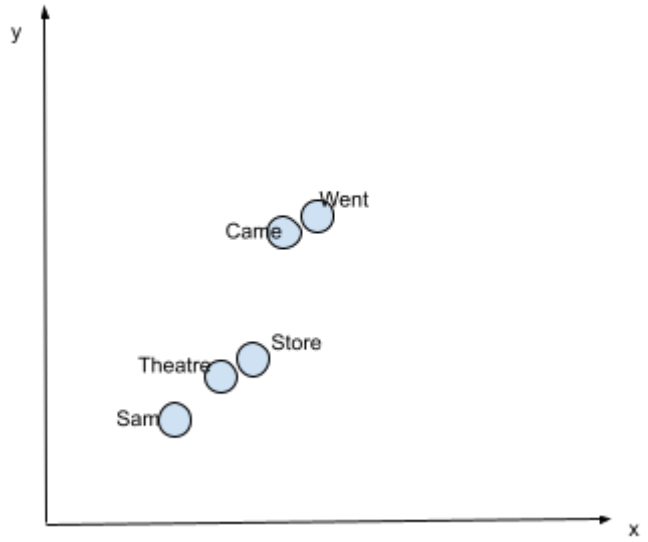
COSC440 assignment4 RNN conceptual questions

1. What are the dimensions of an embedding matrix? What do they represent?
An embedding matrix is a mapping of discrete variables to their continuous vector representations, where each dimensional vector represents meaningful relationships with other vectors in the space. For a word embedding matrix, it represents words in a coordinate system where related words, based on a corpus of relationships, are placed closer together. Each row in the word embedding matrix represents a word vector and each column corresponds to a dimension. Each row of the dimensional values reflects the position of a word within the vector space and is learned based on the words that surround the word when it is used.

2. Given the following 3 sentences, plot in 2d (or create an adjacency matrix) reasonable embeddings for Sam, Came, Theater, Store, and Went. (Hint: A simple graph with some clusters is fine and you don't need to specifically follow a CBOW or other algorithm to generate one). See lecture notes or https://medium.com/@Petuum/embeddings-a-matrix-of-meaning-4de877c9aa27 for an overview.
Sam went to the store.
Then Sam came to the theater.
I went to the theater.



3. What are the limitations of feed forward networks for language modelling that RNNS solve? (3-6 sentences)
In Language modeling, knowing the relationships between previous words helps to predict the next word better in a sequence. In a feed forward network, information travels one way from inputs to outputs and there are no feedbacks, resulting the network nodes unaware of previous relationships and historic information. While RNN resolves this by feeding previous state back into the current input using a loop and hence predicts the next word better in a sequence. In addition, RNN is like a layered feedforward where the model size of an RNN doesn't increase for longer input and same weights are applied on every timestep during training. This is not the case with feedforward which has to deal with a fixed length input and output. Hence an RNN is also more flexible in this regard.

4. Explain in your own words how your RNN in this assignment predicts the next word in the sequence.
In RNN_Part1, the network first receives sequences of words input of a certain window size and encodes all the words into an embedding matrix in an embedding layer. This word embedding matrix maps the words represented by numeric indexes to their dense vector representations. It is a representation of text where words that have the same meaning have a similar representation.

Then in the LSTM layer, the deep network decides what to keep in and what to eliminate from the memory, combining the previous state feedback, the current memory, and the current input into consideration and covert the sequences of embedding vectors to a compressed representation. This process also efficiently solves the vanishing gradient problem. The compressed representation effectively captures historic information in the sequence of words in the text.

Then the fully connected layer with softmax activation takes the deep representation from the LSTM layer and transforms it into the final output class scores, which predicts the likelihood of each word in the sequence.

After the forward pass above, the loss is computed by comparing the true labels with our predictions. Gradients are then computed using backpropagation in order to minimise the loss and training weights are updated based on the gradients.

During each training iteration, the model predicts the next word with the highest likelihood for a given sequence.

5. What are the limitations of RNNS that Transformers solve? (3-6 sentences)
Transformer solves RNN's' limitations of sequential processing that causes long-term memory dependencies and restricts parallel computation. Unlike RNN, Transformer does not necessarily process the sequential input data in order and can process the sentence as a whole. It uses positional embedding to encode the positions of words and Attention mechanism to compute similarity scores between words in a sentence. Such mechanism provides information about the relationships between different words for any position in the input sequence, avoids recursion and allows for more parallelization than RNN and therefore reduces training times.

6. Explain in your own words how your RNN encoder and RNN decoder work together to translate from French to English in this assignment.
The Encoder LSTM layer takes the embedded representation of French sentence input over time and tries to encapsulate all its information and store it in its final internal states (hidden state and cell state). The internal states are then passed onto the decoder part. The outputs at each time-step of the encoder LSTM layer are discarded.

The Decoder LSTM layer receives the French hidden states from the encoder and the embedded representation of the true target English sentence, trying to encapsulate all the relationships within and between the sentences of the two languages to output the next predicted English word.

The output predictions is the probability distribution over the entire vocabulary in the output dataset which is generated using the Softmax activation function. The word with the maximum probability is chosen to be the predicted word.

As we are using Teacher Forcing to train our decoder, at each timestep, the true target English word (not predicted output) from the previous timestep is fed into the current timestep to make predictions. This improves model stability and training performance.

Finally, the loss is calculated on the predicted outputs from each timestep and the errors are backpropagated through time to update the training parameters of the model. The final states of the decoder are discarded.

Student ID: 97245310
Name: Yuezhang Zhu