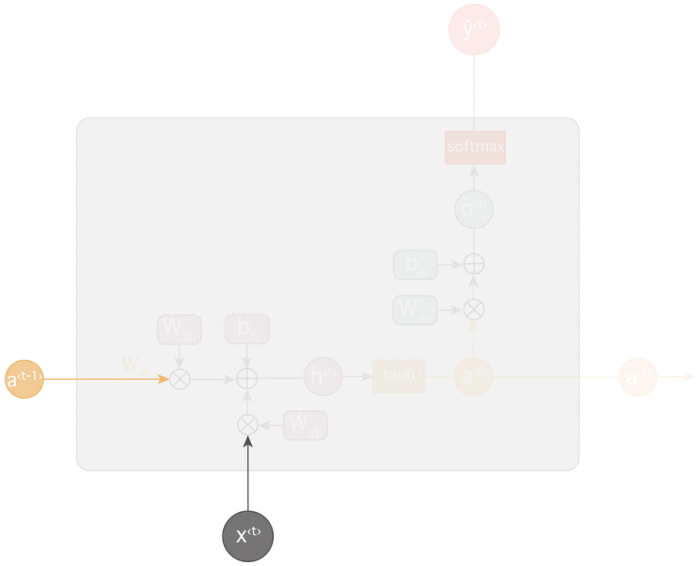
May 9 · 3 min read

RNN Training: Welcome to your Tape - Side A

Forward propagation in RNN

Forward propagation in single RNN Cell

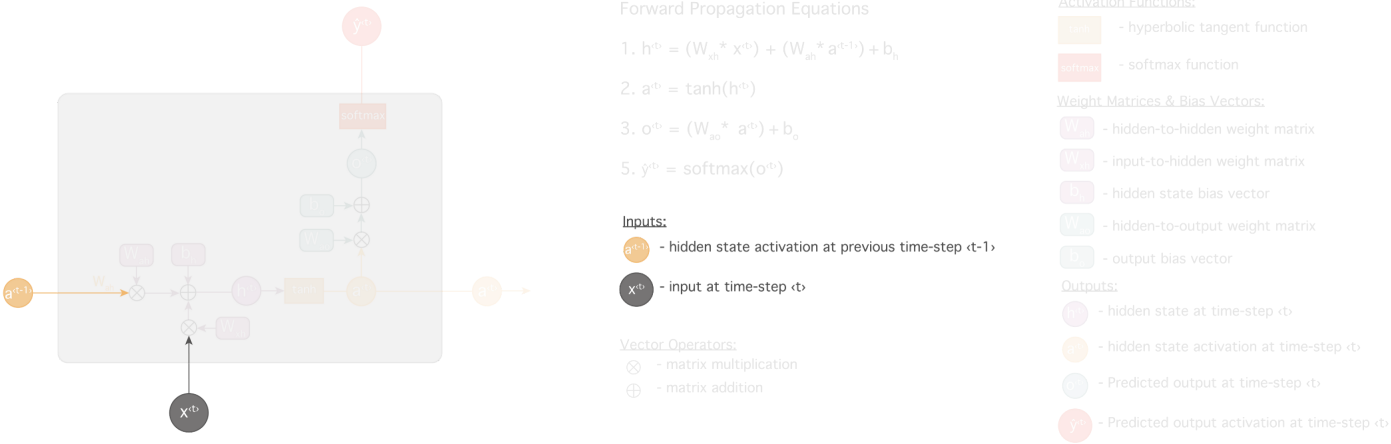
In a previous post, we looked at the basic RNN structure and its unrolling into repetitive chain of cells or time-steps, which we call RNN cells. Below, we would explore the internal structure and forward propagation computations that occur in a single time-step RNN cell.



RNN Cell

Lets break down what happens into individual steps:

Step 1: The cell takes in two inputs: `x(t)` and `a(t-1)`.



