2024/03/26 Back Propagation Through Time For Recursive Neural Networks

Forward Pass

$$h_t = f_h(x_t, h_{t-1}) = \phi_h(W h_{t-1} + U x_t + b)$$

take 0 when t-1 is(-1
take 0 when t+1 is(T

$$\hat{y}_t = f_o(h_t) = \phi_o(V h_t + c), \quad o_t = V h_t + c$$

Backpropagation through Time

I assumed that $f_h()$ is $\tanh()$

$f_o()$ is $softmax()$ (Usually this has been being used).

$$L(\hat{y}, y) = \sum_{t=1}^{T} L_t(\hat{y}_t, y_t)$$

$$= -\sum y_t \log \hat{y}_t$$

$$= -\sum y_t \log (softmax(o_t))$$

$$SoftMax(x_t) = S_t = \frac{e^{x_t}}{\sum e^{x_k}} \quad \text{for } t = 1, \ldots, k$$

Since softMax is a $\mathbb{R}^K \to \mathbb{R}^k$ mapping function, most general Jacobian matrix for i

$$\frac{\partial S}{\partial x} = \begin{bmatrix} \frac{\partial S_1}{\partial x_1} & \cdots & \frac{\partial S_1}{\partial x_k} \\ \vdots & & \vdots \\ \frac{\partial S_k}{\partial x_1} & \cdots & \frac{\partial S_k}{\partial x_k} \end{bmatrix}$$

Let's compute $\frac{\partial S_i}{\partial x_j}$ for some arbitary i and j:

$$\frac{\partial S_i}{\partial x_j} = \frac{\partial}{\partial x_j} \frac{e^{x_i}}{\sum_K e^{x_K}}$$

$$f(x) = \frac{g(x)}{h(x)} \quad ; \quad f'(x) = \frac{g'(x)\,h(x) - g(x)\,h'(x)}{h(x)^2}$$

In our case $g_i = e^{x_i}$ and $h_i = \sum e^{x_k}$. No matter which x_j, when we compute thederivative of h_i with respect to x_j, the answer will always be e^{x_j}.

$$\frac{\partial}{\partial x_j} h_i = \frac{\partial}{\partial x_j} \sum e^{x_k} = \sum \frac{\partial e^{x_k}}{\partial x_j} = e^{x_j}$$

because $\frac{\partial e^{x_k}}{\partial x_j} = 0$ for $k \neq j$. There are only meaningfull derivatives for $i = j$

in $\frac{\partial S}{\partial x}$ matricesfor our problem.

Let's look at $i = j$ case

$$\frac{\partial \frac{e^{x_i}}{\sum e^{x_k}}}{\partial x_j} = \frac{e^{x_i} \sum e^{x_k} - e^{x_j} e^{x_i}}{\left(\sum e^{x_k}\right)^2}$$

$$= e^{x_i} \frac{\left(\sum e^{x_k} - e^{x_j}\right)}{\left(\sum e^{x_k}\right)^2}$$

$$= \frac{e^{x_i}}{\sum e^{x_k}} \cdot \left(\frac{\sum e^{x_k}}{\sum e^{x_k}} - \frac{e^{x_j}}{\sum e^{x_k}}\right)$$

$$= S_i (1 - S_j)$$

Now we found What derivative of soft-max. let's go back to the loss function.

$$L(\hat{y}, y) = -\sum y_t \log(\text{softmax}(O_t))$$

and we know that $\boxed{\frac{\partial S_i}{\partial x_j} = S_i(1 - S_j) \quad \forall\, i=j}$

$$\frac{\partial L}{\partial O_t} = -\frac{\partial}{\partial O_t} \sum y_t \log(\text{softmax}(O_t))$$

$$= -\sum y_t \frac{\partial}{\partial O_t} \log(\text{softmax}(O_t))$$

$\boxed{f(x) = \log_y x \quad f'(x) = \frac{x'}{x} \cdot \log_e y}$

$$= -\sum y_t \frac{\partial \log(\text{softmax}(O_t))}{\partial O_t}$$

$$= -\sum \frac{y_t}{\text{softmax}(O_t)} \frac{\partial \text{softmax}(O_t)}{\partial O_t} \cdot \log_e e$$

