DERIVING BACKPROPAGATION EQUATIONS FOR AN LSTM

JUNE 19, 2019

In this post I will derive the backpropagation equations for a LSTM cell in vectorised form. It assumes basic knowledge of LSTMs and backpropagation, which you can refresh at Understanding LSTM Networks and A Quick Introduction to Backpropagation.

DERIVATIONS

FORWARD PROPAGATION

We will firstly remind ouselves of the forward propagation equations. The nomenclature followed is demonstrated in Figure 1. All equations correspond to one time step.

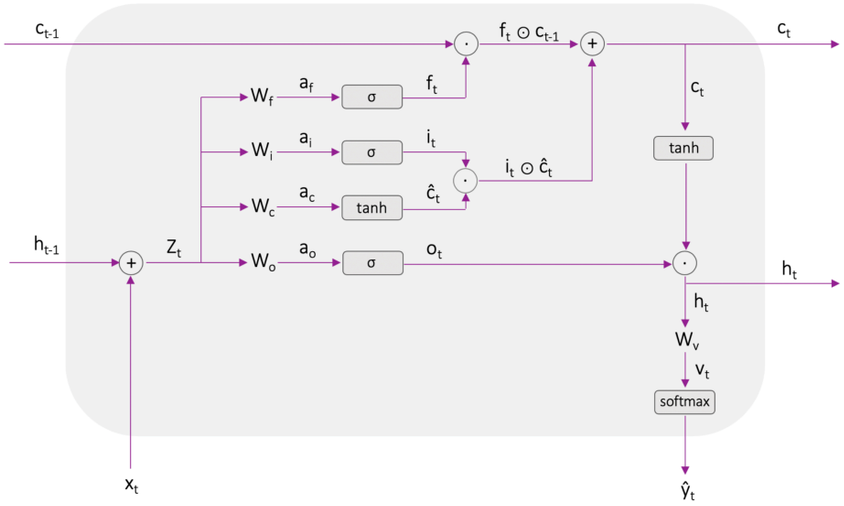

FIGURE 1: ARCHITECTURE OF A LSTM MEMORY CELL AT TIMESTEP T

$$h\_t - 1 \in \mathbb{R}^{n\_h}, \qquad x\_t \in \mathbb{R}^{n\_x}$$
$$z\_t = [h\_t - 1, x\_t]$$

$$a\_f = W\_f \cdot z\_t + b\_f, \qquad f\_t = \sigma(a\_f)$$
$$a\_i = W\_i \cdot z\_t + b\_i, \qquad i\_t = \sigma(a\_i)$$
$$a\_o = W\_o \cdot z\_t + b\_o, \qquad o\_t = \sigma(a\_o)$$
$$a\_c = W\_c \cdot z\_t + b\_c, \qquad \hat{c}\_t = tanh(a\_c)$$

$$c\_t = i\_t \odot \hat{c}\_t + f\_t \odot c\_t - 1$$
$$h\_t = o\_t \odot tanh(c\_t)$$

$$v\_t = W\_v \cdot h\_t + b\_v$$
$$\hat{y}\_t = softmax(v\_t)$$

BACKWARD PROPAGATION

Backpropagation through a LSTM is not as straightforward as through other common Deep Learning architectures, due to the special way its underlying layers interact. Nonetheless, the approach is largely the same; identifying dependencies and recursively applying the chain rule.

