VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE | DEPARTMENT OF COMPUTER SCIENCE

Deep Learning

Recurrent Neural Networks

Jan Platoš, Radek Svoboda

May 6, 2024

Department of Computer Science
Faculty of Electrical Engineering and Computer Science
VŠB - Technical University of Ostrava

Recurrent Neural Networks

Recurrent Neural Networks

- How to process a sequence of events?

Recurrent Neural Networks

- How to process a sequence of events?
- How to process an input of variable length?

Recurrent Neural Networks

- How to process a sequence of events?
- How to process an input of variable length?
- Dense or Convolution networks are not well suited to this task.

Recurrent Neural Networks

- How to process a sequence of events?
- How to process an input of variable length?
- Dense or Convolution networks are not well suited to this task.
- We need an model that is able to deal with them...

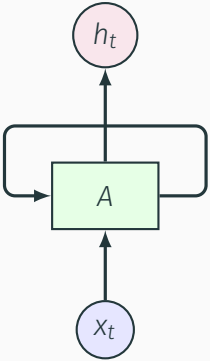
Recurrent Neural Networks

- How to define a model that works with a sequences?

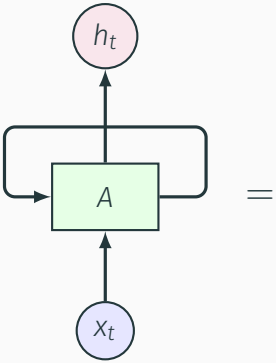
Recurrent Neural Networks

- How to define a model that works with a sequences?
- The main property is a **memory**.
- Normal neural network is memory-less.
- The memory allows the network to remember the past to be able to deal with the current situation.

