Recurrent Neural Network (CNN)
Training

M. R. Hasan
Deep Learning
Readings

• Geron: 15
• Zhang: 8
• Chollet: 6
What We Will Cover

• RNN: Training
• Backpropagation Through Time (BPTT) algorithm
• Forward Propagation
• Backward Propagation
• Interpretation of the loss gradient
RNN: Training
RNN: Training

- RNNs are trained by using the backpropagation algorithm.
- However, applying the backpropagation is slightly tricky in RNNs.
- We need to unroll the network in time and backward propagate the loss gradients.

This technique is known as backpropagation through time (BPTT).
RNN: Training

• Let’s **formalize** the training problem.
• We have a collection of sequence inputs \( X \) and outputs \( Y^* \), where
  • \( X = X_1, X_2, ..., X_T \)
  • \( Y^* = Y_1^*, Y_2^*, ..., Y_T^* \)

\[
\begin{align*}
W_{xh}: & \text{ size } d \times k \\
W_{hh}: & \text{ size } k \times k \\
X_t: & \text{ size } m \times d \\
H_t: & \text{ size } m \times k \\
\vec{b}_h: & \text{ size } k \times 1 \\
W_{hy}: & \text{ size } k \times k \\
Y_t: & \text{ size } m \times k
\end{align*}
\]
RNN: Training

- We want to **train RNN weight matrices** to minimize the loss between the predicted outputs of the network $Y$ and the desired outputs $Y^*$.
- This is the **most generic setting**.
- In other settings we just “remove” some of the input or output entries.

$W_{xh}$: size $d \times k$
$W_{hh}$: size $k \times k$
$X_t$: size $m \times d$
$H_t$: size $m \times k$
$\vec{b}_h$: size $k \times 1$
$W_{hy}$: size $k \times k$
$Y_t$: size $m \times k$
RNN: Backward Propagation through Time (BPTT) Algorithm
Backward Propagation Through Time (BPTT)

- The output signal is calculated at each timestep via forward propagation.
- It gives a sequence of output vectors $\vec{y}_1, \vec{y}_2, \ldots, \vec{y}_T$
- Where $T$ is the maximum timestep.
Backward Propagation Through Time (BPTT)

• For each output vector, its **loss** $l(\hat{y}_t, \hat{y}_t^*)$ is **computed** using a loss function.

• Here $\hat{y}_t^*$ is the **true output**.
Backward Propagation Through Time (BPTT)

- A suitable choice for the loss function is **cross-entropy**: 
  \[ l(\vec{y}_t, \vec{y}_t^*) = -\vec{y}_t^* \log(\vec{y}_t) \]
• The **total loss** $\mathcal{L}$ incurred by the output sequence is calculated by taking the sum of the individual losses at each timestep from $t=1 \sim T$:

$$\mathcal{L} = \sum_{t=1}^{T} l(\hat{y}_t, \hat{y}_t^*)$$

$$\mathcal{L} = -\sum_{t=1}^{T} \hat{y}_t^* \log(\hat{y}_t)$$
Backward Propagation Through Time (BPTT)

• The loss $\mathcal{L}$ is a **scalar function of a series of vectors**.
• Note that the total loss **may only include** the output of the last few steps (or just the last step).
• It depends on the problem and the architecture.
• For example, in a sequence-to-vector model, such as in **sentiment classification**, the loss is computed based on the **output of the final timestep**.

\[
\mathcal{L} = - \sum_{t=1}^{T} \tilde{y}_t^* \log(\tilde{y}_t)
\]
Backward Propagation Through Time (BPTT)

- Also, the total loss is **not always the sum** of the losses of each input.
- For example, in speech recognition, or machine translation, the predicted output sequence **length may differ** from the desired output sequence length.
- In this illustration, we consider a **special case** in which both the predicted and the desired output sequence has the **same length**.

$$\mathcal{L} = -\sum_{t=1}^{T} \tilde{y}_t \log(\tilde{y}_t)$$
Backward Propagation Through Time (BPTT)

- Based on the total loss, the **loss gradients** w.r.t. the weights are computed and propagated **backward** through the unrolled network, shown by the red arrows.
- Finally the **model parameters are updated** using the gradients computed during backward propagation.
RNN: Forward Propagation
RNN: Forward Propagation

- We pass the **entire input sequence** through the network and generate outputs.
- Following is the pseudocode description of the forward propagation in a single layer RNN.
- We use **one training instance** for the illustration.

For $t = 0 : T$

\[
\begin{align*}
    z(t) &= W_{xh} \ast x(t) + W_{hh} \ast h(t-1) + b \quad \# \text{hidden state preactivation} \\
    h(t) &= \text{activation}(z(t)) \quad \# \text{hidden state activation} \\
    o(t) &= W_{hy} \ast h(t) + b \quad \# \text{output preactivation} \\
    y(t) &= \text{activation}(o(t)) \quad \# \text{output activation}
\end{align*}
\]
RNN: Forward Propagation

- **Hidden state output**
- We use the following two forward equations for computing the hidden state and output at timestep $t$ for a single input vector $x_t$.

Preactivation signal: \[ \tilde{z}_t = W_{xh} \tilde{x}_t + W_{hh} \tilde{h}_{t-1} + \tilde{b}_h \]
Activation signal: \[ \tilde{h}_t = \phi \otimes \tilde{z}_t \]
RNN: Forward Propagation

- **Hidden state output**
- In the forward equations, we separate the preactivation and activation terms.
- The preactivation terms are denoted by $z$.
- The activation terms are denoted by $h$.

