UNIVERSITY OF TARTU
1632

Recurrent Neural Networks

Datamining Seminar

Kaspar Märtens
Karl-Oskar Masing

Today's Topics

- Modeling sequences: a brief overview
- Training RNNs with back propagation
- A toy example of training an RNN
- Why is it difficult to train RNN
- Long-term Short-term-memory

Today's Topics

- Modeling sequences: a brief overview
- Training RNNs with back propagation
- A toy example of training an RNN
- Why is it difficult to train RNN
- Long-term Short-term-memory

Modeling sequences

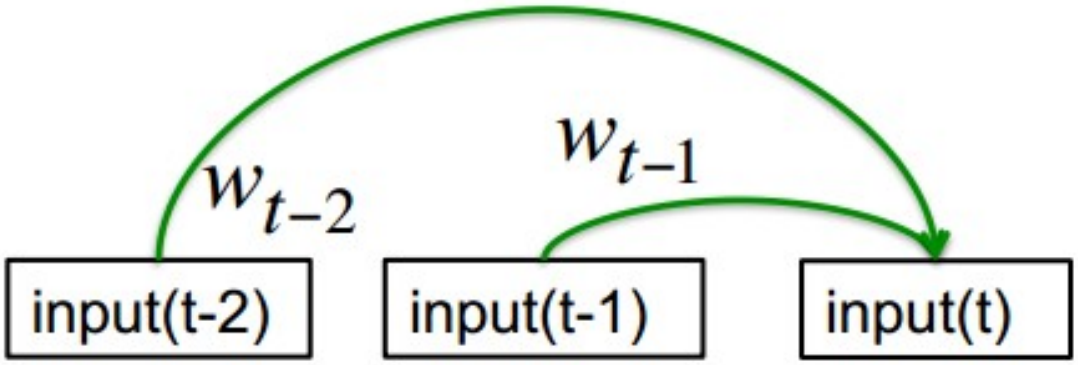
- Goal
  - Turn an input sequence into an output sequence

    01101010 → 00100110
- We often don't have a separate target sequence
  - We can try to predict the next term in the sequence

    011 → 011 + ?
  - Blurred distinction between supervised and unsupervised learning
    - Methods from supervised
    - Doesn't need separate teaching signal

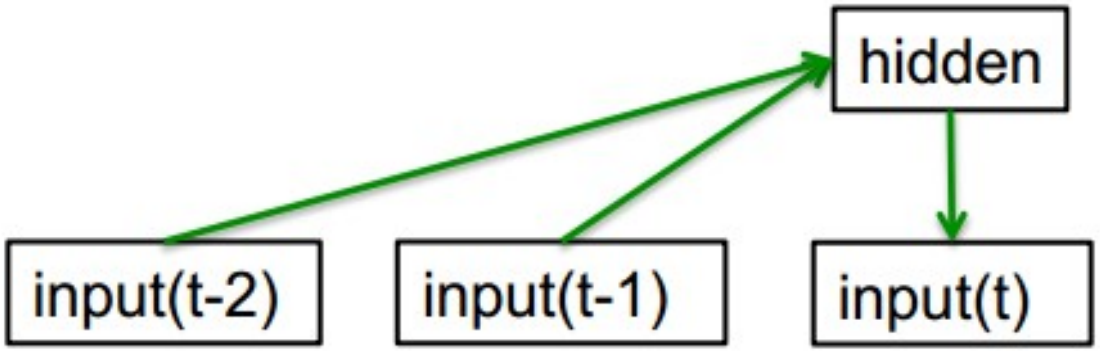
Memoryless models for sequences

- Autoregressive models
  - Predict the next term in a sequence from a fixed number of previous terms using “delay taps”.



Memoryless models for sequences

- Feed-forward neural nets
  - These generalize autoregressive models by using one or more layers of non-linear hidden units.



