Recurrent Neural Networks

CS114 Lab 8

Kenneth Lai

March 13, 2020

Sequence Labeling

- Suppose we observe a list of words X. What are the respective parts of speech Y?

Sequence Labeling

- Suppose we observe a list of words X. What are the respective parts of speech Y?
  - What is $P(Y|X)$?

Hidden Markov Models

- Generative approach: Hidden Markov Models

Hidden Markov Models

- Generative approach: Hidden Markov Models
  - $P(Y|X) \propto P(X,Y) = P(X|Y)P(Y)$

Hidden Markov Models

- Generative approach: Hidden Markov Models
  - $P(Y|X) \propto P(X, Y) = P(X|Y)P(Y)$
- Independence Assumptions

Hidden Markov Models

- Generative approach: Hidden Markov Models
  - $P(Y|X) \propto P(X,Y) = P(X|Y)P(Y)$
- Independence Assumptions
  - (First-order) Markov Assumption: the probability of a tag depends only on the previous tag
    - $P(Y) = \prod_{i=1}^{T} P(y_i|y_{i-1})$

Hidden Markov Models

- Generative approach: Hidden Markov Models
  - $P(Y|X) \propto P(X,Y) = P(X|Y)P(Y)$
- Independence Assumptions
  - (First-order) Markov Assumption: the probability of a tag depends only on the previous tag
    - $P(Y) = \prod_{i=1}^{T} P(y_i|y_{i-1})$
  - Output Independence: the probability of a word at time i depends only on the tag at time i
    - $P(X|Y) = \prod_{i=1}^{T} P(x_i|y_i)$

Hidden Markov Models

- Generative approach: Hidden Markov Models
  - $P(Y|X) \propto P(X,Y) = P(X|Y)P(Y)$
- Independence Assumptions
  - (First-order) Markov Assumption: the probability of a tag depends only on the previous tag
    - $P(Y) = \prod_{i=1}^{T} P(y_i|y_{i-1})$
  - Output Independence: the probability of a word at time i depends only on the tag at time i
    - $P(X|Y) = \prod_{i=1}^{T} P(x_i|y_i)$
- $P(Y|X) \propto \prod_{i=1}^{T} P(x_i|y_i) \times \prod_{i=1}^{T} P(y_i|y_{i-1})$

Neural Networks

- Discriminative approach: Neural Networks

Neural Networks

- Discriminative approach: Neural Networks
  - Start small: at time i, compute $P(y_i|...)$ directly

Neural Networks

- Discriminative approach: Neural Networks
  - Start small: at time i, compute $P(y_i|\ldots)$ directly
  - For now, factor ... into two parts:

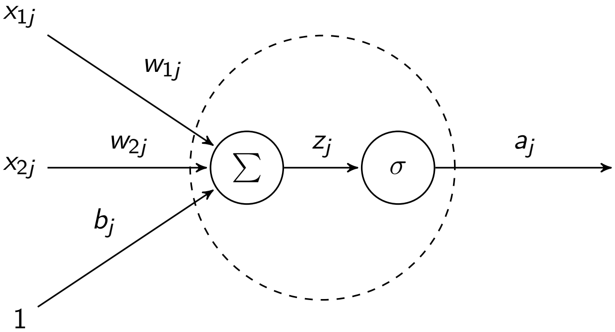
Neural Networks

- Discriminative approach: Neural Networks
  - Start small: at time i, compute $P(y_i|\ldots)$ directly
  - For now, factor ... into two parts:
    - Current word x_i

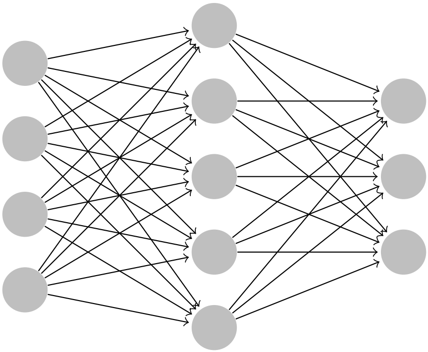
Neural Networks

- Discriminative approach: Neural Networks
  - Start small: at time i, compute $P(y_i|\ldots)$ directly
  - For now, factor ... into two parts:
    - Current word x_i
    - History/(past) context h_{i-1} = everything else useful for computing y_i