Preactivation signal: $\hat{z}_t = W_{xh} \hat{x}_t + W_{hh} \hat{h}_{t-1} + b_h$

Activation signal: $\hat{h}_t = \phi \otimes \hat{z}_t$
RNN: Forward Propagation

- **Hidden state output**
  - First, the input and the previous hidden state output are linearly combined to create the preactivation signal.

\[ \tilde{z}_t = W_{xh} \tilde{x}_t + W_{hh} \tilde{h}_{t-1} + \tilde{b}_h \]

- Then, the preactivation signal is transformed through an activation function.

\[ \tilde{h}_t = \phi \otimes \tilde{z}_t \]

Preactivation signal: \( \tilde{z}_t = W_{xh} \tilde{x}_t + W_{hh} \tilde{h}_{t-1} + \tilde{b}_h \)

Activation signal: \( \tilde{h}_t = \phi \otimes \tilde{z}_t \)
RNN: Forward Propagation

- **Output layer output**
- Similarly, for the output layer, the preactivation is computed by the linear combination of the hidden states activation (denoted by $o$) and output weights.
  \[
  \tilde{o}_t = \mathbf{W}_{hy} \tilde{h}_t + \tilde{b}_y
  \]
- Then, an activation is applied to create an output.
  \[
  \tilde{y}_t = \phi \otimes \tilde{o}_t
  \]

Output Preactivation signal: $\tilde{o}_t = \mathbf{W}_{hy} \tilde{h}_t + \tilde{b}_y$
Output Activation signal: $\tilde{y}_t = \phi \otimes \tilde{o}_t$
RNN: Forward Propagation

- Forward equations.

Hidden Preactivation signal: \( \tilde{z}_t = W_{xh} \tilde{x}_t + W_{hh} \tilde{h}_{t-1} + b_h \)

Hidden Activation signal: \( \tilde{h}_t = \phi \otimes \tilde{z}_t \)

Output Preactivation signal: \( \tilde{o}_t = W_{hy} \tilde{h}_t + b_y \)

Output Activation signal: \( \tilde{y}_t = \phi \otimes \tilde{o}_t \)
RNN: Backward Propagation
RNN: Backward Propagation

- RNN has **at least 3 types of weights** $W_{hy}$, $W_{hh}$, & $W_{xh}$ that are updated via the BPTT algorithm.
- This is done by calculating the **gradients of the loss** with respect to three weight matrices:
  - $\frac{\partial L}{\partial W_{hy}}$: loss due to the output weight
  - $\frac{\partial L}{\partial W_{hh}}$: loss due to the **recurrent** weight
  - $\frac{\partial L}{\partial W_{xh}}$: loss due to the input weight

$$W = W - \eta \frac{\partial L}{\partial W}$$
BPTT: Calculation of Loss Gradients

• Compute of the loss gradient w.r.t. $W_{hy}$

$$\frac{\partial L}{\partial W_{hy}} = \sum_{t=1}^{T} \frac{\partial l(y_t, y^*_t)}{\partial W_{hy}}$$

Applying the chain rule of calculus, we can write.

$$\frac{\partial L}{\partial W_{hy}} = \sum_{t=1}^{T} \frac{\partial l(y_t, y^*_t)}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial \hat{o}_t} \frac{\partial \hat{o}_t}{\partial W_{hy}}$$

$$\frac{\partial L}{\partial W_{hy}} = \sum_{t=1}^{T} \frac{\partial l(y_t, y^*_t)}{\partial \hat{y}_t} \left[ \phi' \otimes (\hat{o}_t) \right] \hat{h}_t$$
BPTT: Calculation of Loss Gradients

- To compute the 2\textsuperscript{nd} term, we take the derivative of a vector function $\hat{y}_t$ w.r.t. a vector $\hat{\sigma}_t$.
- Thus, the result is a matrix whose elements are all the pointwise derivatives.
- It is known as the Jacobian matrix.

Activation values depend on the choice of activation function.

$$\frac{\partial L}{\partial W_{hy}} = \sum_{t=1}^{T} \frac{\partial l(y_t, y_t^*)}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial \hat{\sigma}_t} \frac{\partial \hat{\sigma}_t}{\partial W_{hy}}$$

$$\frac{\partial L}{\partial W_{hy}} = \sum_{t=1}^{T} \frac{\partial l(y_t, y_t^*)}{\partial \hat{y}_t} \left[ \phi' \otimes (\hat{\sigma}_t) \right] \hat{h}_t$$

$$\hat{y}_t = \phi \otimes \hat{\sigma}_t$$
BPTT: Calculation of Loss Gradients

• The 3rd term is computed trivially by applying the rule of partial derivative.

\[ \hat{\sigma}_t = W_{hy} \hat{h}_t + b_y \]

\[ \frac{\partial L}{\partial W_{hy}} = \sum_{t=1}^{T} \frac{\partial l(\hat{y}_t, \hat{y}_t^*)}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial \hat{\sigma}_t} \frac{\partial \hat{\sigma}_t}{\partial W_{hy}} \]

\[ \frac{\partial L}{\partial W_{hy}} = \sum_{t=1}^{T} \frac{\partial l(\hat{y}_t, \hat{y}_t^*)}{\partial \hat{y}_t} [\phi' \otimes (\hat{\sigma}_t)] \hat{h}_t \]

\[ \hat{y}_t = \phi \otimes \hat{\sigma}_t \]
BPTT: Calculation of Loss Gradients