$\boxed{\hat{y}_t = \text{softmax}(O_t) = S_t}$

$$= -\sum \frac{\partial \hat{y}}{\partial O_t} \cdot \frac{y_t}{\hat{y}_t}$$

$$= -\sum S_t(1 - S_j) \cdot \frac{y_t}{\hat{y}_t}$$

$$= -\sum (1 - S_j)\, y_t$$

$$= -\sum (y_t - S_j\, y_t)$$

$$= \sum S_j\, \hat{y}_t - \sum y_t$$

$$= S_j \sum \hat{y}_t - \sum y_t$$

S_j = 1

$$= \sum \hat{y}_t - y_t$$

We know that i always equals j like

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \rightarrow \begin{bmatrix} 0.25 \\ 0.30 \\ 0.40 \\ 0.65 \end{bmatrix} \quad S_t$$

$\uparrow$

x_t

We need one derivetive for one index to that one index to S_t

j represents predict value and i represents real value at a time

Now we found $\frac{\partial L}{\partial o} = \sum \hat{y}_t - y_t$

Back propagation goes backwords by propagating over functions

$$a_t = Wh_{t-1} + Ux_t + b$$
$$h_t = \tanh(a_t)$$ } Encoder

$$o_t = Vh_t + c$$
$$\hat{y}_t = \text{softmax}(o_t)$$ } decoder

$$\frac{\partial L}{\partial V} = \sum \frac{\partial L_t}{\partial V}$$

$$= \sum \underbrace{\frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}}{\partial o_t}}_{\hat{y} - y_t} \underbrace{\frac{\partial o_t}{\partial V}}_{\text{simple coef of } V}$$

(T) this is just for integrating for multi-hidden layer.

$$= \sum (\hat{y} - y_t) \cdot h_t$$

$$\frac{\partial L}{\partial c} = \sum \frac{\partial L_t}{\partial c}$$

$$\frac{\partial L}{\partial c} = \sum \underbrace{\frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial o_t}}_{\hat{y}_t - y_t} \underbrace{\frac{\partial o_t}{\partial c}}_{\text{its constant number}}$$

$$= \sum (\hat{y}_t - y_t)$$

$$\frac{\partial L_{(t+1)}}{\partial W} = \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \underbrace{\frac{\partial h_t}{\partial W}}_{h_{t-1}}$$

- Derivation of W also depends on h_{t-1} according to recursive functi
- ($\underbrace{h_t}$ = tanh(W_t $\underbrace{h_{t-1}}$ + $U_t x_t$ + b)))

$$\frac{\partial L_t}{\partial W} = \sum_{K} \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \left(\prod_{j=k}^{T-1} \frac{\partial h_{j+1}}{\partial h_j} \right) \frac{\partial h_k}{\partial W}$$

where

$$\prod_{j=k}^{t} \frac{\partial h_{j+1}}{\partial h_j} = \frac{\partial h_{t+1}}{\partial h_k} = \frac{\partial h_{t+1}}{\partial h_t} \frac{\partial h_t}{\partial h_{t-1}} \cdots \frac{\partial h_{k+1}}{\partial h_k}$$

Thus at the time-step t+1, we can compute the gradient and futher use backpropagation through time from t+1 to 1 to compute the overall gradient with respect to W. We will finaly yield the followi gradient with respect to W:

$$\frac{\partial L_{t+1}}{\partial W} = \sum_{K=1} \frac{\partial L_{t+1}}{\partial \hat{y}_t} \cdot \frac{\partial \hat{y}_t}{\partial h_{t+1}} \cdot \frac{\partial h_{t+1}}{\partial h_k} \cdot \frac{\partial h_k}{\partial W}$$

