LSTM (Long Short-Term Memory)

The **main limitation of RNNs** is that RNNs can't remember very long sequences and get into the problem of **vanishing gradient.**

What is the vanishing gradient problem?

The gradients of the loss function in neural networks approach zero when more layers with certain activation functions are added, making the network difficult to train.

Long Short-Term Memory (LSTM)

LSTMs come to the rescue to solve the vanishing gradient problem. It does so by ignoring (forgetting) useless data/information in the network. The LSTM will forget the data if there is no useful information from other inputs (prior sentence words). When new information comes, the network determines which information to be overlooked and which to be remembered.

LSTM Architecture

- In RNNs, we have a very simple structure with a single activation function (***tanh***)
- In LSTMs, instead of just a simple network with a single activation function, we have multiple components, giving power to the network to forget and remember information.
- **LSTMs have 4 different components, namely**
    - **Cell state (Memory cell)**
        - It is the first component of LSTM which runs through the entire LSTM unit. It kind of can be thought of as a conveyer belt.
        - This cell state is responsible for remembering and forgetting. This is based on the context of the input. This means that some of the previous information should be remembered while some of them should be forgotten and some of the new information should be added to the memory. The first operation (**X**) is the pointwise operation which is nothing but multiplying the cell state by an array of [-1, 0, 1]. The information multiplied by 0 will be forgotten by the LSTM. Another operation is (**+**) which is responsible to add some new information to the state.
    - **Forget gate**
        - The forget LSTM gate, as the name suggests, decides what information should be forgotten. A sigmoid layer is used to make this decision. This sigmoid layer is called the **"forget gate layer".**
        - It does a dot product of *h(t-1)* and *x(t)* and with the help of the sigmoid layer, outputs a number between 0 and 1 for each number in the cell state *C(t-1)*. If the output is a '1', it means we will keep it. A '0' means to forget it completely.
    - **Input gate**
        - **The input gate gives new information to the LSTM and decides if that new information is going to be stored in the cell state.**
        - **This has 3 parts:**
            - A ***sigmoid*** layer decides the values to be updated. This layer is called the "input gate layer"

        - A ***tanh*** activation function layer creates a vector of new candidate values, ***Č(t)***, that could be added to the state.

        - Then we combine these 2 outputs, ***i(t) * Č(t),*** and update the cell state.

    - The new cell state ***C(t)*** is obtained by adding the output from forget and input gates.

- **Output gate**
    - The output of the LSTM unit depends on the new cell state.
    - First, a sigmoid layer decides what parts of the cell state we’re going to output. Then, a *tanh* layer is used on the cell state to squash the values between -1 and 1, which is finally multiplied by the sigmoid gate output.