- We see that calculating the loss gradient w.r.t. $W_{hy}$ is trivial.
- It involves only matrix multiplication.
- However, we will see that the loss gradients w.r.t. $W_{hh}$ and $W_{xh}$ are non-trivial.

$$\frac{\partial L}{\partial W_{hy}} = \sum_{t=1}^{T} \frac{\partial l(y_t, \hat{y}_t^*)}{\partial \hat{y}_t} \left[ \phi' \otimes (\hat{o}_t) \right] \hat{h}_t$$

- $\frac{\partial L}{\partial W_{hy}}$: loss due to the output weight
- $\frac{\partial L}{\partial W_{hh}}$: loss due to the recurrent weight
- $\frac{\partial L}{\partial W_{xh}}$: loss due to the input weight
BPTT: Calculation of Loss Gradients

• Compute of the loss gradient w.r.t. $W_{hh}$

\[
\frac{\partial L}{\partial W_{hh}} = \sum_{t=1}^{T} \frac{\partial l(y_t, y^*_t)}{\partial y_t} \frac{\partial y_t}{\partial h_t} \frac{\partial h_t}{\partial W_{hh}}
\]

\[
\frac{\partial L}{\partial W_{hh}} = \sum_{t=1}^{T} \frac{\partial l(y_t, y^*_t)}{\partial y_t} \frac{\partial y_t}{\partial o_t} \frac{\partial o_t}{\partial h_t} \frac{\partial h_t}{\partial W_{hh}}
\]

\[
\frac{\partial L}{\partial W_{hh}} = \sum_{t=1}^{T} \frac{\partial l(y_t, y^*_t)}{\partial y_t} \left[ \phi' \otimes (\partial o_t) \right] W_{hy} \frac{\partial h_t}{\partial W_{hh}}
\]

Applying the chain rule of calculus, we can write.

\[
\tilde{y}_t = \phi \otimes \tilde{o}_t
\]

\[
\tilde{o}_t = W_{hy} \tilde{h}_t + \tilde{b}_y
\]

\[
\tilde{h}_t = \phi \otimes \tilde{z}_t
\]

\[
\tilde{z}_t = W_{xh} \tilde{x}_t + W_{hh} \tilde{h}_{t-1} + \tilde{b}_h
\]
BPTT: Calculation of Loss Gradients

The 2\textsuperscript{nd} and the 3\textsuperscript{rd} terms are computed trivially by applying the rule of partial derivative.

\[
\frac{\partial L}{\partial W_{hh}} = \sum_{t=1}^{T} \frac{\partial l(\hat{y}_t, \hat{y}_t^*)}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial \hat{o}_t} \frac{\partial \hat{o}_t}{\partial \hat{h}_t} \frac{\partial \hat{h}_t}{\partial W_{hh}}
\]

\[
\frac{\partial L}{\partial W_{hh}} = \sum_{t=1}^{T} \frac{\partial l(\hat{y}_t, \hat{y}_t^*)}{\partial \hat{y}_t} \left[ \phi' \otimes (\hat{o}_t) \right] W_{hy} \frac{\partial \hat{h}_t}{\partial W_{hh}}
\]

We need to compute the 4\textsuperscript{th} term.

Before we do this, let’s compute the final loss gradient w.r.t. $W_{xh}$.
BPTT: Calculation of Loss Gradients

- Compute of the loss gradient w.r.t. $W_{xh}$

\[
\frac{\partial L}{\partial W_{xh}} = \sum_{t=1}^{T} \frac{\partial l(y_t, y_t^*)}{\partial y_t} \frac{\partial y_t}{\partial h_t} \frac{\partial h_t}{\partial W_{xh}}
\]

- Applying the chain rule of calculus, we can write.

\[
\frac{\partial L}{\partial h_t} = \frac{\partial l(y_t, y_t^*)}{\partial y_t} \frac{\partial y_t}{\partial o_t} \frac{\partial o_t}{\partial h_t} \frac{\partial h_t}{\partial W_{xh}}
\]

\[
\frac{\partial L}{\partial W_{hy}} = \sum_{t=1}^{T} \frac{\partial l(y_t, y_t^*)}{\partial y_t} \left[ \phi' \otimes (\hat{o}_t) \right] W_{hy} \frac{\partial h_t}{\partial W_{xh}}
\]

\[
\hat{y}_t = \phi \otimes \hat{o}_t
\]

\[
\hat{o}_t = W_{hy} \hat{h}_t + \hat{b}_y
\]

\[
\hat{h}_t = \phi \otimes \hat{z}_t
\]

\[
\hat{z}_t = W_{xh} \hat{x}_t + W_{hh} \hat{h}_{t-1} + \hat{b}_h
\]
BPTT: Calculation of Loss Gradients

- We need to compute $\frac{\partial \vec{h}_t}{\partial W_{xh}}$ and $\frac{\partial \vec{h}_t}{\partial W_{hh}}$.

\[ \vec{h}_t = \phi \otimes \vec{z}_t \]

\[ \vec{z}_t = W_{xh} \vec{x}_t + W_{hh} \vec{h}_{t-1} + \vec{b}_h \]

\[ \vec{\gamma}_t = \phi \otimes \vec{\delta}_t \]

\[ \vec{\delta}_t = W_{hy} \vec{h}_t + \vec{b}_y \]
BPTT: Calculation of Loss Gradients

