INFO 284 – Machine Learning

Recurrent Neural Networks

Marija Slavkovik & Bjørnar Tessem

UNIVERSITY OF BERGEN

Recurrent Neural Networks = RNN

- CNN exploits spatial locality of static images
  - Learns features to use in recognition
- Recurrent Neural Networks can learn temporal sequence during training.
  - Use to predict the next input in a sequence
  - Use to classify last input in a sequence
- RNN learns patters over time:
  - speech recognition,
  - continuous handwritten digit recognition,
  - text-to-speech
  - text-generation.

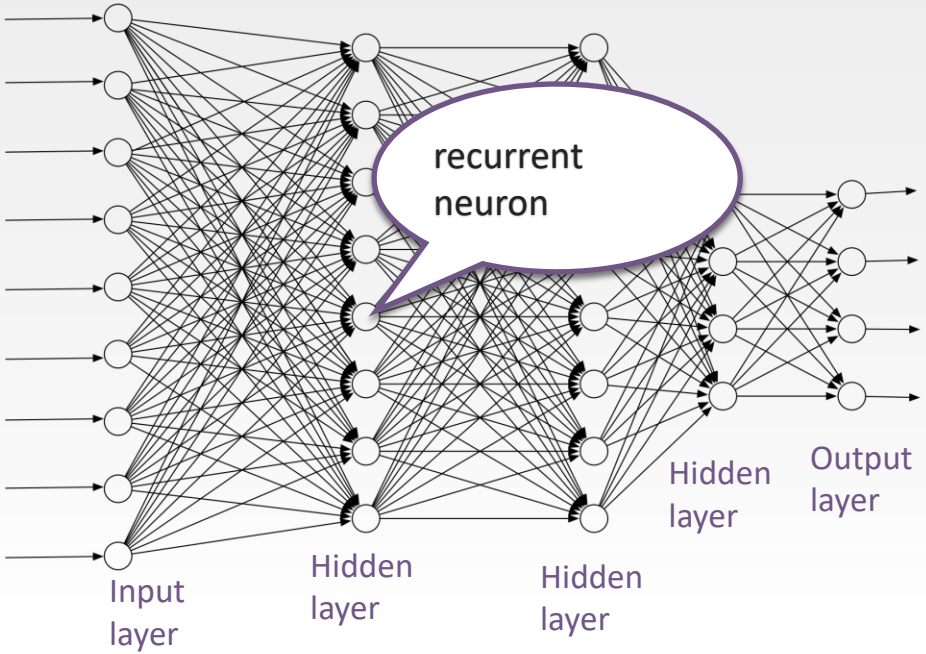
Recurrent network

- Recurrent network feeds its outputs back into its input. This allows it to exhibit dynamic temporal behaviour for a time sequence
- The response of the network to a given input depends on its state, which may depend on previous inputs.
    - These networks can support “short term memory”.
- Recurrent neural networks are applicable to sequence data
- Example: sentiment analysis

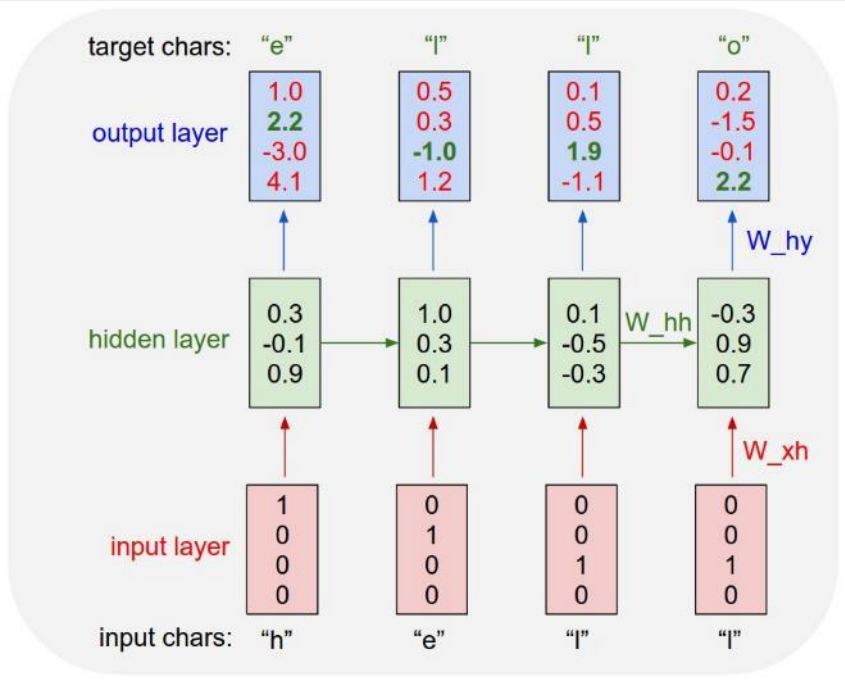
Machine learning class was bad, not ok at all