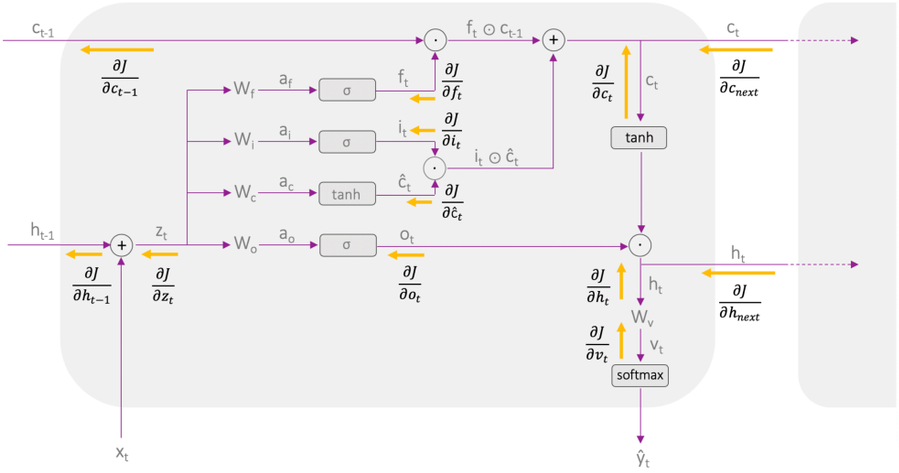

FIGURE 2: BACKPROPAGATION THROUGH A LSTM MEMORY CELL

Cross-entropy loss with a softmax function are used at the output layer. The standard definition of the derivative of the cross-entropy loss ($\frac{\partial J}{\partial v_t}$) is used directly; a detailed derivation can be found here.

OUTPUT

$$\frac{\partial J}{\partial v\_t} = \hat{y}\_t - y\_t$$
$$\frac{\partial J}{\partial W\_v} = \frac{\partial J}{\partial v\_t} \cdot \frac{\partial v\_t}{\partial W\_v} \Rightarrow \frac{\partial J}{\partial W\_v} = \frac{\partial J}{\partial v\_t} \cdot h\_t^T$$
$$\frac{\partial J}{\partial b\_v} = \frac{\partial J}{\partial v\_t} \cdot \frac{\partial v\_t}{\partial b\_v} \Rightarrow \frac{\partial J}{\partial b\_v} = \frac{\partial J}{\partial v\_t}$$

HIDDEN STATE

$$\frac{\partial J}{\partial h\_t} = \frac{\partial J}{\partial v\_t} \cdot \frac{\partial v\_t}{\partial h\_t} \Rightarrow \frac{\partial J}{\partial h\_t} = W\_v^T \cdot \frac{\partial J}{\partial v\_t}$$
$$\frac{\partial J}{\partial h\_t} += \frac{\partial J}{\partial h\_next}$$

OUTPUT GATE

$$\frac{\partial J}{\partial o\_t} = \frac{\partial J}{\partial h\_t} \cdot \frac{\partial h\_t}{\partial o\_t} \Rightarrow \frac{\partial J}{\partial o\_t} = \frac{\partial J}{\partial h\_t} \odot tanh(c\_t)$$

$$\frac{\partial J}{\partial a\_o} = \frac{\partial J}{\partial o\_t} \cdot \frac{\partial o\_t}{\partial a\_o} \Rightarrow \frac{\partial J}{\partial a\_o} = \frac{\partial J}{\partial h\_t} \odot tanh(c\_t) \odot \frac{d(\sigma(a\_o))}{da\_o}$$
$$\Rightarrow \frac{\partial J}{\partial a\_o} = \frac{\partial J}{\partial h\_t} \odot tanh(c\_t) \odot \sigma(a\_o)(1 - \sigma(a\_o))$$
$$\Rightarrow \frac{\partial J}{\partial a\_o} = \frac{\partial J}{\partial h\_t} \odot tanh(c\_t) \odot o\_t(1 - o\_t)$$

$$\frac{\partial J}{\partial W\_o} = \frac{\partial J}{\partial a\_o} \cdot \frac{\partial a\_o}{\partial W\_o} \Rightarrow \frac{\partial J}{\partial W\_o} = \frac{\partial J}{\partial a\_o} \cdot z\_t^T$$
$$\frac{\partial J}{\partial b\_o} = \frac{\partial J}{\partial a\_o} \cdot \frac{\partial a\_o}{\partial b\_o} \Rightarrow \frac{\partial J}{\partial b\_o} = \frac{\partial J}{\partial a\_o}$$