- Compute $\frac{\partial \hat{h}_t}{\partial W_{hh}}$.
- To do this, we use a simple illustration below.
- It consists of timesteps $t_1$ to $t_4$.
- We want to calculate variation in the hidden state value at $t_4$ w.r.t $W_{hh}$, i.e., $\frac{\partial \hat{h}_4}{\partial W_{hh}}$.
Compute \[ \frac{\partial \vec{h}_t}{\partial W_{hh}} \]

• Observe that current value of \( \vec{h}_4 \) is not only influenced by the current weight \( W_{hh} \).
• But also by all weights \( W_{hh} \) from previous timesteps via the intermediate hidden state activations.

\[ \vec{h}_t = \phi \otimes W_{xh} \vec{x}_t + W_{hh} \vec{h}_{t-1} + \vec{b}_h \]
Compute $\frac{\partial \vec{h}_t}{\partial W_{hh}}$

- Thus, we cannot calculate $\frac{\partial \vec{h}_4}{\partial W_{hh}}$ keeping $h_3$ as constant.

$$\vec{h}_t = \phi \otimes W_{xh} \vec{x}_t + W_{hh} \vec{h}_{t-1} + \vec{b}_h$$
Compute \( \frac{\partial \tilde{h}_t}{\partial W_{hh}} \)

- To accommodate the contribution of the previous hidden states, we divide the computation of \( \frac{\partial \tilde{h}_t}{\partial W_{hh}} \) in two parts.
  
  - **First part**: use only the influence of the weights \( W_{hh} \) at the current timestep
  
  - **Second part**: add the influence of the weights \( W_{hh} \) from the previous timestep
Compute \[
\frac{\partial \vec{h}_t}{\partial W_{hh}}
\]

\[
\vec{h}_t = \phi \otimes W_{xh} \vec{x}_t + W_{hh} \vec{h}_{t-1} + \vec{b}_h
\]

- Then, **recursively apply this calculation** backward until we reach the starting hidden state at \( t = 1 \).

\[
\frac{\partial \vec{h}_4}{\partial W_{hh}} = \frac{\partial \vec{h}_4}{\partial W_{hh}} + \frac{\partial \vec{h}_4}{\partial \vec{h}_3} \frac{\partial \vec{h}_3}{\partial W_{hh}}
\]

\[
\Rightarrow \frac{\partial \vec{h}_4}{\partial W_{hh}} = \frac{\partial \vec{h}_4}{\partial W_{hh}} + \frac{\partial \vec{h}_4}{\partial \vec{h}_3} \left[ \frac{\partial \vec{h}_3}{\partial W_{hh}} + \frac{\partial \vec{h}_3}{\partial \vec{h}_2} \frac{\partial \vec{h}_2}{\partial W_{hh}} \right]
\]

\[
\Rightarrow \frac{\partial \vec{h}_4}{\partial W_{hh}} = \frac{\partial \vec{h}_4}{\partial W_{hh}} + \frac{\partial \vec{h}_4}{\partial \vec{h}_3} \frac{\partial \vec{h}_3}{\partial W_{hh}} + \frac{\partial \vec{h}_4}{\partial \vec{h}_3} \frac{\partial \vec{h}_3}{\partial \vec{h}_2} \left[ \frac{\partial \vec{h}_2}{\partial W_{hh}} + \frac{\partial \vec{h}_2}{\partial \vec{h}_1} \frac{\partial \vec{h}_1}{\partial W_{hh}} \right]
\]

\[
\Rightarrow \frac{\partial \vec{h}_4}{\partial W_{hh}} = \frac{\partial \vec{h}_4}{\partial W_{hh}} + \frac{\partial \vec{h}_4}{\partial \vec{h}_3} \frac{\partial \vec{h}_3}{\partial W_{hh}} + \frac{\partial \vec{h}_4}{\partial \vec{h}_3} \frac{\partial \vec{h}_3}{\partial \vec{h}_2} \frac{\partial \vec{h}_2}{\partial W_{hh}} + \frac{\partial \vec{h}_4}{\partial \vec{h}_3} \frac{\partial \vec{h}_3}{\partial \vec{h}_2} \frac{\partial \vec{h}_2}{\partial \vec{h}_1} \frac{\partial \vec{h}_1}{\partial W_{hh}}
\]
Compute \( \frac{\partial \vec{h}_t}{\partial W_{hh}} \)

- We need a **simple expression** to compute the sum for a longer sequence.
- To do this, we **merge some intermediate terms**, as follows.

\[
\frac{\partial \vec{h}_4}{\partial W_{hh}} = \frac{\partial \vec{h}_4}{\partial W_{hh}} + \frac{\partial \vec{h}_4}{\partial \vec{h}_3} \frac{\partial \vec{h}_3}{\partial W_{hh}} + \frac{\partial \vec{h}_4}{\partial \vec{h}_3} \frac{\partial \vec{h}_2}{\partial W_{hh}} + \frac{\partial \vec{h}_4}{\partial \vec{h}_3} \frac{\partial \vec{h}_1}{\partial W_{hh}}
\]

\[
\frac{\partial \vec{h}_4}{\partial W_{hh}} = \frac{\partial \vec{h}_4}{\partial \vec{h}_4} \frac{\partial \vec{h}_4}{\partial W_{hh}} + \frac{\partial \vec{h}_4}{\partial \vec{h}_3} \frac{\partial \vec{h}_3}{\partial W_{hh}} + \frac{\partial \vec{h}_4}{\partial \vec{h}_2} \frac{\partial \vec{h}_2}{\partial W_{hh}} + \frac{\partial \vec{h}_4}{\partial \vec{h}_1} \frac{\partial \vec{h}_1}{\partial W_{hh}}
\]

\[
\frac{\partial \vec{h}_4}{\partial W_{hh}} = \sum_{j=1}^{4} \frac{\partial \vec{h}_4}{\partial \vec{h}_j} \frac{\partial \vec{h}_j}{\partial W_{hh}}
\]
Compute \( \frac{\partial \tilde{h}_t}{\partial W_{hh}} \)