Machine learning class was ok, not bad at all

RNN Architecture

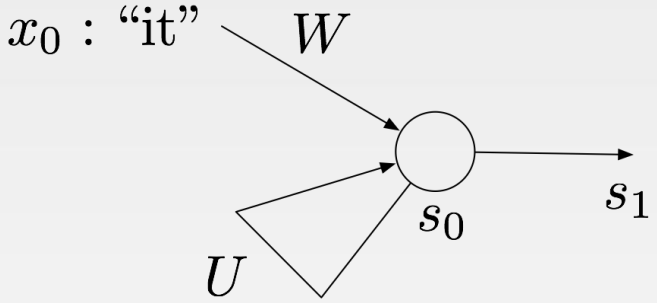


Recurrent Neurons



http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Remembering state

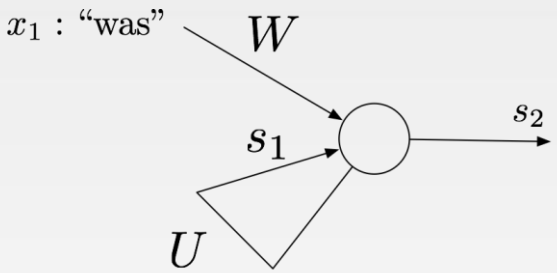


W, U : weight matrices $\quad t = 0$

x_0 : vector representing first word (one-hot)
S_0: neuron state at t = 0 (initial value)
S_1: neuron state at t = 1

$S_1 = \tanh(W^T x_0 + U^T s_0)$

Remembering state

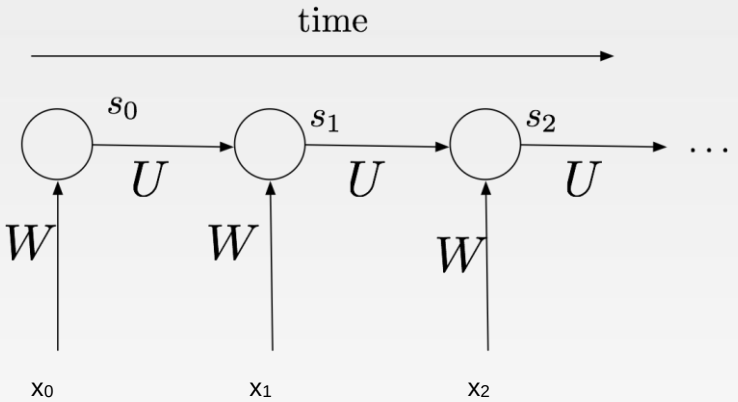


W, U : weight matrices $t = 1$

x_1 : vector representing second word (one-hot)
S_1: neuron state at t = 1
S_2: neuron state at t = 2

s_2 7 $S_2 = \tanh(W^T x_1 + U^T s_1)$

The recurrent neuron across time



- W,U stay the same throughout the sequence
- The state of the unit at any time contains information from previous states

In action

https://demo.allennlp.org

My dog is very ???