CELL STATE

$$\frac{\partial J}{\partial c\_t} = \frac{\partial J}{\partial h\_t} \cdot \frac{\partial h\_t}{\partial c\_t} \Rightarrow \frac{\partial J}{\partial c\_t} = \frac{\partial J}{\partial h\_t} \odot o\_t \odot (1 - tanh(c\_t)^2)$$

$$\frac{\partial J}{\partial c\_t} += \frac{\partial J}{\partial c\_next}$$

$$\frac{\partial J}{\partial \hat{c}\_t} = \frac{\partial J}{\partial c\_t} \cdot \frac{\partial c\_t}{\partial \hat{c}\_t} \Rightarrow \frac{\partial J}{\partial \hat{c}\_t} = \frac{\partial J}{\partial c\_t} \odot i\_t$$

$$\frac{\partial J}{\partial a\_c} = \frac{\partial J}{\partial \hat{c}\_t} \cdot \frac{\partial \hat{c}\_t}{\partial a\_c} \Rightarrow \frac{\partial J}{\partial a\_c} = \frac{\partial J}{\partial c\_t} \odot i\_t \odot \frac{d(tanh(a\_c))}{da\_c}$$
$$\Rightarrow \frac{\partial J}{\partial a\_c} = \frac{\partial J}{\partial c\_t} \odot i\_t \odot (1 - tanh(a\_c)^2)$$
$$\Rightarrow \frac{\partial J}{\partial a\_c} = \frac{\partial J}{\partial c\_t} \odot i\_t \odot (1 - \hat{c}\_t^2)$$

$$\frac{\partial J}{\partial W\_c} = \frac{\partial J}{\partial a\_c} \cdot \frac{\partial a\_c}{\partial W\_c} \Rightarrow \frac{\partial J}{\partial W\_c} = \frac{\partial J}{\partial a\_c} \cdot {z\_t}^T$$

$$\frac{\partial J}{\partial b\_c} = \frac{\partial J}{\partial a\_c} \cdot \frac{\partial a\_c}{\partial b\_c} \Rightarrow \frac{\partial J}{\partial b\_c} = \frac{\partial J}{\partial a\_c}$$

INPUT GATE

$$\frac{\partial J}{\partial i\_t} = \frac{\partial J}{\partial c\_t} \cdot \frac{\partial c\_t}{\partial i\_t} \Rightarrow \frac{\partial J}{\partial i\_t} = \frac{\partial J}{\partial c\_t} \odot \hat{c}\_t$$

$$\frac{\partial J}{\partial a\_i} = \frac{\partial J}{\partial i\_t} \cdot \frac{\partial i\_t}{\partial a\_i} \Rightarrow \frac{\partial J}{\partial a\_i} = \frac{\partial J}{\partial c\_t} \odot \hat{c}\_t \odot \frac{d(\sigma(a\_i))}{da\_i}$$
$$\Rightarrow \frac{\partial J}{\partial a\_i} = \frac{\partial J}{\partial c\_t} \odot \hat{c}\_t \odot \sigma(a\_i)(1 - \sigma(a\_i))$$
$$\Rightarrow \frac{\partial J}{\partial a\_i} = \frac{\partial J}{\partial c\_t} \odot \hat{c}\_t \odot i\_t(1 - i\_t)$$

$$\frac{\partial J}{\partial W\_i} = \frac{\partial J}{\partial a\_i} \cdot \frac{\partial a\_i}{\partial W\_i} \Rightarrow \frac{\partial J}{\partial W\_i} = \frac{\partial J}{\partial a\_i} \cdot {z\_t}^T$$

$$\frac{\partial J}{\partial b\_i} = \frac{\partial J}{\partial a\_i} \cdot \frac{\partial a\_i}{\partial b\_i} \Rightarrow \frac{\partial J}{\partial b\_i} = \frac{\partial J}{\partial a\_i}$$