Here we need to look at derivative of tanh() because The Gradient related to $f(x) = \tanh(x)$. Take the derivative of $\tanh(x)$ w.r.t. x

$$\frac{\partial \tanh(x)}{\partial x} = \frac{\partial \frac{\sinh(x)}{\cosh(x)}}{\partial x} = \frac{\frac{\partial \sinh(x)}{\partial (x)} \cosh(x) - \sinh(x) \frac{\partial \cosh(x)}{\partial x}}{\cosh(x)^2} = \frac{\cosh(x)^2 - \sinh(x)}{\cosh(x)^2}$$

$$= 1 - \tanh(x$$

$$\frac{\partial L_{t+1}}{\partial w} = \sum_{k=1}^{t} \frac{\partial L_{t+1}}{\partial \hat{y}_t} \cdot \frac{\partial \hat{y}_t}{\partial h_{t+1}} \cdot \frac{\partial h_{t+1}}{\partial h_k} \cdot \frac{\partial h_k}{\partial w}$$

$$\frac{\partial L}{\partial w} = \sum_{t} \sum_{k=1}^{t+1} \frac{\partial L_{t+1}}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_{t+1}} \frac{\partial h_{t+1}}{\partial h_k} \frac{\partial h_k}{\partial w}$$

Let's assume that $\frac{\partial L_{t+1}}{\partial h_{raw_k}} = \frac{\partial L_{t+1}}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_{t+1}} \frac{\partial h_{t+1}}{\partial h_k}$

Let's try to simplify it.

$$\frac{\partial L}{\partial h_{raw_k}} = \sum_{t}^{T} \sum_{k=1}^{t+1} \frac{\partial L_{t+1}}{\partial h_{raw_k}}$$

$$= \sum_{t}^{T} \frac{\partial L_{t+1}}{\partial h_{row_1}} + \frac{\partial L_{t+1}}{\partial h_{row_2}} + \frac{\partial L_{t+1}}{\partial h_{row_3}} + \cdots + \frac{\partial L_{t+1}}{\partial h_{row_{t+1}}}$$

▼ from t+1 to 0
● from T to 0

$$= \left(\frac{\partial L_{t+1}}{\partial h_{raw_1}} + \frac{\partial L_{t+1}}{\partial h_{row_2}} + \cdots + \frac{\partial L_{t+1}}{\partial h_{row_t}} + \frac{\partial L_{t+1}}{\partial h_{row_{t+1}}}\right) +$$

$$\left(\frac{\partial L_t}{\partial h_{row_1}} + \frac{\partial L_t}{\partial h_{raw_2}} + \cdots \frac{\partial L_t}{\partial h_{row_t}}\right) +$$

$$\vdots$$

$$+ \left(\frac{\partial L_1}{\partial h_{row_1}}\right)$$

We unfolded these 2 $\sum$s. Lets try to fold column by colum

instead of rowwise

$$= \sum_{t=1}^{T} \frac{\partial L_t}{\partial \text{row}_1} + \sum_{t=2}^{T} \frac{\partial L_t}{\partial \text{row}_2} + \sum_{t=3}^{T} \frac{\partial L_t}{\partial \text{row}_3} \quad \text{---------} \quad + \frac{L_t}{\partial \text{row}_t}$$

$$= \sum_{i=1}^{T} \sum_{t=i}^{T} \frac{\partial L_t}{\partial \text{row}_i}$$

We define it as T-1 sin
we are sure that it will
simplified by last element
of products

it's time to combine them.

