<https://skymind.ai/wiki/neural-network>

Neural networks are a set of algorithms that are designed to recognize patterns. We will be using this in our project to detect a user’s mood, by recognizing patterns in their voice and tone.

<https://towardsdatascience.com/recurrent-neural-networks-by-example-in-python-ffd204f99470>

A recurrent neural network processes sequences one element at a time while retaining memory of what has come previously in the sequence. This memory allows the network to learn long-term dependencies in a sequence which means it can take the entire context into account when making a prediction (in our case, mood). A RNN is designed to mimic the human way of processing sequences: we consider the entire sentence when forming a response instead of words by themselves.

At the heart of an RNN is a layer made of memory cells. The Long Short-Term Memory (LSTM) maintains a cell state as well as a carry for ensuring that the signal (information in the form of a gradient) is not lost as the sequence is processed. At each time step the LSTM considers the current word, the carry, and the cell state. The LSTM has 3 different gates and weight vectors: there is a “forget” gate for discarding irrelevant information; an “input” gate for handling the current input, and an “output” gate for producing predictions at each time step.

There are several ways we can formulate the task of training an RNN to write text, in this case patent abstracts. However, we will choose to train it as a many-to-one sequence mapper. That is, we input a sequence of words and train the model to predict the very next word. The words will be mapped to integers and then to vectors using an embedding matrix (either pre-trained or trainable) before being passed into an LSTM layer.

<https://keras.io/>

Keras is a high-level neural networks API, written in Python.

Keras is an incredible library: it allows us to build state-of-the-art models in a few lines of understandable Python code. Although other neural network libraries may be faster or allow more flexibility, nothing can beat Keras for development time and ease-of-use.