- Now we can derive a general formula.

\[
\frac{\partial \tilde{h}_T}{\partial W_{hh}} = \sum_{t=1}^{T} \frac{\partial \tilde{h}_T}{\partial h_t} \frac{\partial h_t}{\partial W_{hh}}
\]

Using chain rule: \( \tilde{h}_T \) is influenced by an arbitrary past state \( \tilde{h}_t \):

\[
= \sum_{t=1}^{T} \frac{\partial \tilde{h}_T}{\partial h_t} \frac{\partial h_t}{\partial \tilde{z}_t} \frac{\partial \tilde{z}_t}{\partial W_{hh}}
\]

\[
\tilde{y}_t = \phi \otimes \tilde{\delta}_t
\]

\[
\tilde{\delta}_t = W_{hy} \tilde{h}_t + \vec{b}_y
\]

\[
\tilde{h}_t = \phi \otimes \tilde{z}_t
\]

\[
\tilde{z}_t = W_{xh} \tilde{x}_t + W_{hh} \tilde{h}_{t-1} + \vec{b}_h
\]
Compute $\frac{\partial \vec{h}_t}{\partial W_{hh}}$

- Now we can derive a general formula.

\[
\frac{\partial h_T}{\partial W_{hh}} = \sum_{t=1}^{T} \frac{\partial h_T}{\partial h_t} \frac{\partial h_t}{\partial W_{hh}}
\]

\[
\frac{\partial h_T}{\partial W_{hh}} = \sum_{t=1}^{T} \frac{\partial h_T}{\partial h_t} \frac{\partial h_t}{\partial W_{hh}} \frac{\partial z_t}{\partial W_{hh}}
\]

\[
\vec{h}_t = \phi \otimes \vec{z}_t
\]

\[
\vec{z}_t = W_{xh} \vec{x}_t + W_{hh} \vec{h}_{t-1} + \vec{b}_h
\]

\[
\Rightarrow \frac{\partial h_T}{\partial W_{hh}} = \sum_{t=1}^{T} \frac{\partial h_T}{\partial h_t} \left[ \phi' \otimes (\vec{z}_t) \right] \vec{h}_{t-1}
\]

[computing the derivative for the last two terms]

\[
\Rightarrow \frac{\partial h_T}{\partial W_{hh}} = \sum_{t=1}^{T} \left( \frac{\partial h_T}{\partial h_{T-1}} \frac{\partial h_{T-1}}{\partial h_{T-2}} \ldots \frac{\partial h_{t+1}}{\partial h_{t}} \right) \left[ \phi' \otimes (\vec{z}_t) \right] \vec{h}_{t-1}
\]

[applying chain rule]

\[
\Rightarrow \frac{\partial h_T}{\partial W_{hh}} = \sum_{t=1}^{T} \left( \prod_{j=t+1}^{T} \frac{\partial h_j}{\partial h_{j-1}} \right) \left[ \phi' \otimes (\vec{z}_t) \right] \vec{h}_{t-1}
\]
Compute $\frac{\partial \vec{h}_t}{\partial W_{hh}}$

- Based on this derivation, we now write an expression for an arbitrary timestep $t$, i.e., $\frac{\partial \vec{h}_t}{\partial W_{hh}}$.

\[
\frac{\partial h_T}{\partial W_{hh}} = \sum_{t=1}^{T} \left( \prod_{j=t+1}^{T} \frac{\partial h_j}{\partial h_{j-1}} \right) \left[ \phi' \otimes (\vec{z}_t) \right] \vec{h}_{t-1}
\]

\[
\frac{\partial h_t}{\partial W_{hh}} = \sum_{i=1}^{t} \left( \prod_{j=i+1}^{t} \frac{\partial h_j}{\partial h_{j-1}} \right) \left[ \phi' \otimes (\vec{z}_i) \right] \vec{h}_{i-1}
\]
Compute \( \frac{\partial \vec{h}_t}{\partial W_{hh}} \)

- Let’s compute \( \frac{\partial \vec{h}_t}{\partial \vec{h}_{t-1}} \).
- To do this, we use the forward equation.

\[
\vec{h}_t = \phi \otimes (W_{xh} \vec{x}_t + W_{hh} \vec{h}_{t-1} + \vec{b}_h)
\]

\[
\frac{\partial \vec{h}_t}{\partial \vec{h}_{t-1}} = \phi' \otimes (W_{xh} \vec{x}_t + W_{hh} \vec{h}_{t-1} + \vec{b}_h) W_{hh}
\]

\[
\frac{\partial \vec{h}_t}{\partial \vec{h}_{t-1}} = \phi' \otimes (.) W_{hh}
\]
Compute \( \frac{\partial \hat{h}_t}{\partial \mathbf{W}_{hh}} \)

- Resume computation of \( \frac{\partial \hat{h}_t}{\partial \mathbf{W}_{hh}} \)
- Let’s get back to the following calculation and complete it.