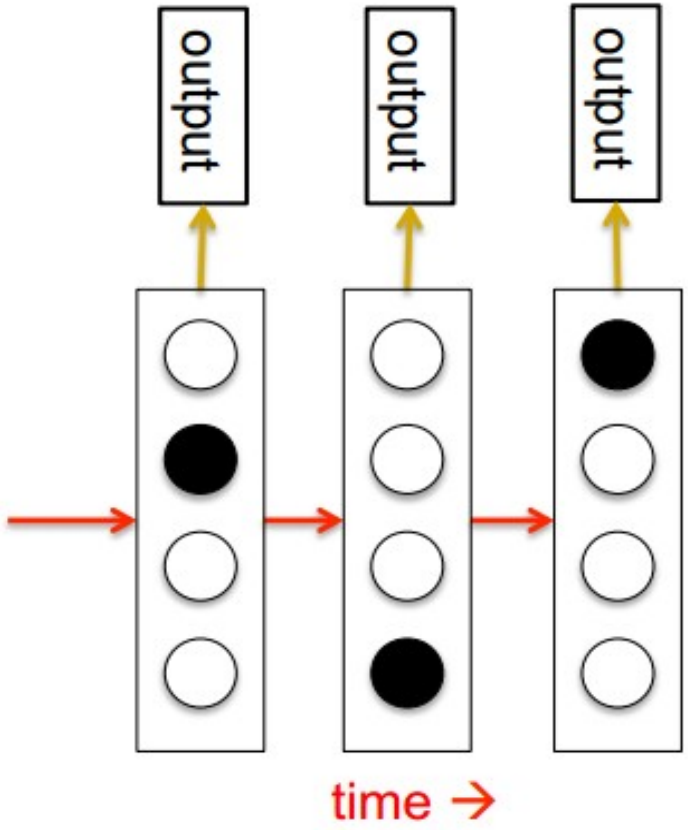
Beyond memoryless models

- We can give our generative model some hidden state

Hidden Markov Models

- Has a discrete one-of-N hidden state.
- Transitions between states are stochastic.
- The outputs produced by a state are stochastic.
  - We can't observe the states as they're hidden.

Hidden Markov Models

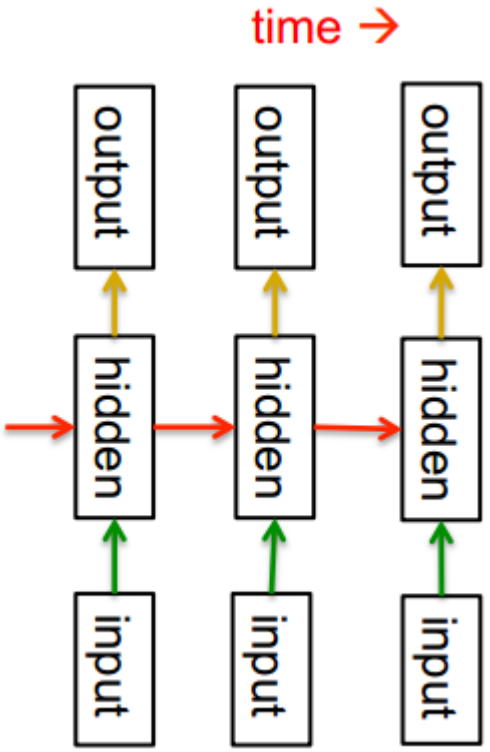


A fundamental limitation of HMMs

- At each time step it must select one of its hidden states.
  - With N hidden states it can only remember log(N) bits about what it generated so far.
- There are difficult problems
  - For utterances in voice recognition we need to store more than 100 bits of information
    - We need 2^100 hidden states

Recurrent neural networks

- Very powerful
  - All Turing machines may be simulated by fully connected recurrent networks built of neurons with sigmoidal activation functions.
    - Proof: Siegelmann & Sontag, 1991, Applied Mathematics Letters, vol 4, pp 77-80.



Recurrent neural networks

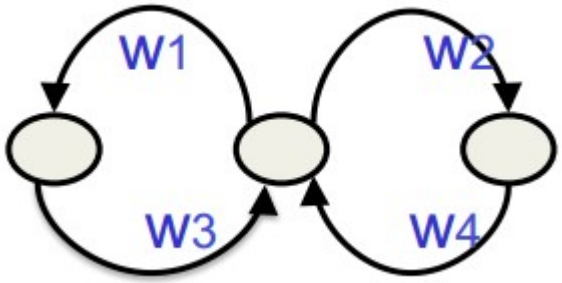
- Loops
- Applications
  - Reading cursive handwriting
  - Voice recognition
  - Collaborative filtering