Recurrent Neural Networks

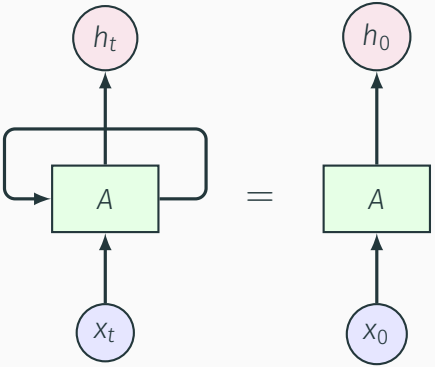


Recurrent Neural Networks

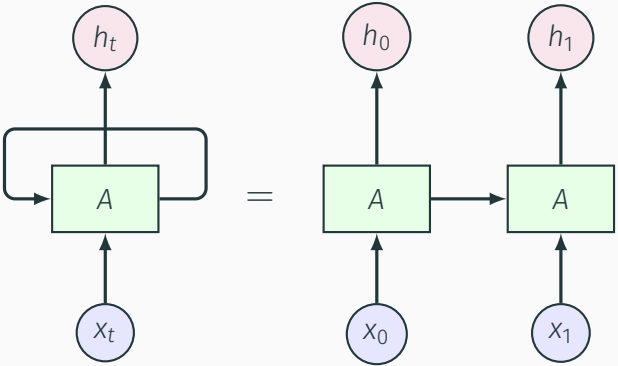


$=$

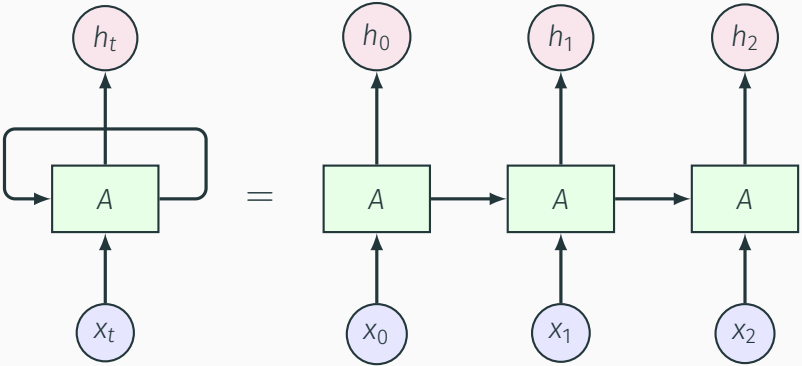
Recurrent Neural Networks



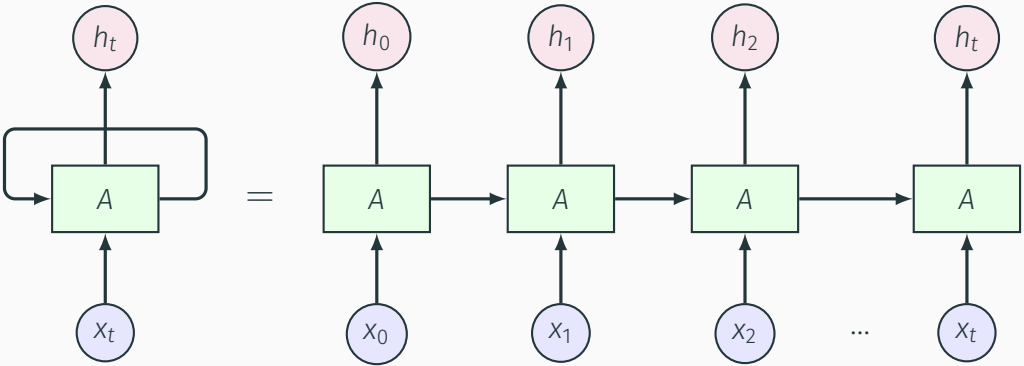
Recurrent Neural Networks



Recurrent Neural Networks



Recurrent Neural Networks



Recurrent Neural Networks

- The hidden state A is designed as h_t and its formula is as follows:

$$h_t = f(h_{t-1}, x_t)$$

- h_{t-1} is the previous state and x_t is the current input.
- The RNN neuron uses a non-linear activation functions.
- The Sigmoid, hyperbolic tangent, or ReLu is used.

Recurrent Neural Networks

- When an **tanh** activation is used then the formula is:

$$h_t = \tanh(W_h h_{t-1} + W_x x_t)$$

- W_h is the weight of the recurrent neuron and W_x is the weight of the input neuron.

$$y_t = W_y h_t$$

- w_y is the output weight and y_t it the current output.

Recurrent Neural Networks - Training phase

1. A single time step of the input is provided to the network.
2. The current state using set of current input and the previous state is computed.
3. One can go as many time steps according to the problem and join the information from all the previous states.
4. Once all the time steps are completed the final current state is used to calculate the output.
5. The output is then compared to the actual output i.e the target output and the error is generated.
6. The error is then back-propagated to the network to update the weights and hence the network (RNN) is trained.

Recurrent Neural Networks - Training phase

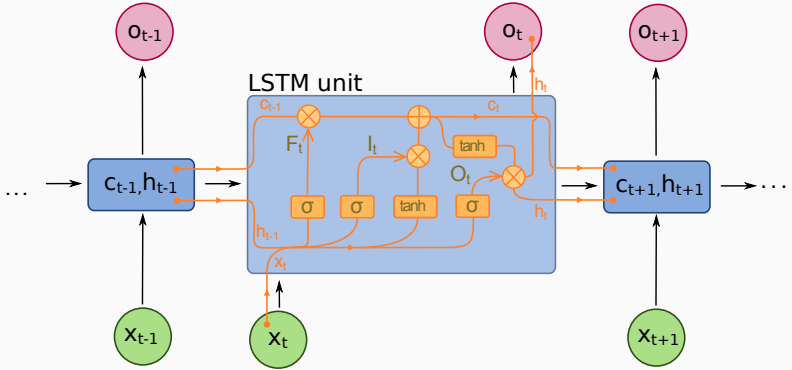
1. Weights update for each single time-step is very problematic and computation expensive.
2. We should evaluate the efficiency for the all time steps of a single input.
3. The process is called Back Propagation Through Time (BPTT).
4. For longer sequences it requires huge amount of memory and computations.
5. A Truncated version of BPTT is then used to split the time steps into chunks and approximate the final error.