\[
\frac{\partial \hat{h}_t}{\partial \mathbf{W}_{hh}} = \sum_{i=1}^{t} (\prod_{j=i+1}^{t} \frac{\partial \hat{h}_j}{\partial \mathbf{W}_{hh}}) [\phi' \otimes (\tilde{z}_i)] \hat{h}_{i-1}
\]

\[
\frac{\partial \hat{h}_t}{\partial \mathbf{W}_{hh}} = \sum_{i=1}^{t} (\prod_{j=i+1}^{t} \phi' \otimes (\mathbf{W}_{xh} \tilde{x}_j + \mathbf{W}_{hh} \hat{h}_{j-1} + \tilde{b}_h) \mathbf{W}_{hh}) [\phi' \otimes (\tilde{z}_i)] \hat{h}_{i-1}
\]
Complete \( \frac{\partial \mathcal{L}}{\partial \mathbf{W}_{hh}} \)

\[
\frac{\partial \mathbf{h}_t}{\partial \mathbf{W}_{hh}} = \sum_{i=1}^{t} (\prod_{j=i+1}^{t} \phi' \otimes (\mathbf{W}_{xh} \mathbf{x}_j + \mathbf{W}_{hh} \mathbf{h}_{j-1} + \mathbf{b}_h) \mathbf{W}_{hh}) [\phi' \otimes (\mathbf{z}_i)] \mathbf{h}_{i-1}
\]

- Using the formula of \( \frac{\partial \mathbf{h}_t}{\partial \mathbf{W}_{hh}} \) we can now complete the calculation of \( \frac{\partial \mathcal{L}}{\partial \mathbf{W}_{hh}} \):

\[
\frac{\partial \mathcal{L}}{\mathbf{W}_{hh}} = \sum_{t=1}^{T} \frac{\partial l(\mathbf{y}_t, \mathbf{y}_t^*)}{\partial \mathbf{y}_t} [\phi' \otimes (\mathbf{o}_t)] \mathbf{W}_{hy} \frac{\partial \mathbf{h}_t}{\partial \mathbf{W}_{hh}}
\]

\[
\frac{\partial \mathcal{L}}{\mathbf{W}_{hh}} = \sum_{t=1}^{T} \frac{\partial l(\mathbf{y}_t, \mathbf{y}_t^*)}{\partial \mathbf{y}_t} [\phi' \otimes (\mathbf{o}_t)] \mathbf{W}_{hy} \sum_{i=1}^{t} (\prod_{j=i+1}^{t} \phi' \otimes (\mathbf{W}_{xh} \mathbf{x}_j + \mathbf{W}_{hh} \mathbf{h}_{j-1} + \mathbf{b}_h) \mathbf{W}_{hh}) [\phi' \otimes (\mathbf{z}_i)] \mathbf{h}_{i-1}
\]
Complete Calculation \( \frac{\partial L}{\partial W_{xh}} \)

\[
\frac{\partial L}{\partial W_{xh}} = \sum_{t=1}^{T} \frac{\partial l(y_t, y_t^*)}{\partial y_t} [\phi' \otimes (\hat{o}_t)] W_{hy} \frac{\partial \hat{h}_t}{\partial W_{xh}}
\]

- The calculation of \( \frac{\partial L}{\partial W_{xh}} \) is exactly the same except for one change in the calculation of \( \frac{\partial \hat{h}_t}{\partial W_{xh}} \):

\[
\frac{\partial \hat{h}_T}{\partial W_{xh}} = \sum_{t=1}^{T} \frac{\partial \hat{h}_T}{\partial \hat{h}_t} \frac{\hat{h}_t}{\partial W_{xh}}
\]

\[
\frac{\partial \hat{h}_T}{\partial W_{xh}} = \sum_{t=1}^{T} \frac{\partial \hat{h}_T}{\partial \hat{h}_t} \frac{\partial \hat{z}_t}{\partial W_{xh}}
\]
Complete Calculation \( \frac{\partial L}{\partial W_{xh}} \)

\[
\frac{\partial L}{\partial W_{xh}} = \sum_{t=1}^{T} \frac{\partial l(\vec{y}_t, \vec{y}_t^*)}{\partial \vec{y}_t} [\phi' \otimes (\vec{o}_t)] W_{hy} \frac{\partial \vec{h}_t}{\partial W_{xh}}
\]

\[
\vec{h}_t = \phi \otimes \vec{z}_t
\]

\[
\frac{\partial \vec{h}_T}{\partial W_{xh}} = \sum_{t=1}^{T} \frac{\partial \vec{h}_T}{\partial \vec{h}_t} \frac{\partial \vec{h}_t}{\partial \vec{z}_t} \frac{\partial \vec{z}_t}{\partial W_{xh}} \quad \text{[computing the derivative for the last two terms]}
\]

\[
\frac{\partial \vec{h}_T}{\partial W_{xh}} = \sum_{t=1}^{T} \frac{\partial \vec{h}_T}{\partial \vec{h}_{T-1}} \frac{\partial \vec{h}_{T-1}}{\partial \vec{h}_{T-2}} \ldots \frac{\partial \vec{h}_{t+1}}{\partial \vec{h}_t} [\phi' \otimes (\vec{z}_t)] \vec{x}_t \quad \text{[applying chain rule]}
\]

\[
\frac{\partial \vec{h}_T}{\partial W_{xh}} = \sum_{t=1}^{T} (\prod_{j=t+1}^{T} \frac{\partial \vec{h}_j}{\partial \vec{h}_{j-1}}) [\phi' \otimes (\vec{z}_t)] \vec{x}_t
\]
Complete Calculation \[ \frac{\partial L}{\partial W_{xh}} = \sum_{t=1}^{T} \frac{\partial l(y_t, y_t^*)}{\partial y_t} [\phi' \otimes (\tilde{o}_t)] W_{hy} \frac{\partial h_t}{\partial W_{xh}} \]