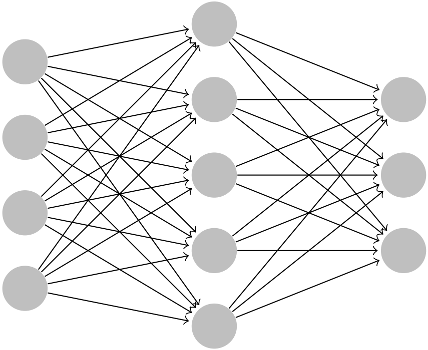
Graphical Representation of a Neuron



Neural Networks

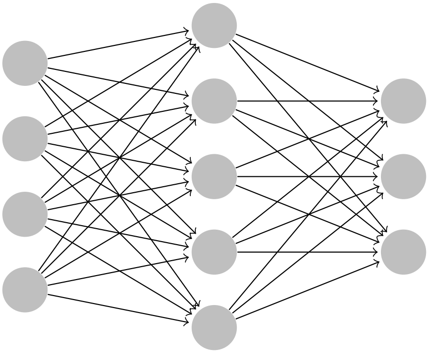


Neural Networks



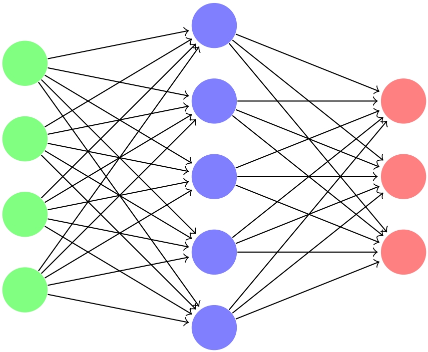
- Simplifying assumptions:

Neural Networks

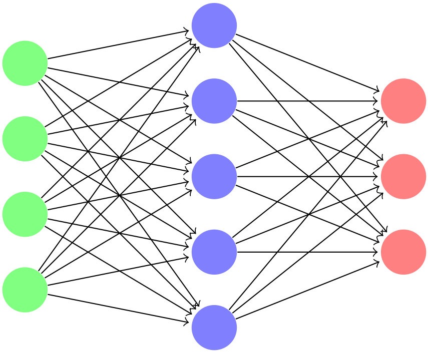


- Simplifying assumptions:
  - Suppose that our neurons are grouped into a sequence of layers
  - Also suppose that these layers are fully connected (every neuron in one layer is connected to every neuron in the next layer, and no others)

Neural Networks

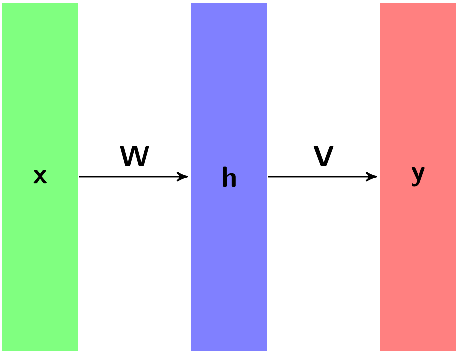


Neural Networks



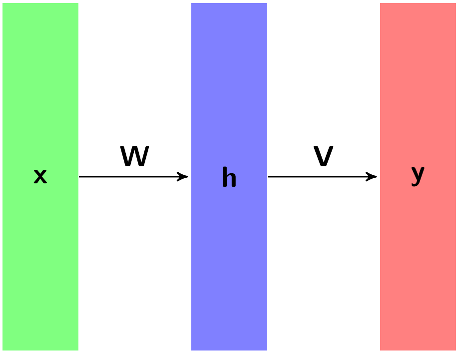
- Input layer
- Hidden layer
- Output layer

Neural Networks



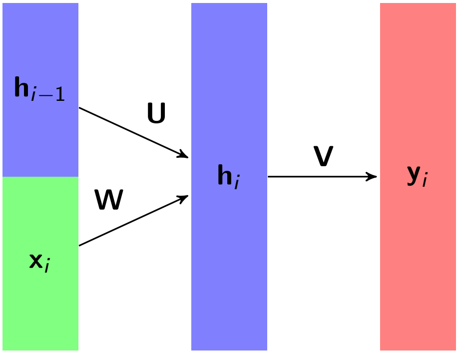
- Input layer $\mathbf{x}$
- Hidden layer $\mathbf{h}$
- Output layer $\mathbf{y}$