What we found is:

$$\frac{\partial L}{\partial W} = \sum_{t}^{T} \sum_{k=1}^{t+1} \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \left(\prod_{j=k}^{T-1} \frac{\partial h_{j+1}}{\partial h_j} \right) \frac{\partial h_T}{W}$$

also we can write

$$= \sum_{t}^{T} \frac{\partial h_{T-1}^{(\text{current state})}}{W} \sum_{K=1} \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \frac{\partial h_t}{\partial h_k}$$

$$\sum_{t=i}^{T} \frac{\partial L_t}{\partial \text{raw}_i}$$

$$\frac{\partial L}{\partial W} = \sum_{t} \overbrace{h_{t-1}}^{T} \cdot \partial h_{\text{next}_i} \leftarrow \left((1 - h_t^2) \cdot V^T \cdot \frac{\partial L}{\partial \hat{y}_t} + \partial h_{\text{next}_{i+1}} \right)$$

finally we yield the following gradient with respect to W

$$\frac{\partial L}{\partial W} = \sum_{t=T}^{t=1,\ to\ 1} h_{t-1}^T \cdot next_i() \quad \begin{cases} next_i() = (1-h_t^2) \cdot V^T \cdot \frac{\partial L}{\partial \hat{y}_t} \\ \hookrightarrow next_{i+1} \end{cases}$$

Let's work on to derive the gradient w.r.t. U. Similarly, further, we can take derivate over the whole sequence as:

$$\frac{\partial L}{\partial U} = \sum_{t}^{T} \sum_{k=1}^{t+1} \frac{\partial L_{t+1}}{\partial \hat{y}_{t+1}} \frac{\partial \hat{y}_t}{\partial h_{t+1}} \left(\prod_{j=k}^{T-1} \frac{\partial h_{j+1}}{\partial h_j} \right) \frac{\partial h_T}{\partial U}$$

$$= \sum_{t}^{T} \frac{\partial h_{T-1}^{(current)}}{\partial U} \sum_{k=1}^{t+1} \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \frac{\partial h_{t-1}}{\partial h_k}$$

$$\sum_{t}^{T} \frac{\partial h_T^{(current)}}{W} \sum_{t=i}^{T} \frac{\partial L_t}{\partial row_i}$$

$$= \sum_{t=T}^{t=1,\ to\ 0} X_t \cdot next_i() \quad \begin{cases} next_i() = (1-h_t^2) \cdot V^T \cdot \frac{\partial L}{\partial \hat{y}_t} + next_{i+1}() \end{cases}$$

Let's work on to derivate the gradient w.r.t. b. Similarly, futher, we can take derivate over the whole sequence as:

$$\frac{\partial L}{\partial b} = \sum_{t}^{T} \sum_{k=1}^{t+1} \frac{\partial L_{t+1}}{\partial \hat{y}_{t+1}} \frac{\partial \hat{y}_t}{\partial h_{t+1}} \left(\prod_{j=k}^{T} \frac{\partial h_{j+1}}{\partial h_j} \right) \underbrace{\frac{\partial h_T}{\partial V}}_{1\ (its\ coef\ is\ 1.)}$$

$$\sum_{t=i}^{T} \frac{\partial L_t}{\partial row_i}$$

$$= \sum_{t=T}^{t=1,\ to\ 0} next_i() \quad \begin{cases} next_i() = (1-h_t^2) V^T \cdot \frac{\partial L}{\partial \hat{y}_t} + next_{i+1}() \end{cases}$$

Algorithm RNN BPTT $(x, h, \hat{y}, U, W, V, b, c, t, \eta)$

$dU, dW, dV = zeros\_like(U, W, V)$

$db, dc = zeros\_like(b, c)$

$dh_{next} = zeros\_like(h[0])$

for t from $lenght(x[0])$ to 0:

$dy \leftarrow \hat{y} - t$

$dV \leftarrow dV + dy \otimes h_t^T$

$dc \leftarrow dc + dy$

$dh \leftarrow V^T \otimes dy + dh_{next}$

$dh_{raw} \leftarrow (1 - h_t^2) \cdot dh$

$dW \leftarrow dW + dh_{raw} \otimes h_{t-1}^T$

$dU \leftarrow dU + dh_{raw} \otimes x_t^T$

$db \leftarrow db + dh_{raw}$

$U = U - \eta \cdot dU$

$W = W - \eta \cdot dW$

$b = b - \eta \cdot db$

$V = V - \eta \cdot dV$

$c = c - \eta \cdot dc$

return U, W, b, V, c