FORGET GATE

$$\frac{\partial J}{\partial f\_t} = \frac{\partial J}{\partial c\_t} \cdot \frac{\partial c\_t}{\partial f\_t} \Rightarrow \frac{\partial J}{\partial f\_t} = \frac{\partial J}{\partial c\_t} \odot c\_t - 1$$

$$\frac{\partial J}{\partial a\_f} = \frac{\partial J}{\partial f\_t} \cdot \frac{\partial f\_t}{\partial a\_f} \Rightarrow \frac{\partial J}{\partial a\_f} = \frac{\partial J}{\partial c\_t} \odot c\_t - 1 \odot \frac{d(\sigma(a\_f))}{da\_f}$$
$$\Rightarrow \frac{\partial J}{\partial a\_f} = \frac{\partial J}{\partial c\_t} \odot c\_t - 1 \odot \sigma(a\_f)(1 - \sigma(a\_f)$$
$$\Rightarrow \frac{\partial J}{\partial a\_f} = \frac{\partial J}{\partial c\_t} \odot c\_t - 1 \odot f\_t(1 - f\_t)$$

$$\frac{\partial J}{\partial W\_f} = \frac{\partial J}{\partial a\_f} \cdot \frac{\partial a\_f}{\partial W\_f} \Rightarrow \frac{\partial J}{\partial W\_f} = \frac{\partial J}{\partial a\_f} \cdot z\_t^T$$
$$\frac{\partial J}{\partial b\_f} = \frac{\partial J}{\partial a\_f} \cdot \frac{\partial a\_f}{\partial b\_f} \Rightarrow \frac{\partial J}{\partial b\_f} = \frac{\partial J}{\partial a\_f}$$

INPUT

$$\frac{\partial J}{\partial z\_t} = \frac{\partial J}{\partial a\_f} \cdot \frac{\partial a\_f}{\partial z\_t} + \frac{\partial J}{\partial a\_i} \cdot \frac{\partial a\_i}{\partial z\_t} + \frac{\partial J}{\partial a\_o} \cdot \frac{\partial a\_o}{\partial z\_t} + \frac{\partial J}{\partial a\_c} \cdot \frac{\partial a\_c}{\partial z\_t}$$
$$\Rightarrow \frac{\partial J}{\partial z\_t} = W\_f^T \cdot \frac{\partial J}{\partial a\_f} + W\_i^T \cdot \frac{\partial J}{\partial a\_i} + W\_o^T \cdot \frac{\partial J}{\partial a\_o} + W\_c^T \cdot \frac{\partial J}{\partial a\_c}$$

$$\frac{\partial J}{\partial h\_t - 1} = \frac{\partial J}{\partial z\_t}[: n\_h, :]$$
$$\frac{\partial J}{\partial c\_t - 1} = \frac{\partial J}{\partial c\_t} \cdot \frac{\partial c\_t}{\partial c\_t - 1} \Rightarrow \frac{\partial J}{\partial c\_t - 1} = \frac{\partial J}{\partial c\_t} \odot f\_t$$

The above equations for forward propagation and back propagation will be calculated T times (number of time steps) in each training iteration. At the end of each training iteration, the weights will be updated using the accumulated cost gradient with respect to each weight for all time steps. Assuming Stochastic Gradient Descent, the update equations are the following:

$$\frac{\partial J}{\partial W\_f} = \sum\_t^T \frac{\partial J}{\partial W\_f^t}, \qquad W\_f += \alpha * \frac{\partial J}{\partial W\_f}$$
$$\frac{\partial J}{\partial W\_i} = \sum\_t^T \frac{\partial J}{\partial W\_i^t}, \qquad W\_i += \alpha * \frac{\partial J}{\partial W\_i}$$
$$\frac{\partial J}{\partial W\_o} = \sum\_t^T \frac{\partial J}{\partial W\_o^t}, \qquad W\_o += \alpha * \frac{\partial J}{\partial W\_o}$$
$$\frac{\partial J}{\partial W\_c} = \sum\_t^T \frac{\partial J}{\partial W\_c^t}, \qquad W\_c += \alpha * \frac{\partial J}{\partial W\_c}$$
$$\frac{\partial J}{\partial W\_v} = \sum\_t^T \frac{\partial J}{\partial W\_v^t}, \qquad W\_v += \alpha * \frac{\partial J}{\partial W\_v}$$

In the next post, we will implement the above equations using Numpy and train the resulting LSTM model on real data.