**Deep Learning - 236606 – HW4**

Ido Glanz - 302568936 Matan Weksler - 302955372

1. **IMDB Review generator**

In this assignment, we implemented a review generator based on the IMDB reviews data-set whereas the generated review can be conditioned on its sentiment (i.e. positive, negative or mixed).

We choose to run a word-based learning scheme after attempting to implement both methods and running into the ----

The general pipe-line of the task as we implemented it is as follows:

1. Load IMDB reviews
2. Generate sample sequences:
   1. Each review was broken down to sub-sentences with the consecutive word as a label, i.e. for the review “The movie was very good and I loved it” we would generate:
      1. The (with y label “movie”)
      2. The movie (with y label “was”)
      3. The movie was (with y label “very”)

And so on.

Each sample was also matched with a sentiment originated from the original review sentiment as an input.

* 1. From each review, we generated a random number of sub-samples ranging from 10 to 40 words.
  2. In the end, each learning sample was in the shape: [review seed, sentiment] and the label was [following word]

1. Defining and training the RNN model:
   1. The first part was defining the two inputs and combining them together. We choose to fully connect to each LSTM cell a weighted sentiment by adding a fully connected layer between the single sentiment input and each vector input to the LSTM.
   2. Next, we added 3 LSTM layers, each with a state vector of size 256 and with dropout.
   3. Finally, we added fully connected layers ending with a softmax layer with output dimension as the number of words we included (6K).
2. Review generation:
   1. At each iteration, the models input was the accumulated sentence along with a sentiment 0/1.
   2. The predicted outcome was one of the following:
      1. Argmax of the output vector (i.e. greedy selection), either ignoring Out-Of-Vocabulary words or not (flag-dependent)
      2. Soft-sample: sampling from the output probability vector (like rolling a dice biased with the prediction values n-times). Again, ignoring OOVs if wanted.
      3. Selecting the nth-best choice (e.g. the 3rd best guess).