Long Short-Term Memory

December 7, 2023

1 Summary

0. The paper proposes LSTM algorithm to deal with the exploding/vanishing gradient problem in RNNs. Then experiments are used to show that it outperforms existing algorithms.

1. **Background:** The paper illustrates briefly why RNN gradients explode or vanish. We illustrate the same here using a simpler univariate RNN with output only in last time step T. Such an RNN is captured by following equations:

$$\begin{aligned} a^{(t)} &= w_{ax}^{(t)} x^{(t)} + w_{ac} c^{(t-1)} \\ c^{(t)} &= \phi(a^{(t)}) \\ y^{(T)} &= c^{(T)} \end{aligned}$$

where $t \in \{1, \cdots, T\}$ and $c^{(0)} = 0$. If L is error and $\bar{v} \equiv \partial L / \partial v$ for any variable v, then the following equations capture the backprop in time,

$$\begin{aligned} \bar{c}^{(t)} &= \bar{a}^{(t+1)} w_{ac}, \quad \forall t < T \\ \bar{a}^{(t)} &= \bar{c}^{(t)} \phi'(a^{(t)}), \quad \forall t \\ \bar{w}_{ax}^{(t)} &= \bar{a}^{(t)} x^{(t)} \\ \bar{w}_{ac} &= \sum_{t=1}^{T-1} \bar{a}^{(t+1)} c^{(t)} \end{aligned}$$

Assuming ϕ is identity we see that $\bar{c}^{(1)} = w_{ac}^{T-1} \bar{c}^{(T)}$. If $w_{ac} < 1$ it is easy to see that $\bar{c}^{(1)}$ (and hence $\bar{a}^{(1)}$) vanishes and hence learning from $x^{(1)}$ is difficult. The analysis extends easily to multivariate case or other activation functions. The paper shows the same in a more general case.

2. **Constant error flow - naive approach:** From the first two equations in backprop above we derive $\bar{c}^{(t)} = \bar{c}^{(t+1)} \phi'(a^{(t+1)}) w_{ac}$. To enforce constant error flow we have $\phi'(a^{(t+1)}) w_{ac} = 1$. Integrating we get $\phi(a^{(t)}) = \frac{a^{(t)}}{w_{ac}}$ for arbitrary $a^{(t)}$. This implies

$$c^{(t+1)} = \phi(a^{(t+1)}) = \phi(w_{ac} c^{(t)}) = c^{(t)},$$

where is the second equation we ignored the x term. Therefore, ϕ is linear and the activation $c^{(t)}$ remains constant.This linear function means that the weights cannot handle conflicting inputs or outputs. This makes learning difficult in tasks with long time dependencies.

3. **LSTM.** The paper presents the LSTM algorithm centering around the above observation. The algorithm is captured by the following equations:

$$c^{(t)} = c^{(t-1)} + a_{in}^{(t)}\phi(w_{ca}a^{(t-1)})$$
$$a^{(t)} = a_{out}^{(t)}\phi(c^{(t)})$$

   where $a_{in}^{(t)} = f_{in}(w_{ina}a^{(t-1)})$ and $a_{out}^{(t)} = f_{out}(w_{outa}a^{(t-1)})$. The input and output gates learn how error signals are propagated making long term memory possible.

4. Compared to the paper we simplified the above description using the univariate case and also matching more modern notation. The rest of the paper describes experiments to show that LSTM outperforms existing algorithms.

2 Analysis

This seminal paper has stood the test of time being an indispensible tool in dealing with sequence models. Although a slightly modified version of this is used (by adding forget gates), the core idea of adding gates to the recurrent cell proved to very effective. No mathematical proof is given why the gates solve vanishing gradient problem. It has only been argued that the gates provide more avenues for the gradient to not vanish. Particularly, the passing of information through gates and *adding* it to the cell state. In practice this is working well to date.