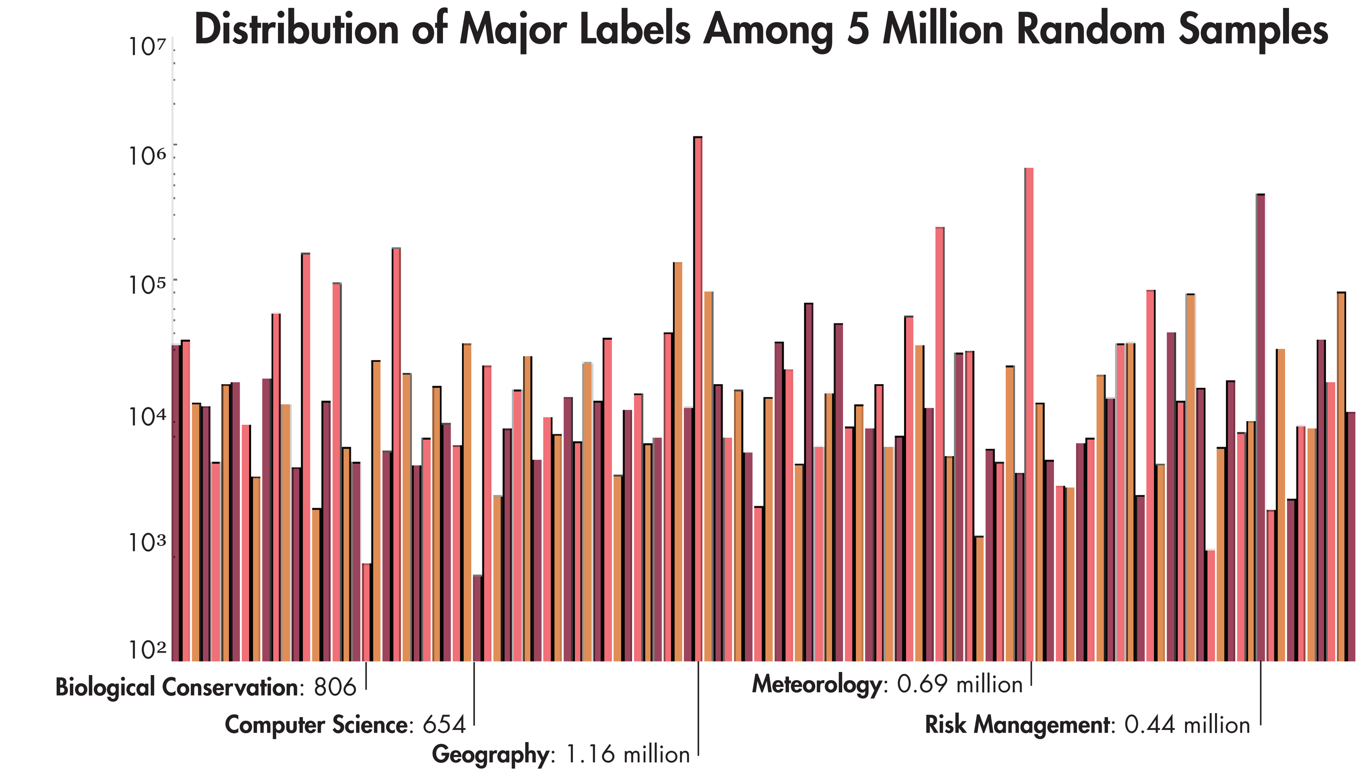
We chose 5 million 5-word samples randomly, ran them through our trained NN07 model and recorded the resultant classifications.

STAGE 2: RESULTS

Using cosine distance to label yielded unexpected results. With a run of 5 million random samples over 118 output classes, an even distribution of predictions would expect around 40,000 samples to be assigned each label. However, there were some very clear outliers.

Using the Google News vectors appears to have resulted in skewed classification of samples towards majors related to words used in news articles.

Distribution of Major Labels Among 5 Million Random Samples



CONCLUSIONS

This experiment aimed to compare perfomances of differently constructed neural networks, as well as compare classifications of pre-trained word embeddings.

We found that choosing a particular activation function can inhibit learning entirely for a neural network, and performance is extremely dependent on the word embeddings used.

In future, we can train a new word embedding model on corpora from relevant websites to improve performance and collect samples with words scraped from professors' and students' websites, labeling the samples with each person's area of specialization (Papoutsaki et al., 2015). This should result in a logical classification distribution and more accurate results.

REFERENCES

Alexandra Papoutsaki, Hua Guo, Danae Metaxa-Kakavouli, Connor Gramazio, Jeff Rasley, Wenting Xie, Guan Wang, and Jeff Huang. 2015. Crowd-sourcing from scratch: A pragmatic experiment in data collection by novice requesters. In *Third AAAI Conference on Human Computation and Crowd-sourcing.*

Google. 2013. Google Code Archive: word2vec. https://code.google.com/archive/p/word2vec/

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*

Yoav Goldberg. 2016. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research* 57:345–420.