Neural Networks

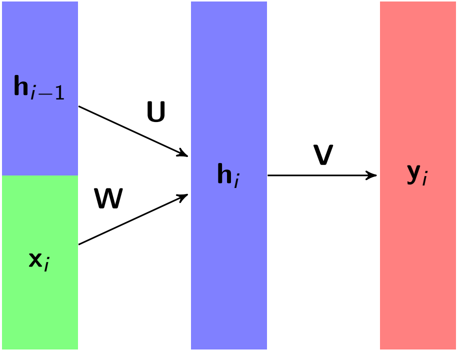


- Input layer $\mathbf{x}$
- Hidden layer $\mathbf{h}$
- Output layer $\mathbf{y}$
- Weight matrices $\mathbf{W}$, $\mathbf{V}$

Neural Networks for Sequence Labeling

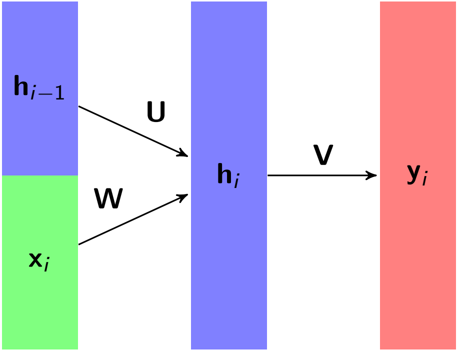


Neural Networks for Sequence Labeling



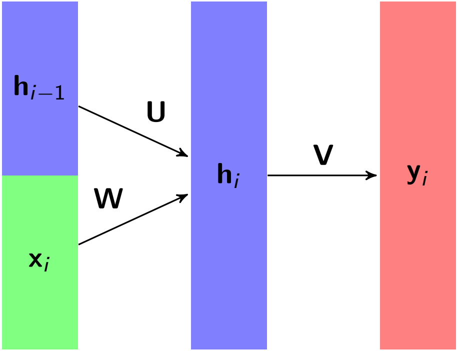
- At each time i, the input to the neural network consists of:

Neural Networks for Sequence Labeling



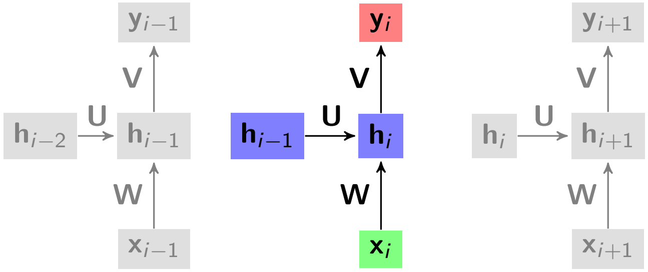
- At each time i, the input to the neural network consists of:
  - Current word vector $\mathbf{x}_i$

Neural Networks for Sequence Labeling

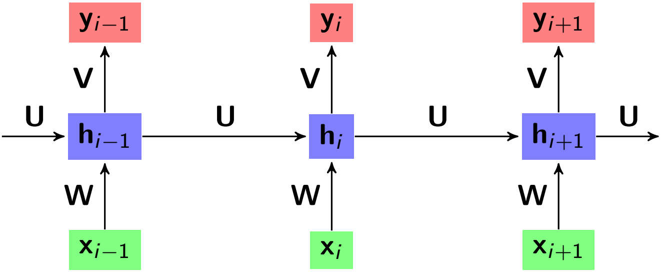


- At each time i, the input to the neural network consists of:
  - Current word vector $\mathbf{x}_i$
  - History/(past) context vector $\mathbf{h}_{i-1}$