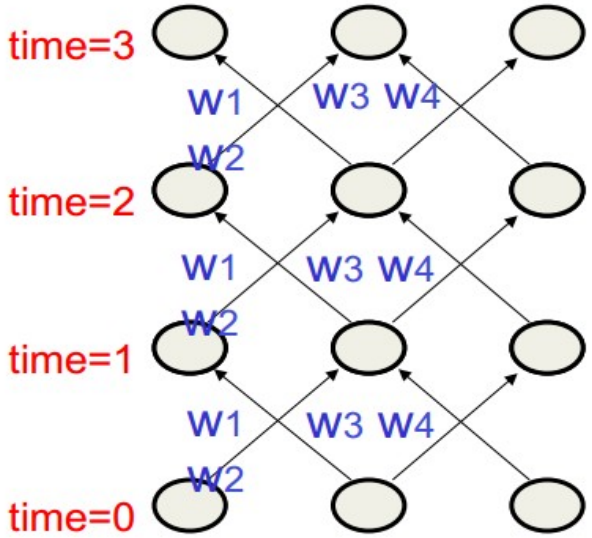
Today's Topics

- Modeling sequences: a brief overview
- Training RNNs with back propagation
- A toy example of training an RNN
- Why is it difficult to train RNN
- Long-term Short-term-memory

Recurrent nets as feed-forward nets

- The recurrent net is a layered net with one time unit per layer that keeps reusing the same weights.





Reminder: backpropagation with weight constraints

- Easy to modify the algorithm to incorporate linear constraints between the weight.
  - Have to compute the gradients as usual, and then modify the gradients so that they satisfy the constraints.

$$\textit{To constrain: } w_1 = w_2$$

$$\textit{we need: } \Delta w_1 = \Delta w_2$$

$$\textit{compute: } \frac{\partial E}{\partial w_1} \textit{ and } \frac{\partial E}{\partial w_2}$$

$$\textit{use } \frac{\partial E}{\partial w_1} + \frac{\partial E}{\partial w_2} \textit{ for } w_1 \textit{ and } w_2$$

Backpropagation through time

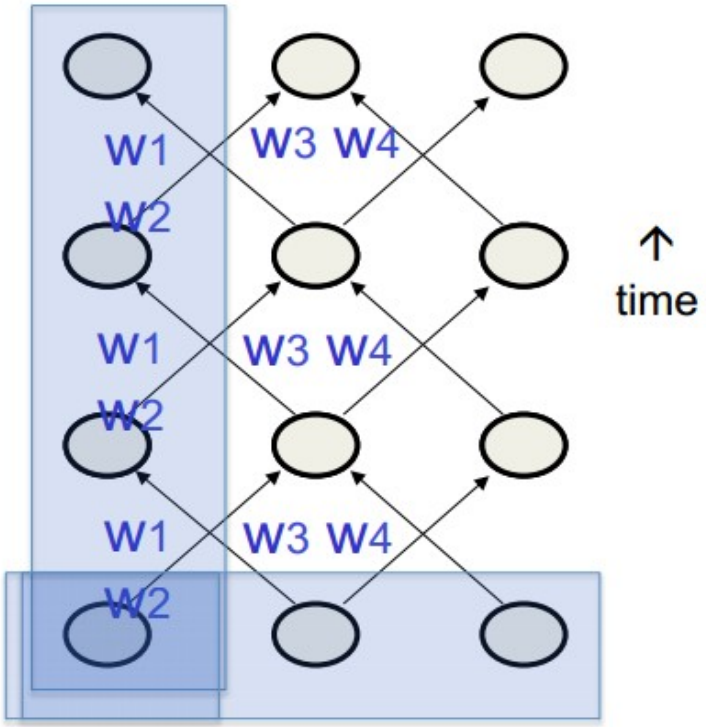
- Think of recurrent net as layered
- Feed-forward net with shared weights
- Train the feed-forward net with weight constraints

An irritating issue

- Need to specify the initial activity state for all the hidden and output unit
  - Fix to some default values (e.g. 0.5)
  - Treat them as parameters which we'll learn

