Deep Learning: Assignment 2

Using Word2Vec and implementation of Feed-forward Neural Networks

This assignment involves the following tasks:

- Use Word2Vec
- Perform multi-label classification using Neural network

Submit the executed code in Jupyter notebook. You can write your observations and results using the heading and markdown cells in Jupyter. If you have memory or GPU constraints, you can use Google colab or Kaggle. The links to the libraries required are added in Moodle. You might need to install the following libraries:

- Tensorflow or Keras
- NLTK
- nltk-data
- gensim
- TSNE

If you are using Colab or Kaggle, some of these packages are not pre-installed, so you might need to install them yourselves.

1. Text classification, Word2Vec (Use scikit-learn)

   (a) i. 20-newgroup dataset is a collection of newsgroups in 20 topics. Fetch 20-newsgroup dataset.
   ii. Pre-process the dataset: Convert to lowercase, remove punctuations, symbols, and stopwords. You can use NLTK or any other library of your choice.
   iii. Convert the words in the dataset to vectors of dimension 100 using Word2Vec. Ignore words whose frequency is less than 10.
   iv. Find the vocabulary size.
   v. Find the most similar words in the corpus to the word "car" along with their similarities.
   vi. Find the top 5 words similar to the following operations:
   - girl + father - boy
   - sports - bat + ball

   vii. Create a TSNE plot for the top 20 words similar to each of the words ['baseball', 'software', 'police', 'government', 'circuit', 'car'] as shown in Figure 1.
   viii. The dataset consists of documents. Each document is a datapoint. Formulate a methodology to represent each document as a vector using the word vectors. Mention the method employed to create the vector representation of the documents.

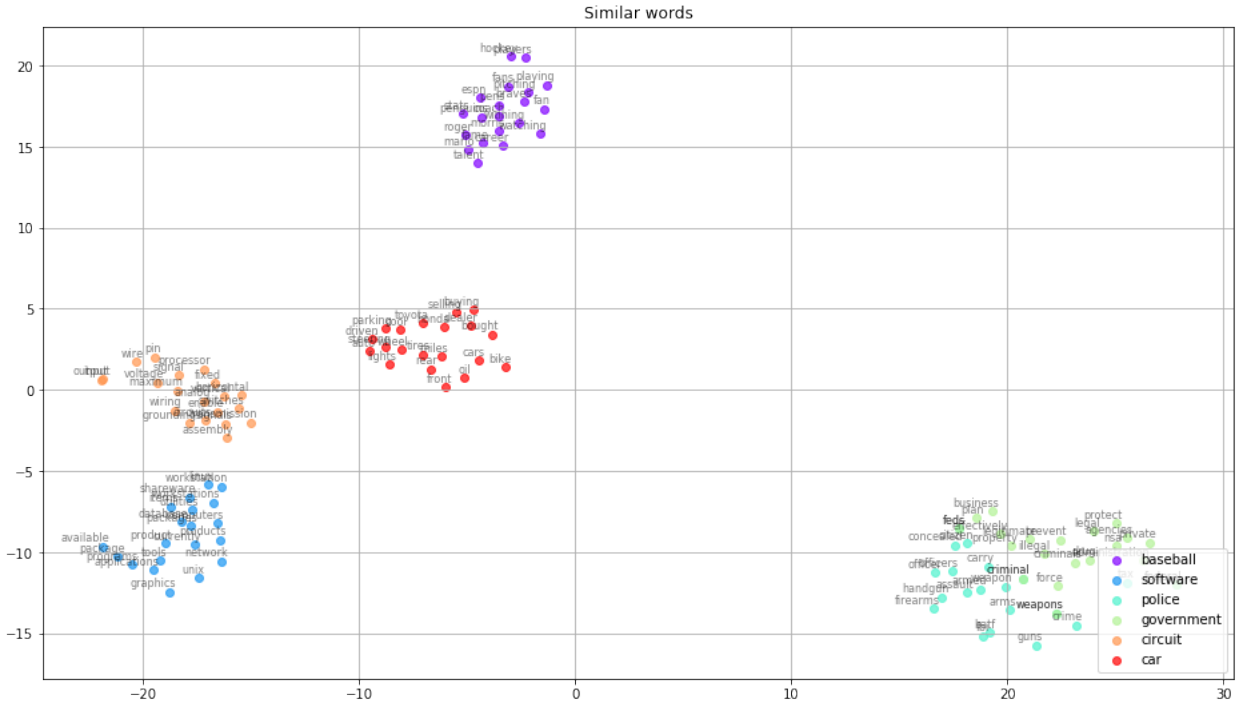

Figure 1: "TSNE visualization of similar words"

(b) i. Split the dataset into training (70%), validation(10%) and testing(20%) data.

ii. Plot the loss vs iteration curve, classification error vs iteration curve, classification accuracy vs iteration curve for training data and report your observations.

iii. Find the classification accuracy, the number of true positives, true negatives, false positives and false negatives for both training and test data.

iv. There are two training algorithms for Word2Vec: skip-gram and bag of words. Which training algorithm is performing better in this data set?

The one student whose accuracy is higher than everyone will get a bonus mark of 1 in the the total score of deep learning course.

2. MNIST digit classification: (Use Tensorflow or Keras).

(a) MNIST is a database of hand written images. Download MNIST data using the built-in functions in Tensorflow or Keras

(b) Get the training, validation and test data sets using the functions in Tensorflow or Keras. If you are using Tensorflow, the dataset is already split into training set of size 55000, validation set of size 5000, and test set of size 10000. If you are using Keras, the data set is split into training set of size 60000, and validation set of size 10000. Then create a validation set of size 5000 from the training set.

(c) Classify the dataset using a feed-forward neural network. Vary the hyperparameters as follows:

i. Create a fully connected feed forward neural network for MNIST classification with one hidden layer(32 nodes). Train the model using Stochastic Gradient Descent optimizer with learning rate 0.1. Use Sigmoid activation function in the hidden layer.

ii. Normalize the dataset to range (0,1). Compare both the normalized and unnormalized models in terms of training time and accuracy.

iii. Choose the best performing model among (i) and (ii). Train different models by varying the number of hidden layers in the model as 2 and 3. Record the observations. Other hyperparameters are same as in (i).

iv. Choose the best performing model in (iii). Train models by varying the learning rates as 0.001 and 0.0001 and record your observations.

v. Choose the best performing model in (iv). Train models by varying the number of nodes in each hidden layer to 64 and 128.

vi. Choose the best performing model in (v). Train models by varying the activation functions in each of the hidden layers to tanh, relu and leaky relu and record your observations.

vii. Among all the configurations of hyper-parameters that you trained above, which setting is best. How did you decide which setting is better?

viii. Among all the models trained above, how will you choose the best model? Which is the best model?

ix. (Optional) Report the training time and RAM usage for each training.