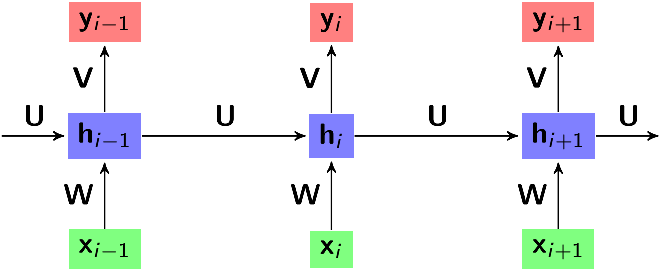
Neural Networks for Sequence Labeling



Neural Networks for Sequence Labeling

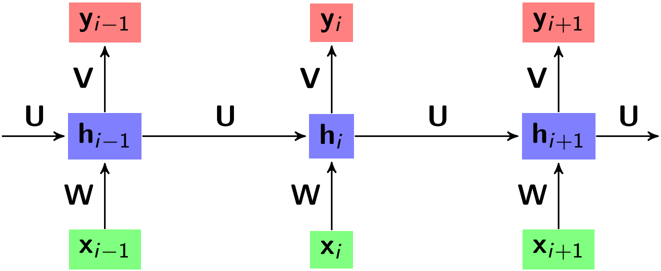


Neural Networks for Sequence Labeling



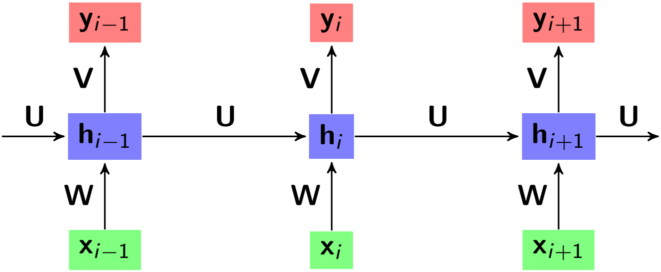
- The output of the hidden state at one time step is the history/past context input for the next time step!

Neural Networks for Sequence Labeling



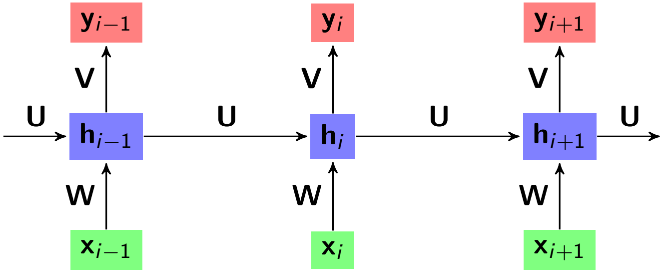
- The output of the hidden state at one time step is the history/past context input for the next time step!
- What context information is embedded in $\mathbf{h}_{i-1}$?

Neural Networks for Sequence Labeling



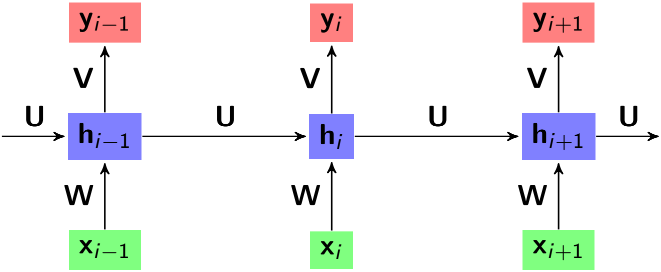
- The output of the hidden state at one time step is the history/past context input for the next time step!
- What context information is embedded in $\mathbf{h}_{i-1}$?
  - Previous word $\mathbf{x}_{i-1}$
  - Previous context $\mathbf{h}_{i-2}$

Neural Networks for Sequence Labeling



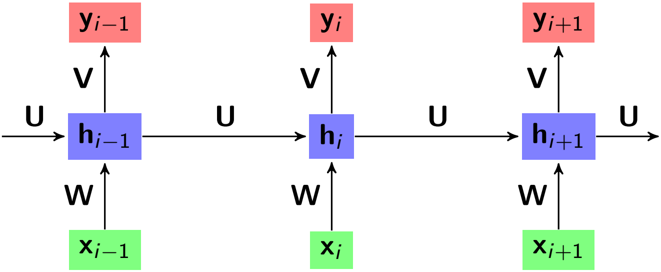
- The output of the hidden state at one time step is the history/past context input for the next time step!
- What context information is embedded in $\mathbf{h}_{i-1}$?
  - Previous word $\mathbf{x}_{i-1}$
  - Previous context $\mathbf{h}_{i-2}$
    - Previous previous word $\mathbf{x}_{i-2}$
    - Previous previous context $\mathbf{h}_{i-3}$

Neural Networks for Sequence Labeling



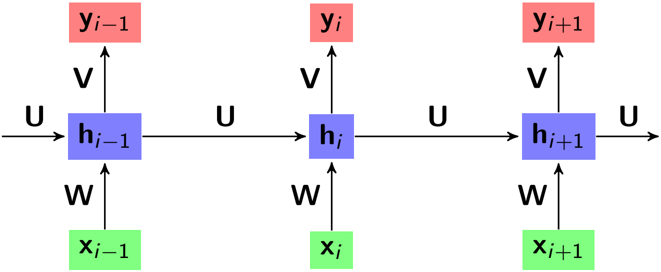
- The output of the hidden state at one time step is the history/past context input for the next time step!
- What context information is embedded in $\mathbf{h}_{i-1}$?
  - All previous words

Neural Networks for Sequence Labeling



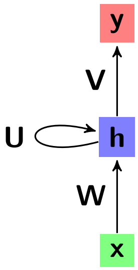
- The output of the hidden state at one time step is the history/past context input for the next time step!
- What context information is embedded in $\mathbf{h}_{i-1}$?
  - All previous words
  - What about previous parts of speech (as in HMMs)?