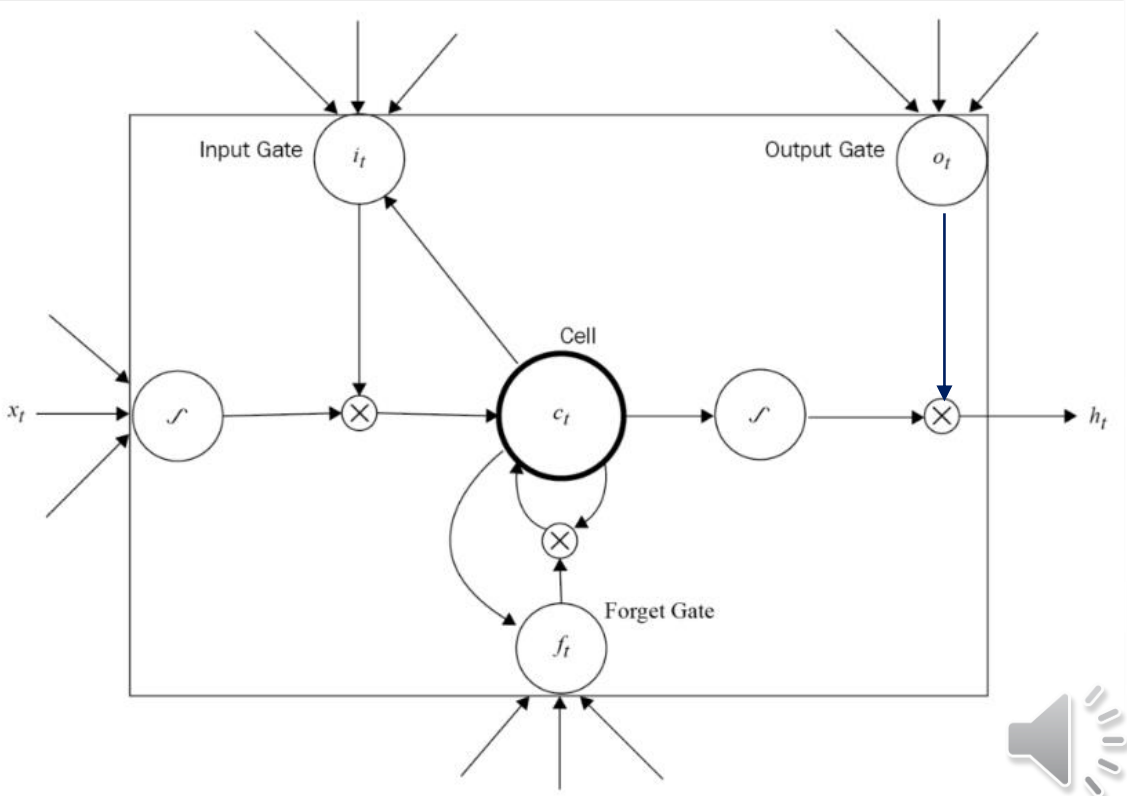
Prediction	Score
My dog is very **easy to pet and she …**	50.1%
My dog is very **homely and like to …**	34.6%
My dog is very **stubbornly stubborn, he …**	14.7%
My dog is very **guarded by others and will …**	0.4%
My dog is very **different", said the dog …**	0.3%

Training of RNN

- There is an error at each time step
- Total error is the sum of the errors at each time step and the gradient is the sum of the gradients at each time step
- RNNs are hard to train
    - vanishing gradient .. long time chains make the gradient too small which introduces bias towards 'closer' error
- Various solutions to the vanishing gradient. One is to change the neuron further into a **gated unit**

LSTM

- Long short-term memory
- Deals with exploding or vanishing gradient problems that made it hard to train RNNs.
- Applications:
  - Connectionist Temporal Classifications
  - Bioinformatics:
    - simultaneous alignment
    - recognition of sequences.