- Thus we can write a **general expression** for timestep up to \( t \):

\[ \frac{\partial h_T}{\partial W_{xh}} = \sum_{t=1}^{T} (\prod_{j=t+1}^{T} \frac{\partial h_j}{\partial h_{j-1}})[\phi' \otimes (\tilde{z}_t)] \tilde{x}_t \]

\[ \frac{\partial h_t}{\partial W_{xh}} = \sum_{i=1}^{t} (\prod_{j=i+1}^{t} \frac{\partial h_j}{\partial h_{j-1}})[\phi' \otimes (\tilde{z}_i)] \tilde{x}_i \]
Complete Calculation \[
\frac{\partial L}{\partial W_{xh}} = \sum_{t=1}^{T} \frac{\partial l(\vec{y}_t, \vec{y}_t^*)}{\partial \vec{y}_t} [\phi' \otimes (\vec{o}_t)] W_{hy} \frac{\partial \vec{h}_t}{\partial W_{xh}}
\]

- With this slight change, we can derive the following expression:

\[
\frac{\partial L}{\partial W_{xh}} = \sum_{t=1}^{T} \frac{\partial l(\vec{y}_t, \vec{y}_t^*)}{\partial \vec{y}_t} [\phi' \otimes (\vec{o}_t)] W_{hy} \sum_{i=1}^{t} (\prod_{j=i+1}^{t} \phi' \otimes (W_{xh} \vec{x}_j + W_{hh} \vec{h}_{j-1} + \vec{b}_h) W_{hh}) [\phi' \otimes (\vec{z}_i)] \vec{x}_i
\]

\[
\frac{\partial \vec{h}_t}{\partial W_{xh}} = \sum_{i=1}^{t} (\prod_{j=i+1}^{t} \frac{\partial \vec{h}_j}{\partial \vec{h}_{j-1}}) [\phi' \otimes (\vec{z}_i)] \vec{x}_i
\]
Summary: Loss Gradients

- The formulas for computing **loss gradients** w.r.t. to the three weight matrices are given below.

\[
\frac{\partial \mathcal{L}}{\partial W_{hy}} = \sum_{t=0}^{T} \frac{\partial l(y_t, \hat{y}_t^*)}{\partial \hat{y}_t} [\phi' \otimes (o_t)] \tilde{h}_t
\]

\[
\frac{\partial \mathcal{L}}{\partial W_{hh}} = \sum_{t=1}^{T} \frac{\partial l(y_t, \hat{y}_t^*)}{\partial \hat{y}_t} [\phi' \otimes (\tilde{o}_t)] W_{hy} \sum_{i=1}^{t} (\prod_{j=i+1}^{t} \phi' \otimes (W_{xh} \tilde{x}_j + W_{hh} \tilde{h}_{j-1} + b_h) W_{hh}) [\phi' \otimes (\tilde{z}_i)] \tilde{h}_{i-1}
\]

\[
\frac{\partial \mathcal{L}}{\partial W_{xh}} = \sum_{t=1}^{T} \frac{\partial l(y_t, \hat{y}_t^*)}{\partial \hat{y}_t} [\phi' \otimes (\tilde{o}_t)] W_{hy} \sum_{i=1}^{t} (\prod_{j=i+1}^{t} \phi' \otimes (W_{xh} \tilde{x}_j + W_{hh} \tilde{h}_{j-1} + b_h) W_{hh}) [\phi' \otimes (\tilde{z}_i)] \tilde{x}_i
\]

These loss gradients are computed **at each timestep**, and are used to update the model parameters starting from timestep \( T \) in the **backward direction** up to timestep 0.
Interpretation of \[
\frac{\partial \mathcal{L}}{\partial \mathcal{W}_{hh}}
\]
Interpretation of $\frac{\partial L}{\partial W_{hh}}$

- Let’s interpret the loss gradient $\frac{\partial L}{\partial W_{hh}}$.

$$\frac{\partial L}{\partial W_{hh}} = \sum_{t=1}^{T} \frac{\partial l(\hat{y}_t, \tilde{y}_t^*)}{\partial \hat{y}_t} [\phi' \otimes (\hat{o}_t)] W_{hy} \sum'_{i=1} (\prod'_{j=i+1} \phi' \otimes (W_{xh} \tilde{x}_j + W_{hh} \tilde{h}_{j-1} + b_h) W_{hh})[\phi' \otimes (\tilde{z}_i)] \tilde{h}_{i-1}$$

$$\Rightarrow \frac{\partial L}{\partial W_{hh}} = \sum_{t=1}^{T} \frac{\partial l(\hat{y}_t, \tilde{y}_t^*)}{\partial \hat{y}_t} [\phi' \otimes (\hat{o}_t)] W_{hy} \sum'_{i=1} (\prod'_{j=i+1} \phi' \otimes (.) W_{hh})[\phi' \otimes (\tilde{z}_i)] \tilde{h}_{i-1}$$

Observe that, for any timestep, the loss gradient is based on the **sum of the repeated multiplication of the product of $W_{hh}$ and the derivative of the activation function, starting from the beginning of the sequence up to the current timestep.**
Interpretation of $\frac{\partial \mathcal{L}}{\partial \mathbf{W}_{hh}}$