Neural Networks for Sequence Labeling

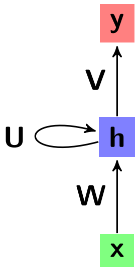


- The output of the hidden state at one time step is the history/past context input for the next time step!
- What context information is embedded in $\mathbf{h}_{i-1}$?
  - All previous words
  - What about previous parts of speech (as in HMMs)?
    - At least enough information to predict previous tags

Recurrent Neural Networks

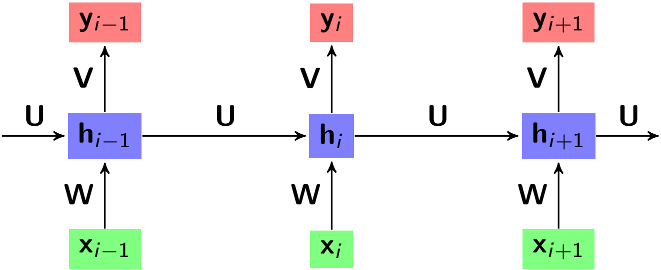


Recurrent Neural Networks

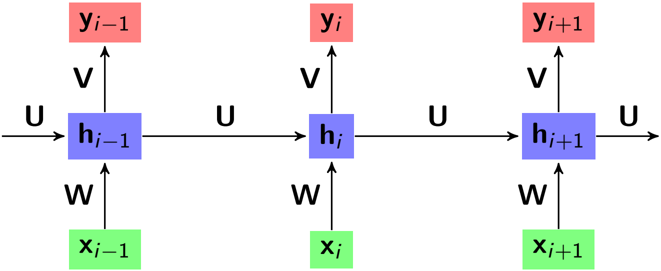


- Neural networks in which the output of a layer in one time step is input to a layer in the next time step

RNN Language Models

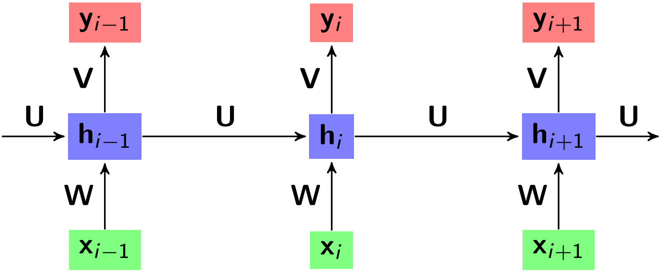


RNN Language Models



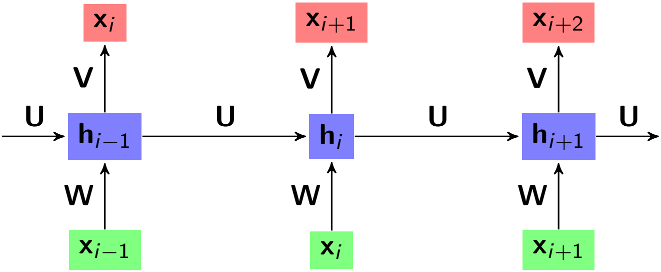
- Sequence labeling: predict current tag given current word, history

RNN Language Models



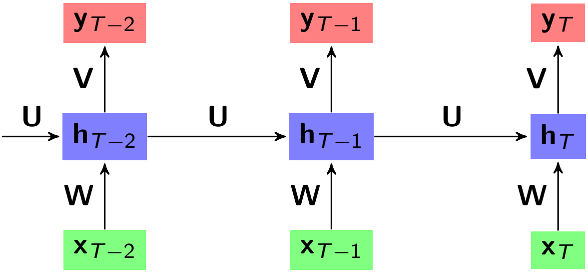
- Sequence labeling: predict current tag given current word, history
- Language modeling: predict next word given current word, history

