

5. Sequence modeling with recurrent neural networks

5.2b. Training an RNN. The backpropagation through time

Manel Martínez-Ramón
Meenu Ajith
Aswathy Rajendra Kurup

Derivation of the gradients

Gradient with respect to the output weights

- The derivation of the gradient with respect to parameters \mathbf{W}_{oh} is done from the output of the recurrent neural network.
- It can be expressed as

$$\mathbf{f}(\mathbf{x}_t) = \text{softmax} \left(\mathbf{z}_t^{(o)} \right) = \text{softmax} \left(\mathbf{W}_{oh}^\top \mathbf{h}_t + \mathbf{b}_o \right) \quad (1)$$

- This matrix is not revisited in the recursion.

Derivation of the gradients

Gradient with respect to the output weights

- The derivative of the cost function at instant t with respect to parameter $w_{oh,i,j}$ is

$$\begin{aligned}\frac{dJ_{ML}}{dw_{oh,i,j}} &= \sum_{t=1}^T \frac{dJ_{ML}}{do_{j,t}} \frac{do_{j,t}}{dz_{j,t}^{(o)}} \frac{dz_{j,t}}{dw_{oh,i,j}} \\ &= \sum_{t=1}^T \frac{dJ_{ML}}{do_{j,t}} o'_{j,t} h_{i,t} \\ &= \sum_{t=1}^T \delta_{j,t} h_{i,t}\end{aligned}\tag{2}$$

Derivation of the gradients

Gradient with respect to the output weights

- In vector notation, the gradient with respect matrix \mathbf{W}_{oh} is then

$$\nabla_{\mathbf{W}_{oh}} J_{ML}(\mathbf{o}_t) = \sum_{t=1}^T \mathbf{h}_t \boldsymbol{\delta}_t^\top \quad (3)$$

where $\boldsymbol{\delta}_t$ has components $\delta_{j,t}$. When the output of the RNN is a softmax, $\delta_{j,t} = \text{softmax}(z_{j,t}) - y_{j,t}$, therefore

$$\boldsymbol{\delta}_t = \text{softmax}(\mathbf{z}_t^{(o)}) - \mathbf{y}_t \quad (4)$$

Derivation of the gradients

Gradient with respect to the output weights

- The derivation can be repeated for biases \mathbf{b}_o

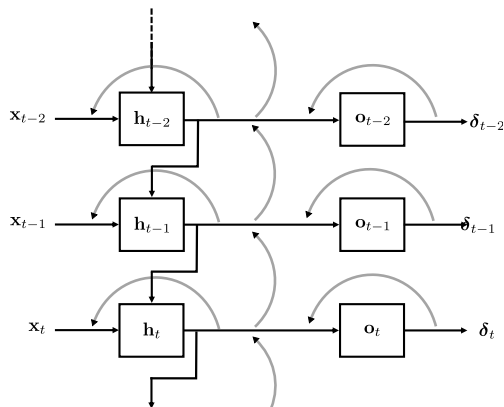
$$\begin{aligned}\frac{dJ_{ML}}{db_{o,j}} &= \sum_{t=1}^T \frac{dJ_{ML}}{do_{j,t}} \frac{do_{j,t}}{dz_{j,t}^{(o)}} \frac{dz_{j,t}}{db_{o,j}} \\ &= \sum_{t=1}^T \frac{dJ_{ML}}{do_{j,t}} o'_{j,t} \\ &= \sum_{t=1}^T \delta_{j,t}\end{aligned}\tag{5}$$

- with the result

$$\nabla_{\mathbf{b}_o} J_{ML}(\mathbf{o}_t) = \sum_{t=1}^T \boldsymbol{\delta}_t\tag{6}$$

Derivation of the gradients

Gradient with respect to the output weights



- The error term δ_t must be backpropagated through \mathbf{W}_{oh} in order to reach \mathbf{W}_{hx} at input \mathbf{x}_t .
- It has to be repeatedly backpropagated through \mathbf{W}_{hh} in order to reach the input matrix at each one of the inputs $\mathbf{x}_{t-t'}$.

Derivation of the gradients

Gradient with respect to the input weights

To compute directly the gradient with respect to the weights, the chain rule must be used as follows.

- 1 Compute the gradient ($\nabla_{\mathbf{h}_t} J_{ML}$) of the cost function respect to \mathbf{h}_t .
- 2 Compute the derivatives of the components of \mathbf{h}_t with respect to each component of $\mathbf{z}_t^{(x)}$, which gives the derivative of the tanh activation.
- 3 Compute the gradient of $\mathbf{z}_t^{(x)}$ with respect to \mathbf{W}_{hx} , which gives vector \mathbf{x}_t .

Derivation of the gradients

Gradient with respect to the input weights

- The product of these elements has to be written in the right order so the gradient has the same dimensions as matrix \mathbf{W}_{hx} .
- The result is

$$\begin{aligned}\nabla_{\mathbf{W}_{hx}} J_{ML} &= \sum_{t=1}^T \nabla_{\mathbf{W}_{hx}} \mathbf{z}_t^{(x)} (\nabla_{\mathbf{h}_t} J_{ML})^\top \frac{\delta \mathbf{h}_t}{\delta \mathbf{z}_t^{(x)}} \\ &= \sum_{t=1}^T \mathbf{x}_t (\nabla_{\mathbf{h}_t} J_{ML})^\top \text{diag} \left(\tanh' \left(\mathbf{z}_t^{(x)} \right) \right)\end{aligned}\tag{7}$$

where the derivative of the hyperbolic tangent activation is expressed as a Jacobian represented by a diagonal matrix.