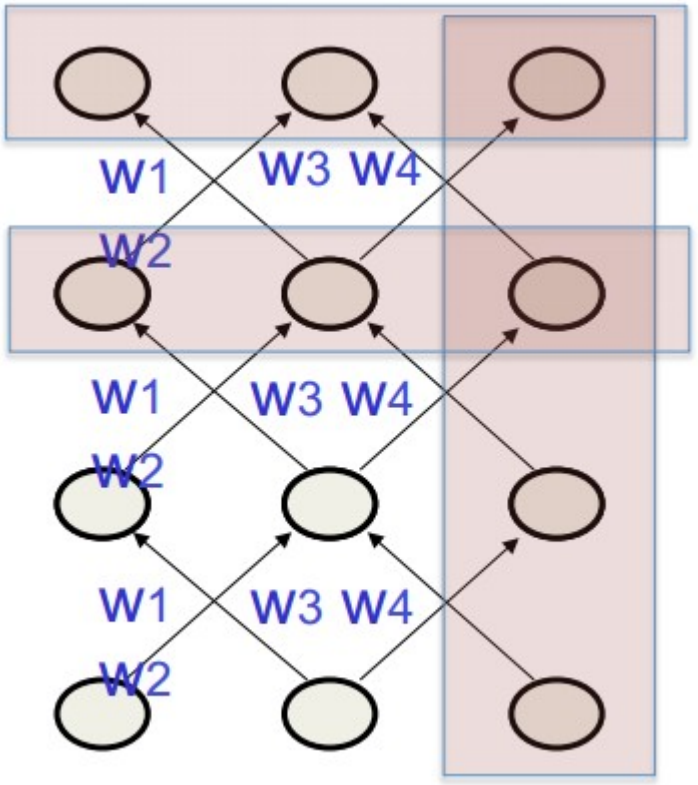
Providing input to recurrent networks

- We can specify inputs in several ways.



Target for recurrent networks

- We can specify targets in serveral ways.

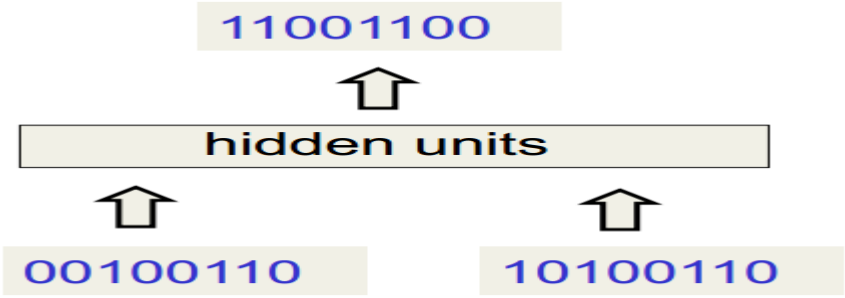


Today's Topics

- Modeling sequences: a brief overview
- Training RNNs with back propagation
- A toy example of training an RNN
- Why is it difficult to train RNN
- Long-term Short-term-memory

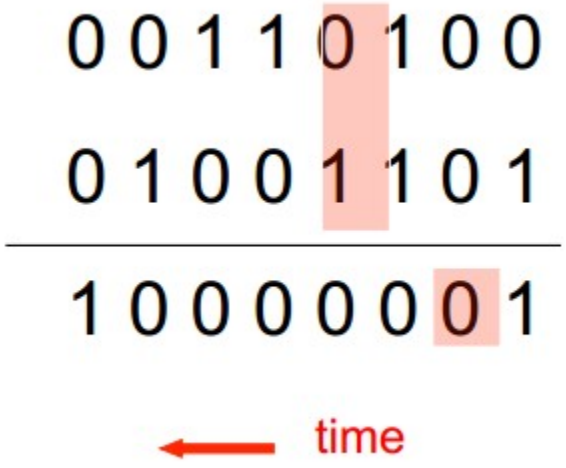
A toy example – binary addition

- We can train a feed-forward net to the the addition, but there are deficiencies
  - The maximum number of digits must be decided in advance
  - Processing applied to the beginning of a long number does not generalize to the end of the long number – uses different weights



A recurrent net for binary addition

- The network has two input units and one output unit
- It is given two input digits at each time step
- The desired output at each time step is the output of the column that provided as input two time steps ago.