RNN Language Models

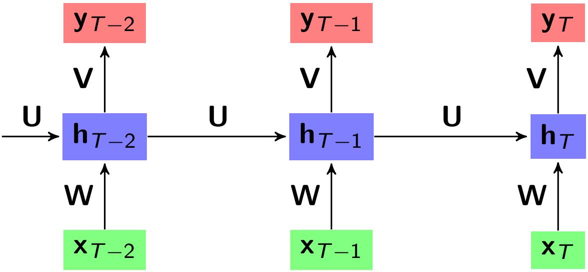


- Sequence labeling: predict current tag given current word, context
- Language modeling: predict next word given current word, context

RNNs for Text Classification

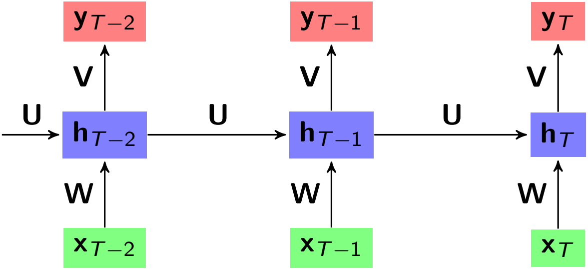


RNNs for Text Classification



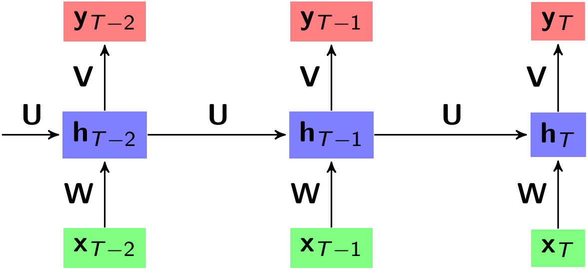
- What context information is embedded in $\mathbf{h}_T$?

RNNs for Text Classification



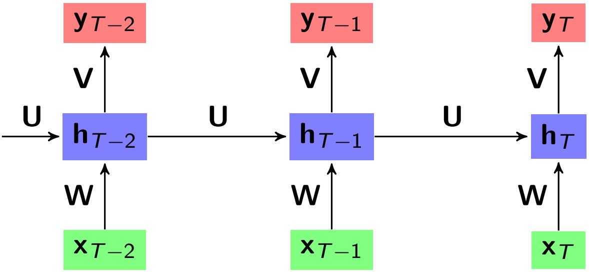
- What context information is embedded in $\mathbf{h}_T$?
  - Current word $\mathbf{x}_T$
  - Context $\mathbf{h}_{T-1}$

RNNs for Text Classification



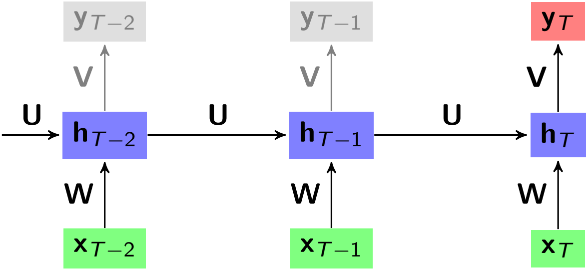
- What context information is embedded in $\mathbf{h}_T$?
  - All words (i.e. the whole text)

RNNs for Text Classification



- What context information is embedded in $\mathbf{h}_T$?
  - All words (i.e. the whole text)
- Use $\mathbf{h}_T$ to predict class $\mathbf{y}_T$ of entire document

RNNs for Text Classification

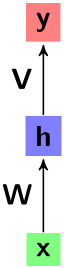


- What context information is embedded in $\mathbf{h}_T$?
  - All words (i.e. the whole text)
- Use $\mathbf{h}_T$ to predict class $\mathbf{y}_T$ of entire document
  - Ignore other outputs

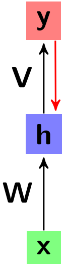
Backpropagation

- For each matrix of weights $\mathbf{W}$, starting from the output and working backwards:
  - Compute gradient $\nabla_{\mathbf{W}} L$
- For each matrix of weights $\mathbf{W}$:
  - Move in direction of negative gradient

Backpropagation

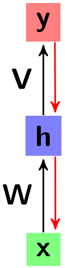


Backpropagation



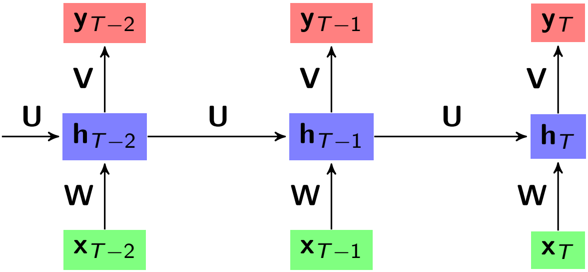
- Compute gradient $\nabla_{\mathbf{V}} L$