Step 2: Next, it calculates the matrix multiplication $\otimes$ between `W_xh` & `x⟨t⟩` and between `W_ah` & `a⟨t−1⟩` . Afterwards, it computes `h⟨t⟩` by adding the above together with the bias term `b_h` .

Step 3: Following the above, it calculates `a⟨t⟩` by passing `h⟨t⟩` through an activation function such as tanh or relu. In this case we use the `tanh` function.

Step 4: The cell outputs $a^{\langle t \rangle}$ and passes it on to the next time-step cell for computation.

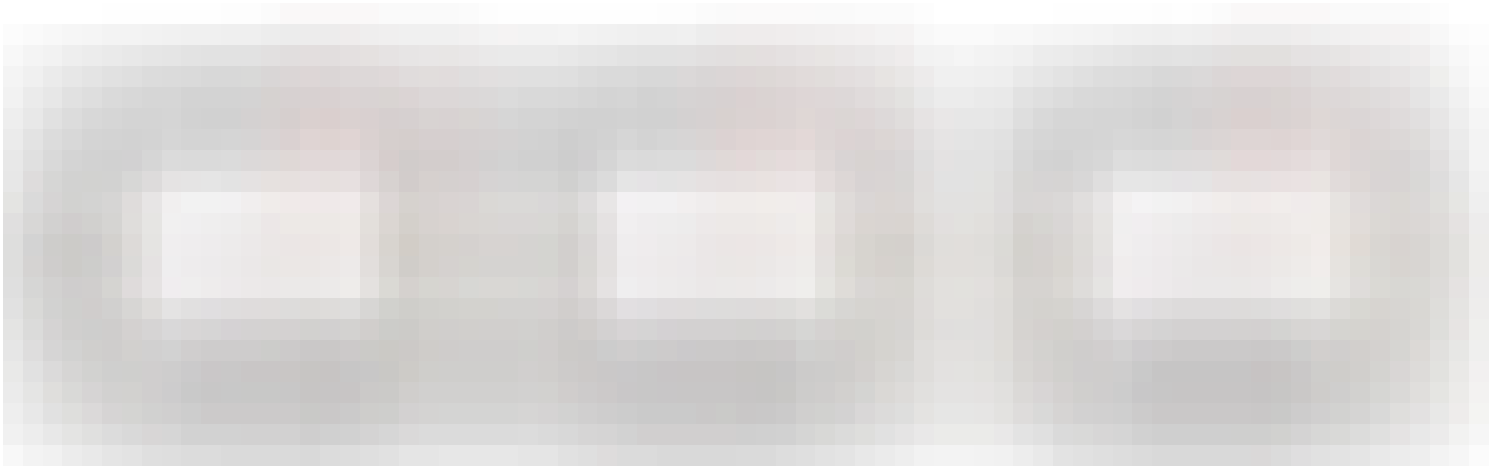
Step 5: Next, it computes $o^{\langle t \rangle}$; which is the unnormalized log probability of each possible value of the true output value. It does this, by calculating the matrix multiplication $\otimes$ between W_{ao} & $a^{\langle t \rangle}$ and adding it together with b_o .

Step 6: Finally, it obtains a vector of normalized probabilities `ŷ⟨t⟩` over the true output, by passing `o⟨t⟩` through an activation function such as sigmoid or softmax. The choice of output activation function usually depends on the expected output (sigmoid for binary class outputs, softmax for multi-class outputs).

Forward Propagation Through Time

Forward propagation through time is simply running these steps in the whole recurrent network rather than in just a single time-step RNN cell. It begins with the initialization of a hidden state `a⟨0⟩` , sharing of the weights and bias vectors `W_xh, W_ah, W_ao, b_h, b_o` across all time steps `t = 1 to T` , and repeating the steps above in each time step.

For example, if we have an 8 sequence input, `x⟨1⟩, x⟨2⟩,....,x⟨8⟩` then forward propagation for this network would be steps 1–6 repeated in a loop 8 times.



The above is the first step in training RNNs, in the second part of this post we would look at the back propagation through time (BPTT) algorithms and how to derive its gradients.