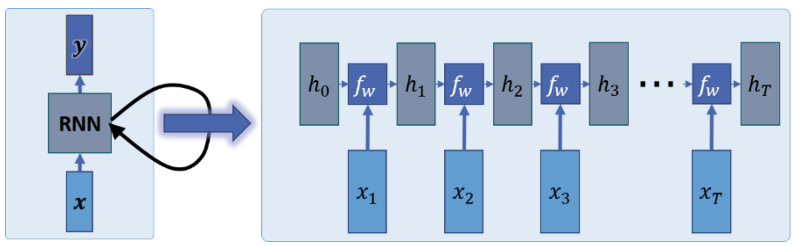
Banafshe Felfeliyan
2019-06-21

Overview

Recurrent neural networks are also known as RNNs, fundamentally are one type of neural network. RNNs main difference from feedforward networks is a feedback loop which considers the network's past decisions. This feedback loop ingests previous outputs (hidden state) as an input to the network (Figure 1). For each time step t, the activation a_t and the output y_t process of carrying memory forward mathematically can express as shown in Eq. 1**:**



$$a_t = b + W_{hh}h_{t-1} + W_{xh}x_t$$
$$h_t = tanh(a_t)$$
$$o_{ht} = c + W_{hy}h_t$$
$$\hat{y}_t = softmax\ (\ o_t)$$
$$L = \sum_t L_t$$

Figure 1: RNN (left) and RNN unfolded (right) computational graph

Eq. 1

The h_t is the hidden state at time step t which is a function of the input x_t at the same time step.

Advantages	**Drawbacks**
• Ability to process input with different length • Input size does not affect the model size • Networks consider past decisions • Weights are shared across time • Learning Sequential Data	• Slow • Does not consider future inputs for the current state • Training an RNN is a very difficult task

Backpropagation through time (BPTT): In feedforward networks, final error propagates backward for updating weights of each node. In RNN the error propagation backward by ordered series of calculation linking one-time step to the previous one is called backpropagation through time. However, this BPTT is computationally extensive, as a solution, sometimes **Truncated BPTT** is used for cost reduction.

Vanishing and exploding gradients: During backpropagation, we need to compute the backward gradient flow of RNN cells, and we need to compute $\frac{\partial L}{\partial o}$. In this situation, we end up multiplying $tanh$ to W^T every time. This amount of multiplicative gradient can be exponentially decreasing (vanishing) or increasing (exploding) with respect to the number of layers.

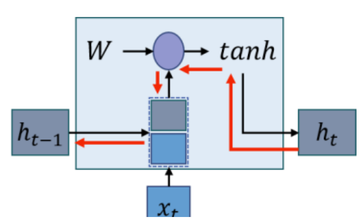


Figure 2: RNN cell and the backpropagation path (red arrows)

Overcome Vanishing Gradience

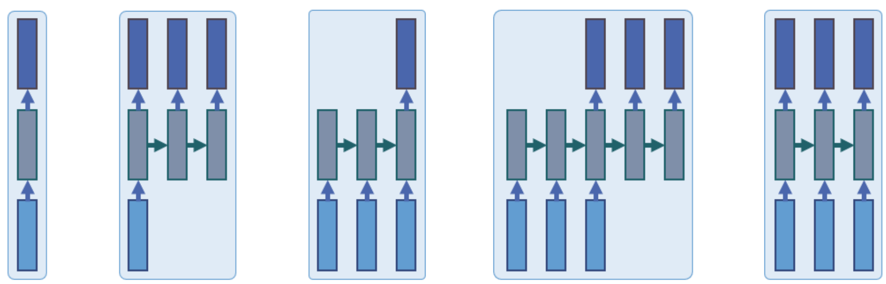
- Relu activation function
- LSTM, GRU

Overcome Exploding Gradience

- Truncated BPTT (instead of starting backprop at the last timestamp, we can choose similar timestamp, which is just before it.)
- Clip Gradience to a threshold.

RNN applications:

One to one	One to many	Many to one	Many to many	Many to many
$T_x = T_y = 1$	$T_x = 1, T_y > 1$	$T_x > 1, T_y = 1$	$T_x = T_y > 1$	$T_x \neq T_y$
Feedforward NN	Music generation	Sentiment Classification	Name entity recognition	Machine Translation



Long Short-Term Memory (LSTM):

LSTMs are one of the members of gated recurrent unit in RNN architectures which is capable of remembering information for long periods by dealing with vanishing gradient problem. LSTMs repeating module instead of having a single neural network has four neural layers known as gates.

- **f: Forget gate**, how much of past should forget
- **i: Input gate and g: gate gate**, how much of this unit is added to the current state
- **o: Output gate**, which part of the current cell makes it to the output

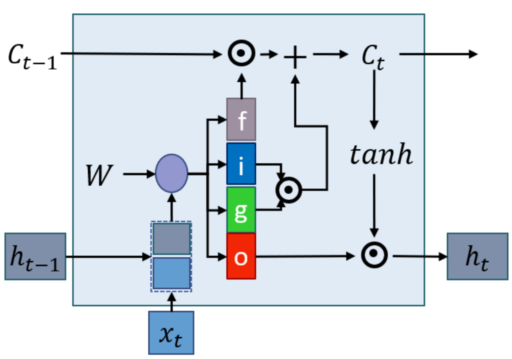


Figure 3. Representation of a LSTM cell

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{pmatrix} W \begin{pmatrix} h_t \\ x_t \end{pmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$

$$h_t = o \odot tanh(c_t)$$

Eq. 2

References: [1] Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016. [2] CS231n: Convolutional Neural Networks for Visual Recognition [3] https://skymind.ai/wiki/lstm [4] https://colah.github.io/posts/2015-08-Understanding-LSTMs/