Backpropagation

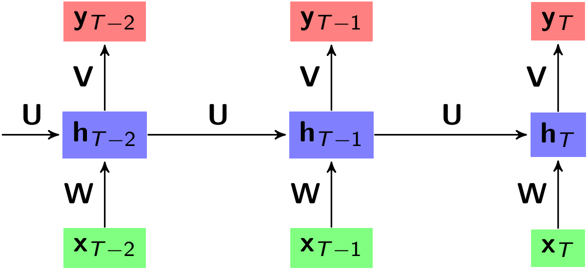


- Compute gradient $\nabla_{\mathbf{V}} L$
- Use $\nabla_{\mathbf{V}} L$ to compute gradient $\nabla_{\mathbf{W}} L$

Backpropagation Through Time

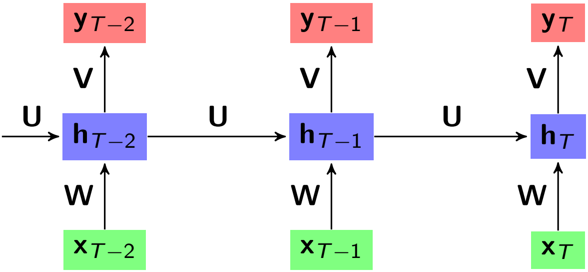


Backpropagation Through Time



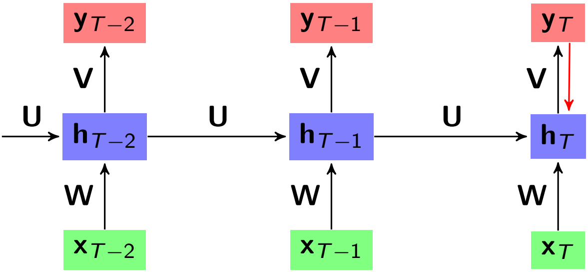
- Start at the end of the text and work backwards

Backpropagation Through Time



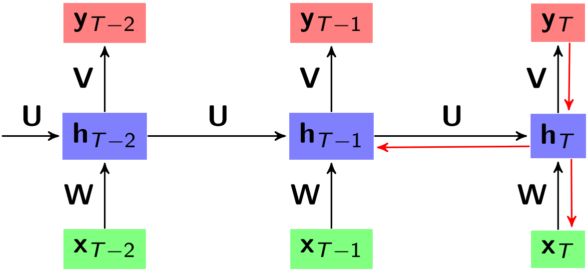
- Start at the end of the text and work backwards
  - Let $\nabla_{\mathbf{W},i,j} L$ denote the part of the gradient for weight matrix $\mathbf{W}$ at time i that comes from the output at time j

Backpropagation Through Time



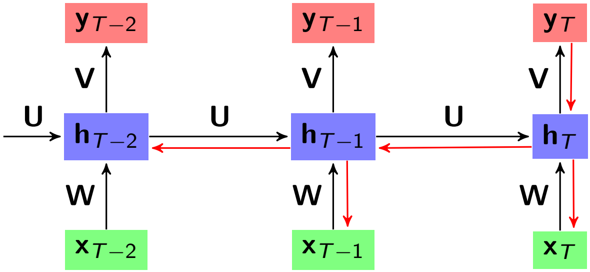
- Start at the end of the text and work backwards
  - Compute gradient $\nabla_{\mathbf{v},T,T} L$

Backpropagation Through Time



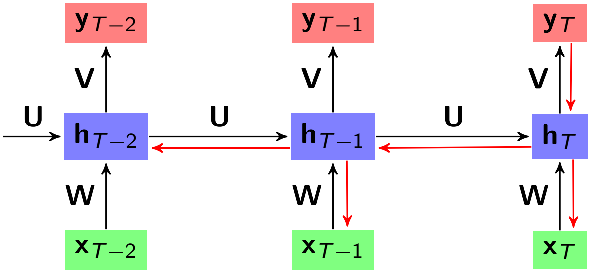
- Start at the end of the text and work backwards
  - Compute gradient $\nabla_{\mathbf{V},T,T} L$
  - Use $\nabla_{\mathbf{V},T,T} L$ to compute gradients $\nabla_{\mathbf{W},T,T} L$ and $\nabla_{\mathbf{U},T,T} L$