- In other words, we need to compute the **sum of repeated products** for $\frac{\partial \hat{h}_t}{\partial \mathbf{W}_{hh}}$

\[
\frac{\partial \mathcal{L}}{\mathbf{W}_{hh}} = \sum_{t=1}^{T} \frac{\partial l(\vec{y}_t, \vec{y}_t^*)}{\partial \vec{y}_t} [\phi' \otimes (\hat{\theta}_t)] \mathbf{W}_{hy} \sum_{i=1}^{t} (\prod_{j=i+1}^{t} \phi' \otimes (\cdot)) \mathbf{W}_{hh}[\phi' \otimes (\vec{z}_i)] \hat{h}_{i-1}
\]

\[
\frac{\partial \hat{h}_t}{\partial \mathbf{W}_{hh}} = \sum_{i=1}^{t} (\prod_{j=i+1}^{t} \phi' \otimes (\cdot)) \mathbf{W}_{hh}[\phi' \otimes (\vec{z}_i)] \hat{h}_{i-1}
\]
Interpretation of $\frac{\partial \mathcal{L}}{\partial W_{hh}}$

- Let’s evaluate the expression $\frac{\partial \tilde{h}_t}{\partial W_{hh}}$ for the first 4 timesteps as illustrated in the following figure.

$$\frac{\partial \tilde{h}_4}{\partial W_{hh}} = \sum_{i=1}^{4} (\prod_{j=i+1}^{4} \phi' \otimes (\cdot) W_{hh})[\phi' \otimes (z_i)] \tilde{h}_{i-1}$$
\[ \frac{\partial \vec{h}_4}{\partial \vec{W}_{hh}} = \sum_{i=1}^{4} (\prod_{j=i+1}^{4} \phi' \otimes (.) \vec{W}_{hh}) [\phi' \otimes (\vec{z}_i)] \vec{h}_{i-1} \]

\[ \Rightarrow \frac{\partial \vec{h}_4}{\partial \vec{W}_{hh}} = \prod_{j=2}^{4} (\phi' \otimes (.) \vec{W}_{hh}) [\phi' \otimes (\vec{z}_1)] \vec{h}_0 \\
+ \prod_{j=3}^{4} (\phi' \otimes (.) \vec{W}_{hh}) [\phi' \otimes (\vec{z}_2)] \vec{h}_1 \\
+ \prod_{j=4}^{4} (\phi' \otimes (.) \vec{W}_{hh}) [\phi' \otimes (\vec{z}_3)] \vec{h}_2 \\
+ \prod_{j=5}^{4} (\phi' \otimes (.) \vec{W}_{hh}) [\phi' \otimes (\vec{z}_4)] \vec{h}_3 \]

\[ \Rightarrow \frac{\partial \vec{h}_4}{\partial \vec{W}_{hh}} = (\phi' \otimes (.) \vec{W}_{hh})^3 [\phi' \otimes (\vec{z}_1)] \vec{h}_0 \\
+ (\phi' \otimes (.) \vec{W}_{hh})^2 [\phi' \otimes (\vec{z}_2)] \vec{h}_1 \\
+ (\phi' \otimes (.) \vec{W}_{hh})^1 [\phi' \otimes (\vec{z}_3)] \vec{h}_2 \\
+ (\phi' \otimes (.) \vec{W}_{hh})^0 [\phi' \otimes (\vec{z}_4)] \vec{h}_3 \]
We see that at timestep $t=1$ (for initial hidden state $h_0$) the product of the activation derivative and $W_{hh}$ is multiplied **three times**, then at timestep $t=2$ (for hidden state $h_1$) it is multiplied **two times**, and so on.

\[
\frac{\partial h_t}{\partial W_{hh}} = (\phi' \otimes (\cdot)W_{hh})^3[\phi' \otimes (\tilde{z}_1)]\tilde{h}_0 + (\phi' \otimes (\cdot)W_{hh})^2[\phi' \otimes (\tilde{z}_2)]\tilde{h}_1 + (\phi' \otimes (\cdot)W_{hh})^1[\phi' \otimes (\tilde{z}_3)]\tilde{h}_2 + [\phi' \otimes (\tilde{z}_4)]\tilde{h}_3
\]
- This illustration indicates that the $\frac{\partial \mathcal{L}}{\partial \mathbf{W}_{hh}}$ at any arbitrary timestep is computed by the sum of the repeated multiplication of the product of weight matrix $\mathbf{W}_{hh}$ and derivative of the activation starting from the beginning up to the current timestep.

\[
\frac{\partial \mathcal{L}}{\partial \mathbf{W}_{hh}} = \sum_{t=1}^{T} \frac{\partial l(\tilde{y}^*_t, \tilde{y}_t)}{\partial \tilde{y}_t} [\phi' \otimes (\tilde{o}_t)] \mathbf{W}_{hy} \frac{\partial \tilde{h}_t}{\partial \mathbf{W}_{hh}}
\]

\[
\frac{\partial \tilde{h}_t}{\partial \mathbf{W}_{hh}} = (\phi' \otimes (.) \mathbf{W}_{hh})^3 [\phi' \otimes (\tilde{z}_1)] \tilde{h}_0 + (\phi' \otimes (.) \mathbf{W}_{hh})^2 [\phi' \otimes (\tilde{z}_2)] \tilde{h}_1 + (\phi' \otimes (.) \mathbf{W}_{hh}) [\phi' \otimes (\tilde{z}_3)] \tilde{h}_2 + [\phi' \otimes (\tilde{z}_4)] \tilde{h}_3
\]

As we go deeper backward in time, the number of multiplications increase.