Recurrent Neural Networks

1. Advantages of the RNNs:
   - An RNN remembers each and every information through time.
   - Recurrent neural network are even used with convolutional layers to extend the effective pixel neighborhood.
2. Disadvantages of the RNNs:
   - Gradient vanishing and exploding problems.
   - Training an RNN is a very difficult task.
   - It cannot process very long sequences if using *tanh* or *ReLU* as an activation function.

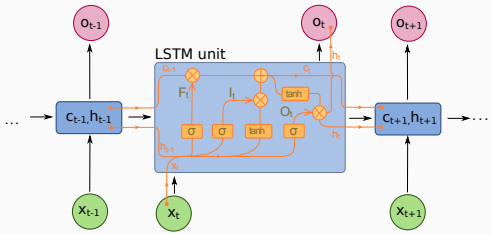
1. Multilayer RNNs are also possible.
2. The RNNs are able to deal with more than single input.
3. More complex architecture were designed to solve the disadvantages.

Long Short Term-Memory (LSTM)



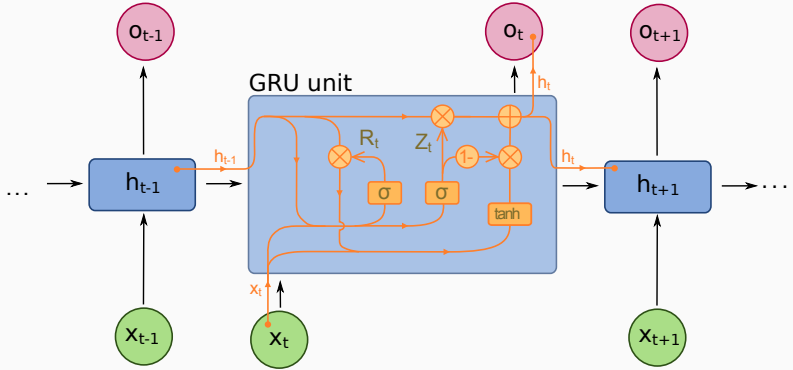
By fdeloche - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=60149410

Long Short Term-Memory (LSTM)



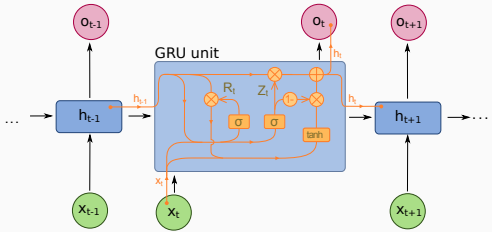
- Contains a set of recurrent connected subnets - memory blocks.
- Each block contains one or more self-connected memory cells and three multiplicative units - represents **write=input**, **read=output** and **reset=forget** operations.

Gate Recurrent Unit (GRU)



By fdeloche - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=60466441

Gate Recurrent Unit (GRU)



- Similar to LSTM.
- Merges cell state and hidden state into one state.
- Combines the **forget** and **input** gate into an **update** gate.
- Therefore, has less parameter and less complex structure.

Further reading

[1] Recurrent Neural Networks cheatsheet - Stanford

[2] Understanding RNN and LSTM - Towards Data Science

[3] Recurrent Neural Networks (RNN) with Keras

[4] Understanding LSTM Networks - Colah's blog

[5] Recurrent Neural Networks Tutorial, Part 1 – Introduction to RNNs - WildML

[6] Recurrent Neural Network - Toronto University

[7] Sequence Modeling: Recurrent and Recursive Nets - Deep Learning Book

[8] Recurrent neural network - Wikipedia

Questions?