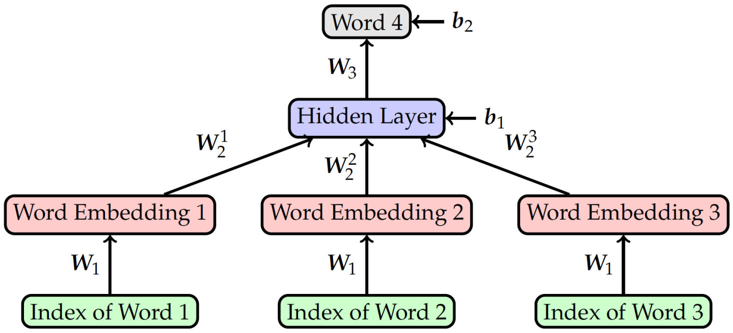
CMPE 597 Sp. Tp. Deep Learning
Spring 2021 Project I
Due: April 30 by 11.59pm

Network

In this project, you will train a neural language model using a multi-layer perceptron given below. This network receives 3 consecutive words as the input and aims to predict the next word. You will train this model using cross-entropy loss function, which corresponds to maximizing the probability of the target word.



The network consists of a 16 dimensional embedding layer, a 128 dimensional hidden layer and one output layer. The input consists of a sequence of 3 consecutive words, provided as integer valued indices representing a word in our 250-word dictionary. You need to convert each word to it's one-hot representation and feed it to the embedding layer which will be 250×16 dimensional. Hidden layer will have sigmoid activation function and the output layer is a softmax over the 250 words in our dictionary. After the embedding layer, 3 word embeddings are concatenated and fed to the hidden layer.

Data

Train/Validation/Test splits are provided to you in the project folder. Words in the dictionary can be found in `vocab.npy` file.

Task

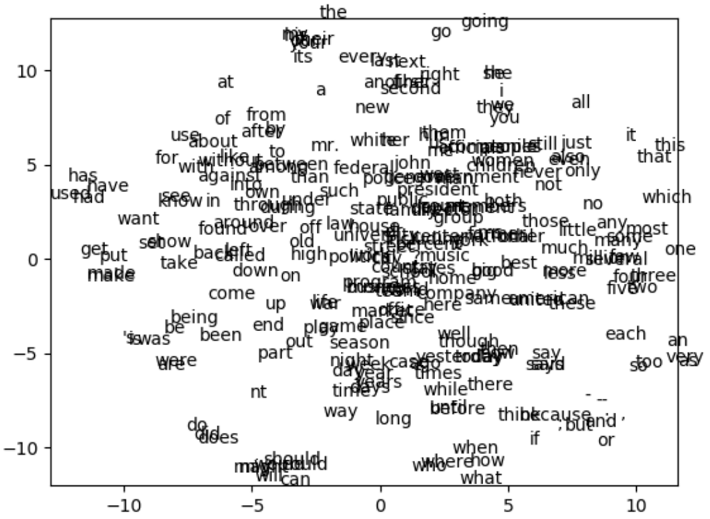
1. (15 pts) Write the mathematical expressions for forward propagation.
2. (15 pts) Write the mathematical expressions of the gradients that you need to compute in backward propagation.
3. (20 pts) Implement a Network class, `Network.py`, where you have the forward, backward propagation, and the activation functions. Use matrix-vector operations.

4. (20 pts) Implement a main function, `main.py`, where you load the dataset, shuffle the training data and divide it into mini-batches, write the loop for the epoch and iterations, and evaluate the model on validation set during training. Report the training and validation accuracy.

5. (5 pts) Implement an evaluation function, `eval.py`, where you load the learned network parameters and evaluate the model on test data. Report the test accuracy.

6. After obtaining 16 dimensional embeddings

   (a) (10 pts) Create a 2-D plot of the embeddings using t-SNE which maps nearby 16 dimensional embeddings close to each other in the 2-D space. You can use of the shelf t-SNE functions. Implement a `tsne.py` file where you load model parameters, return the learned embeddings, and plot t-SNE. Use the words in the `vocab.txt` as the labels in the plot. Your plot should like the plot below.



   (b) (5 pts) Look at the plot and find a few clusters of related words. What do the words in each cluster have in common?

   (c) (5 pts) Pick the following data points; 'city of new', 'life in the', 'he is the'. Use the model to predict the next word. Does the model give sensible predictions?

7. (5 pts) Provide a README file where I can find the steps to train, load, and evaluate your model.

Submission

You need to submit a zip file with the name `NameSurname_Project1.zip` containing the files below.

- A pdf report including your name, your student number, your answers to questions, and references.

- Network.py
- main.py
- eval.py
- tsne.py
- Trained model model.pk
- README.txt

IMPORTANT NOTE: You are NOT allowed to use any deep learning libraries such as tensorflow, and pytorch. You can use Python libraries and functions, such as numpy, matplotlib. Do not forget to cite your references and resources.