Backpropagation Through Time



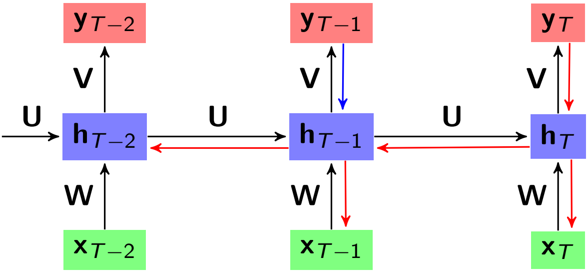
- Start at the end of the text and work backwards
  - Compute gradient $\nabla_{\mathbf{V},T,T}L$
  - Use $\nabla_{\mathbf{V},T,T}L$ to compute gradients $\nabla_{\mathbf{W},T,T}L$ and $\nabla_{\mathbf{U},T}L$
  - Use $\nabla_{\mathbf{V},T-1,T}L$ to compute gradients $\nabla_{\mathbf{W},T-1,T}L$ and $\nabla_{\mathbf{U},T-1,T}L$

Backpropagation Through Time



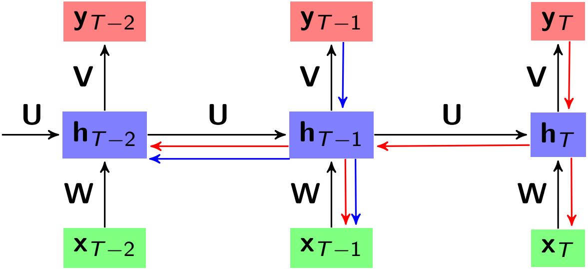
- Start at the end of the text and work backwards
  - Compute gradient $\nabla_{\mathbf{V},T,T}L$
  - Use $\nabla_{\mathbf{V},T,T}L$ to compute gradients $\nabla_{\mathbf{W},T,T}L$ and $\nabla_{\mathbf{U},T}L$
  - Use $\nabla_{\mathbf{V},T-1,T}L$ to compute gradients $\nabla_{\mathbf{W},T-1,T}L$ and $\nabla_{\mathbf{U},T-1,T}L$
  - etc.

Backpropagation Through Time



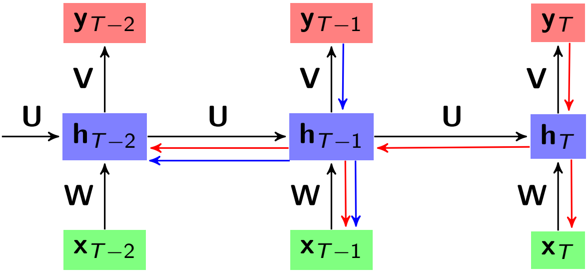
- Start at the end of the text and work backwards
  - Compute gradient $\nabla_{\mathbf{V},T-1,T-1} L$

Backpropagation Through Time



- Start at the end of the text and work backwards
  - Compute gradient $\nabla_{\mathbf{V},T-1,T-1}L$
  - Use $\nabla_{\mathbf{V},T-1,T-1}L$ to compute gradients $\nabla_{\mathbf{W},T-1,T-1}L$ and $\nabla_{\mathbf{U},T-1,T-1}L$

Backpropagation Through Time



- Start at the end of the text and work backwards
  - Compute gradient $\nabla_{\mathbf{V},T-1,T-1}L$
  - Use $\nabla_{\mathbf{V},T-1,T-1}L$ to compute gradients $\nabla_{\mathbf{W},T-1,T-1}L$ and $\nabla_{\mathbf{U},T-1,T-1}L$
  - etc.

Backpropagation Through Time

- The overall gradient for a weight matrix $\mathbf{W}$ is the sum of the gradients at each time i from each output $\mathbf{y}_j$

Backpropagation Through Time

- The overall gradient for a weight matrix $\mathbf{W}$ is the sum of the gradients at each time i from each output $\mathbf{y}_j$
  - $\nabla_{\mathbf{W}} L = \sum_{j=1}^{T} \sum_{i=1}^{j} \nabla_{\mathbf{W},i,j} L$

Backpropagation Through Time

- The overall gradient for a weight matrix $\mathbf{W}$ is the sum of the gradients at each time i from each output $\mathbf{y}_j$
  - $\nabla_{\mathbf{W}} L = \sum_{j=1}^{T} \sum_{i=1}^{j} \nabla_{\mathbf{W},i,j} L$
- Then move in direction of negative gradient (assuming stochastic gradient descent)