Derivation of the gradients

Gradient with respect to the input weights

- A similar result can be found for the biases

$$\nabla_{\mathbf{b}_h} J_{ML} = \sum_{t=1}^T \text{diag} \left(\tanh' \left(\mathbf{z}_t^{(x)} \right) \right) (\nabla_{\mathbf{h}_t} J_{ML}) \quad (8)$$

$$\nabla_{\mathbf{b}_h} J_{ML} = \sum_{t=1}^T (\nabla_{\mathbf{h}_t} J_{ML})^\top \text{diag} \left(\tanh' \left(\mathbf{z}_t^{(x)} \right) \right) \quad (9)$$

Derivation of the gradients

Gradient with respect to the input weights

- The above equations have the same form as any previously computed gradient: it is the product of
 - ① the input sample \mathbf{x}_t as a column vector
 - ② a vector representing the backpropagated error
- The error is embedded in the (recursive) gradient with respect to the hidden state. This is, the error backpropagated to the input can be written as

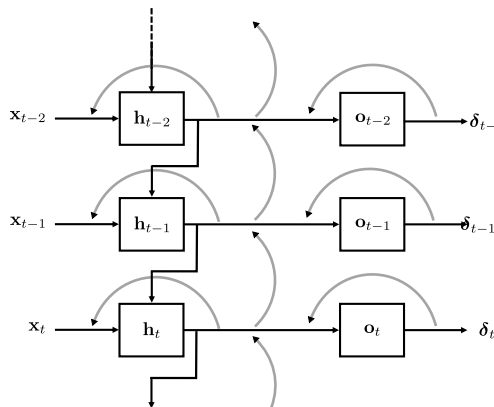
$$\delta_t^{(bp)} = \text{diag} \left(\tanh' \left(\mathbf{z}_t^{(x)} \right) \right) (\nabla_{\mathbf{h}_t} J_{ML})$$

.

Derivation of the gradients

Gradient with respect to the hidden state weights

For this set of weights, the backpropagation of the error at instant t



- 1 Starts from the output o_t .
- 2 It is transformed with output weights \mathbf{W}_{oh}
- 3 This error is used to update \mathbf{W}_{hh} . The input is h_{t-1} .
- 4 The backpropagation then goes to the previous time instant, which requires another transformation of the error \mathbf{W}_{hh} .

Derivation of the gradients

Gradient with respect to the hidden state weights

- To see this, we can compute the gradient with respect to the hidden weights.

$$\begin{aligned}\nabla_{\mathbf{W}_{hh}} J_{ML} &= \sum_{t=1}^T \nabla_{\mathbf{W}_{hh}} \mathbf{z}_t^x (\nabla_{\mathbf{h}_t} J_{ML})^\top \frac{\delta \mathbf{h}_t}{\delta \mathbf{z}_t^{(x)}} \\ &= \sum_{t=1}^T \mathbf{h}_{t-1} (\nabla_{\mathbf{h}_t} J_{ML})^\top \text{diag} \left(\tanh' \left(\mathbf{z}_t^{(x)} \right) \right)\end{aligned}\tag{10}$$

- The gradient of \mathbf{z}_t is computed now with respect to \mathbf{W}_{hh} .
- The result is the input to these weights \mathbf{h}_{t-1} .
- This is known as backpropagation through time (BPTT).
- For \mathbf{b}_h , if we remove \mathbf{h}_{t-1} from (10), we obtain the same result as in Eq. (8).

Summary of the backpropagation through time

Output weights

- Compute the output errors δ_t , $1 \leq t \leq T$.
- Use these errors to update the output weights with Eqs. (3) and (6).

$$\begin{aligned}\mathbf{W}_{oh} &\leftarrow \mathbf{W}_{oh} - \mu \sum_{t=1}^T \mathbf{h}_t \delta_t^\top \\ \mathbf{b}_{oh} &\leftarrow \mathbf{b}_{oh} - \mu \sum_{t=1}^T \delta_t\end{aligned}\tag{11}$$

Summary of the backpropagation through time

Input weights and hidden state weights

- Compute the gradient with respect to the hidden states

$$\nabla_{\mathbf{h}_t} J_{ML} = \mathbf{W}_{oh} \delta_t + \mathbf{W}_{hh} \text{diag} \left(\tanh' \left(\mathbf{z}_{t+1}^{(x)} \right) \right) (\nabla_{\mathbf{h}_{t+1}} J_{ML}) \quad (12)$$

- Update with Eqs. (7), (8), and (10).

$$\mathbf{W}_{hx} \leftarrow \mathbf{W}_{hx} - \mu \sum_{t=1}^T \mathbf{x}_t (\nabla_{\mathbf{h}_t} J_{ML})^\top \text{diag} \left(\tanh' \left(\mathbf{z}_t^{(x)} \right) \right) \quad (13)$$

$$\mathbf{b}_h \leftarrow \mathbf{b}_h - \mu \sum_{t=1}^T \text{diag} \left(\tanh' \left(\mathbf{z}_t^{(x)} \right) \right) (\nabla_{\mathbf{h}_t} J_{ML})$$

$$\mathbf{W}_{hh} \leftarrow \mathbf{W}_{hh} - \mu \sum_{t=1}^T \mathbf{h}_{t-1} (\nabla_{\mathbf{h}_t} J_{ML})^\top \text{diag} \left(\tanh' \left(\mathbf{z}_t^{(x)} \right) \right) \quad (14)$$