Back Propagation in Recurrent Neural Network

Kun Yuan

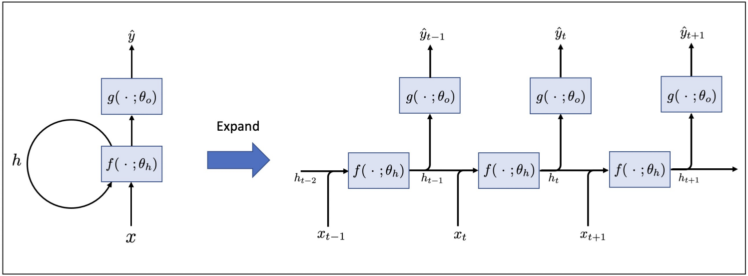
March 12, 2024

Recurrent neural network (RNN)

- RNN has the following recursion:

$$h_t = f(x_t, h_{t-1}; \theta_h)$$
$$\hat{y}_t = g(h_t; \theta_o)$$

where θ_h and θ_o are the parameters of $f(\cdot)$ and $g(\cdot)$, respectively, and h_0 can be initialized to arbitrary values.



Backpropagation in RNN

- Given a sequence of training data $\{x_t, y_t\}_{t=1}^T$, we consider the loss function

$$F(\theta_h, \theta_o) = \frac{1}{T} \sum_{t=1}^{T} L(\hat{y}_t, y_t)$$

  where $L(\hat{y}_t, y_t)$ measures the discrepancy between $\hat{y}_t$ and y_t.

- We next caculate $\nabla_{\theta_h} F(\theta_h, \theta_o)$. To this end, we have

$$\begin{aligned} \frac{\partial F(\theta_h, \theta_o)}{\partial \theta_h} &= \frac{1}{T} \sum_{t=1}^{T} \frac{\partial L(\hat{y}_t, y_t)}{\partial \theta_h} \\ &= \frac{1}{T} \sum_{t=1}^{T} \frac{\partial L(\hat{y}_t, y_t)}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial h_t} \cdot \frac{\partial h_t}{\partial \theta_h} \end{aligned}$$

- The third term $\partial h_t / \partial \theta_h$ is triky to handle.

Backpropagation in RNN

- Since $h_t = f(x_t, h_{t-1}; \theta_h)$, we have

$$\frac{\partial h_t}{\partial \theta_h} = \frac{\partial f(x_t, h_{t-1}; \theta_h)}{\partial \theta_h} + \frac{\partial f(x_t, h_{t-1}; \theta_h)}{\partial h_{t-1}} \cdot \frac{\partial h_{t-1}}{\partial \theta_h} \tag{1}$$

  which is a recursion in terms of $\partial h_t / \partial \theta_h$.

- By letting

$$\begin{aligned} a_t &= \frac{\partial h_t}{\partial \theta_h} \\ b_t &= \frac{\partial f(x_t, h_{t-1}; \theta_h)}{\partial \theta_h} \\ c_t &= \frac{\partial f(x_t, h_{t-1}; \theta_h)}{\partial h_{t-1}} \end{aligned}$$

  Recursion (1) becomes

$$a_t = b_t + c_t a_{t-1}$$

Backpropagation in RNN

- By iterating the above recursion, we have

$$a_t = b_t + \sum_{i=1}^{t-1} \Big(\prod_{j=i+1}^{t} c_j \Big) b_i.$$

Subsittuting a, b, and c, we have

$$\frac{\partial h_t}{\partial \theta_h} = \frac{\partial f(x_t, h_{t-1}; \theta_h)}{\partial \theta_h} + \sum_{i=1}^{t-1} \Big(\prod_{j=i+1}^{t} \frac{\partial f(x_j, h_{j-1}; \theta_h)}{\partial h_{j-1}} \Big) \frac{\partial f(x_i, h_{i-1}; \theta_h)}{\partial \theta_h},$$

where the chain $\prod_{j=i+1}^{t} \frac{\partial f(x_j, h_{j-1}; \theta_h)}{\partial h_{j-1}}$ can be very long for large t.

Backpropagation in RNN

- In summary, the backpropagation in RNN is derived as

$$\frac{\partial F(\theta_h, \theta_o)}{\partial \theta_h} = \frac{1}{T} \sum_{t=1}^{T} \frac{\partial L(\hat{y}_t, y_t)}{\partial \hat{y}_t} \cdot \frac{\partial \hat{y}_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial \theta_h},$$

$$\frac{\partial h_t}{\partial \theta_h} = \frac{\partial f(x_t, h_{t-1}; \theta_h)}{\partial \theta_h} + \sum_{i=1}^{t-1} \Bigg(\prod_{j=i+1}^{t} \frac{\partial f(x_j, h_{j-1}; \theta_h)}{\partial h_{j-1}} \Bigg) \frac{\partial f(x_i, h_{i-1}; \theta_h)}{\partial \theta_h}.$$

- The term $\partial F(\theta_h, \theta_o)/\partial \theta_o$ can be calculated in a similar manner.

- We next consider a concrete example.

Backpropagation in RNN

- Consider the following RNN formulation

$$h_t = W_x x_t + W_h h_{t-1}$$
$$\hat{y}_t = W_o h_t$$

where $W_x \in \mathbb{R}^{n \times d}, W_h \in \mathbb{R}^{n \times n}$, and $W_o \in \mathbb{R}^{m \times n}$ are parameters to learn, $x \in \mathbb{R}^d$ is the input data, $h \in \mathbb{R}^n$ is the hidden state, and $\hat{y} \in \mathbb{R}^m$ is the output label. We omit nonlinear activation for simplicity.

- According to the above derivations for RNN backpropagation, we have

$$\frac{\partial F}{\partial W_x} = \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{t} (W_h^\intercal)^{t-i} W_o^\intercal \frac{\partial L(\hat{y}_t, y_t)}{\partial \hat{y}_t} x_i^\intercal \in \mathbb{R}^{n \times d}.$$

$\partial F / \partial W_h$ and $\partial F / \partial W_o$ can be derived similarly.

Vanishing gradient and exploding gradient

- Recall the gradient in linear RNN:

$$\frac{\partial F}{\partial W_x} = \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{t} (W_h^\intercal)^{t-i} W_o^\intercal \frac{\partial L(\hat{y}_t, y_t)}{\partial \hat{y}_t} x_i^\intercal \in \mathbb{R}^{n \times d}.$$

- $(W_h^\intercal)^t$ will cause a significant numerical issue in $\partial F / \partial W_x$

- If the largest magnitude of the eigenvalue is less than 1, i.e., $|\lambda(W_h^\intercal)| < 1$, it holds that $(W_h^\intercal)^{t-i} \to 0$ as t (or T) gets large; Gradient vanishing!

- If the largest magnitude of the eigenvalue is greater than 1, i.e., $|\lambda(W_h^\intercal)| > 1$, it holds that $(W_h^\intercal)^t \to +\infty$ as t (or T) gets large; Gradient exploding!

- Activation functions may also amplify gradient vanishing and exploding