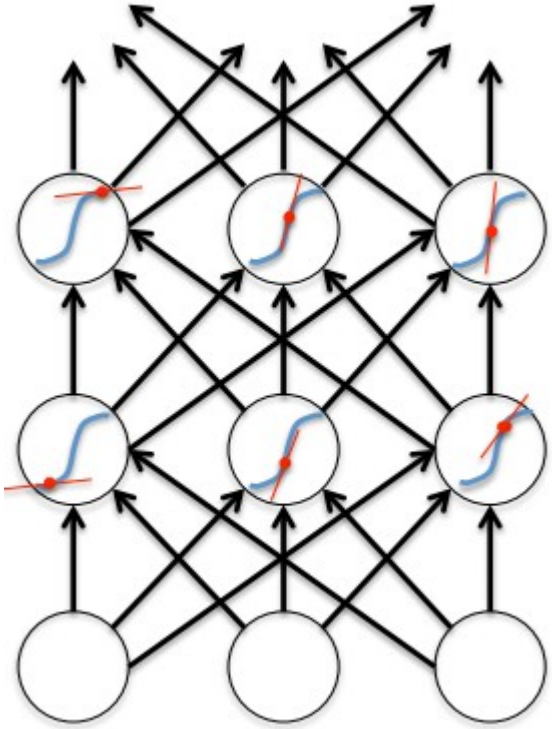


Today's Topics

- Modeling sequences: a brief overview
- Training RNNs with back propagation
- A toy example of training an RNN
- Why is it difficult to train RNN
- Long-term Short-term-memory

The backward pass is linear

- Big differences between the forward and backward passes
  - Forward is kinda bounded by logistic functions
  - Backward is linear – unbouded due to logistic function's derivative

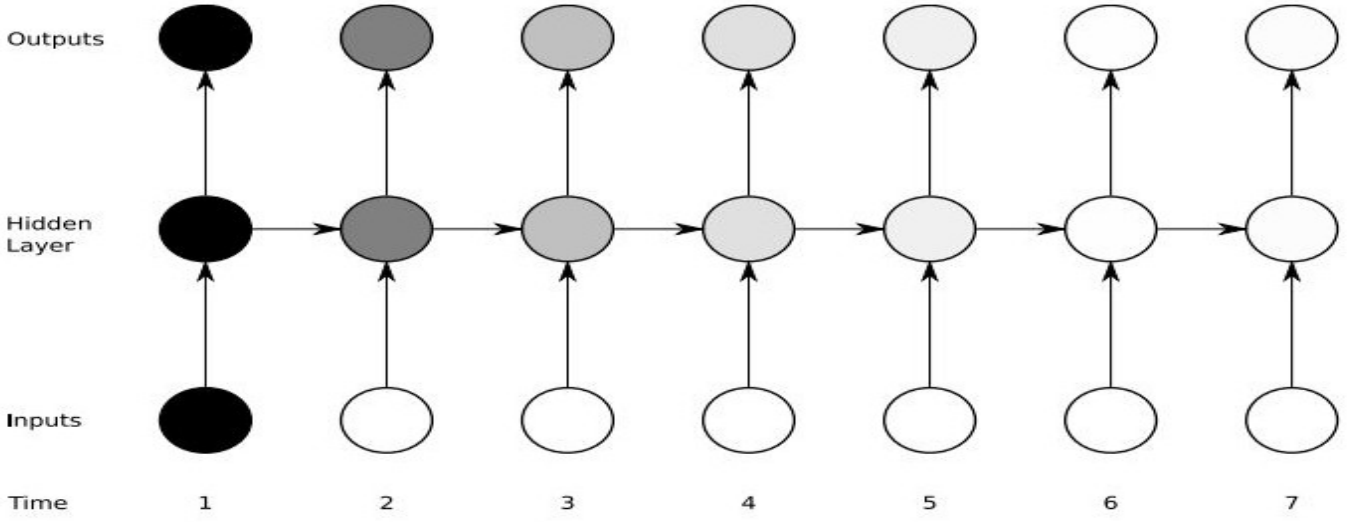


The problem of exploding or vanishing gradients

- Due to large number of layers
  - If the weights are small, then the gradients shrink exponentially
  - If the weights are big, then the gradients grow exponentially
- Typical feed-forward nets can cope with it due to few hidden layers

The problem of exploding or vanishing gradients

- Solution
  - Initialize the weights very carefully?
- RNNs have difficulty dealing with long-range dependencies



Today's Topics

- Modeling sequences: a brief overview
- Training RNNs with back propagation
- A toy example of training an RNN
- Why is it difficult to train RNN
- Long-term Short-term-memory

Long Short Term Memory

- Make the RNN out of little modules that are designed to remember values for a long time.