Neural Networks – a little summary

- Feed Forward Neural Networks has weights for each input and between every neurons of each adjacent layers.
- Convolutional Neural Networks and Recurrent Neural Network
  - Often viewed as a form parameter sharing
    - filters in convolutional neural networks
    - applying the same weights over and over again in RNN
- Combining them may be useful
- RNN has been the cutting edge method for sequential data
  - LSTM
  - Alternative: Attention mechanisms

Transfer Learning and Pretraining

- Neural networks weights? How are they initialized?
- Transfer learning –
  - Learning a new task by using what is learned on previous learned task.
  - Using Image Net trained model for medical imaging.
- Pretraining - training on unlabeled data.
  - BERT for pretraining language representation.
  - Use the Wikipedia corpus.
  - Hide some words in the corpus and train a model to predict them.

Input: the man went to the [MASK1] . he bought a [MASK2] of milk.
Labels: [MASK1] = store; [MASK2] = gallon

Sentence A: the man went to the store .
Sentence B: he bought a gallon of milk .
Label: IsNextSentence

Transfer Learning and Pretraining

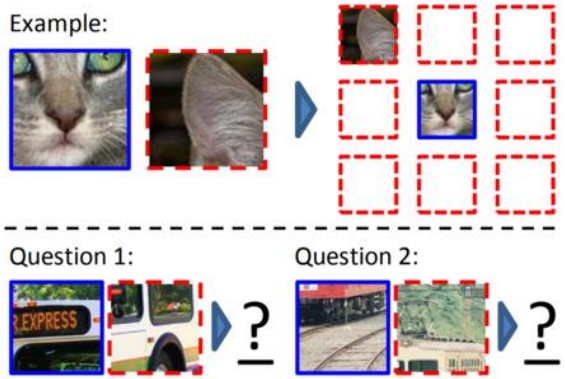
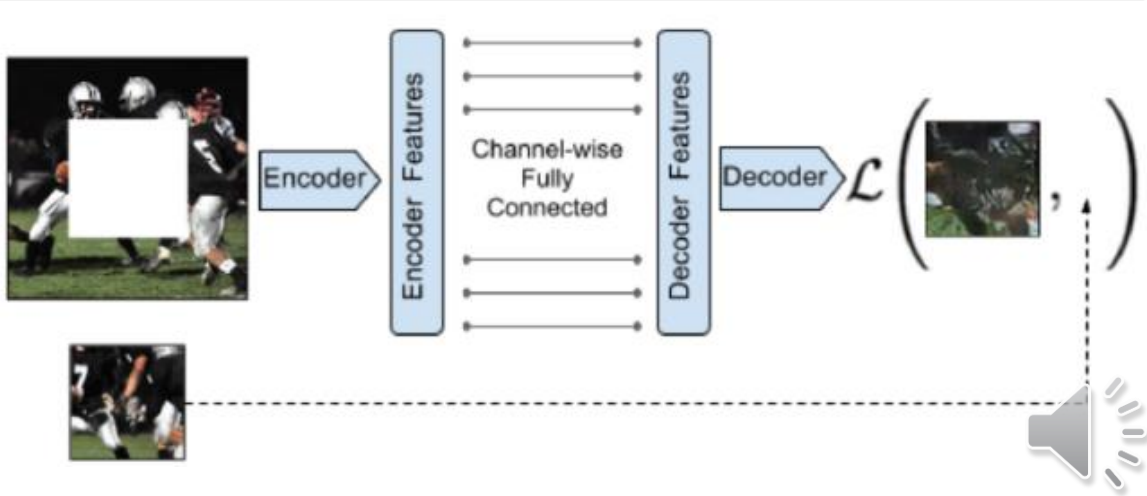


Figure 1. Our task for learning patch representations involves randomly sampling a patch (blue) and then one of eight possible neighbors (red). Can you guess the spatial configuration for the two pairs of patches? Note that the task is much easier once you have recognized the object!

Answer key: Q1: Bottom right Q2: Top center



uib.no