Deep Learning in R with mxnet

Klaudius Kalcher

2016-11-22

Background: Neural Networks – Artificial Neurons

Idea

Arbitrarily complex computations based on a network of neurons of very limited computational power.

Background: Neural Networks – Artificial Neurons

Idea

Arbitrarily complex computations based on a network of neurons of very limited computational power.

Neuron

Takes multiple inputs and decides whether to fire or not (binary output).

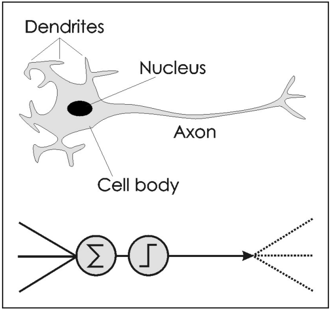
Background: Neural Networks – Artificial Neurons

Idea

Arbitrarily complex computations based on a network of neurons of very limited computational power.

Neuron

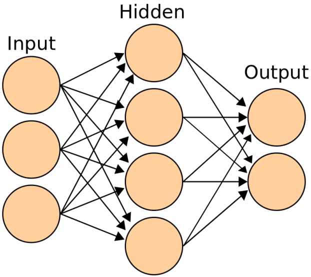
Takes multiple inputs and decides whether to fire or not (binary output).



Neural Networks – Networks of Neurons

Feedforward Network

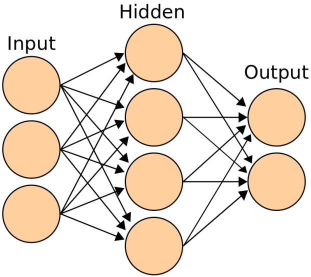
A *Feedforward Network* is a network of neurons arranged in layers from input to output.



Neural Networks – Networks of Neurons

Feedforward Network

A *Feedforward Network* is a network of neurons arranged in layers from input to output.



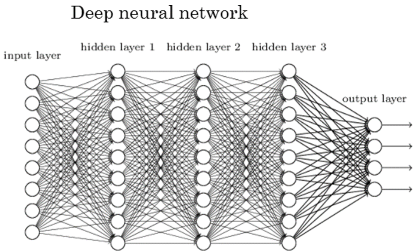
Neural Network with a single hidden layer

Theory: any (arbitrarily complex) function can be approximated by a neural network of a single hidden layer.

Deep Learning

Definition

Deep Learning designates the training of neural networks with more than one hidden layer.



Deep Learning

Why?

Neurons revisited

Neuron

- Sums up multiple inputs

Neurons revisited

Neuron

- Sums up multiple weighted inputs

Neurons revisited

Neuron

- Sums up multiple weighted inputs
- Applies a threshold function to them

Neurons revisited

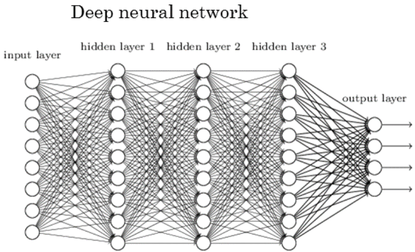
Neuron

- Sums up multiple weighted inputs
- Applies a nonlinear function to them

Neurons revisited

Neuron

- Sums up multiple weighted inputs
- Applies a nonlinear function to them

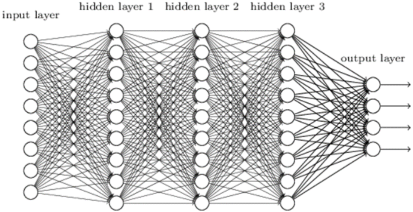


Neurons revisited

Neuron

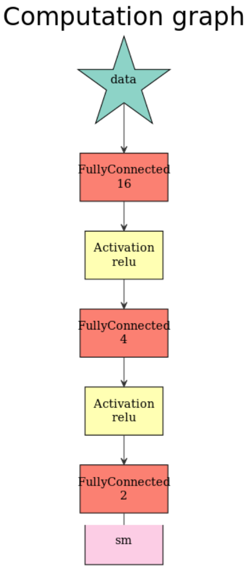
- Sums up multiple weighted inputs
- Applies a nonlinear function to them

Deep neural network

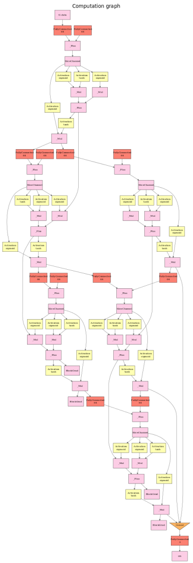


$$
\begin{aligned}
L_1 &= f(W_1 \cdot L_0) \\
L_2 &= f(W_2 \cdot L_1) \\
L_3 &= f(W_3 \cdot L_2) \\
O &= f(W_o \cdot L_3)
\end{aligned}
$$

Computation Graphs

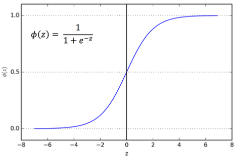


Computation Graphs (2)

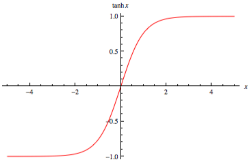


Nonlinearities

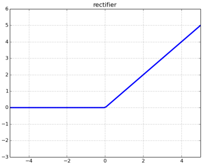
- Sigmoid



- tanh



- ReLU



Training of Deep Networks

Issues for training

- Define a loss function
- Compute loss and gradient of loss for training data
- Perform gradient descent

Frameworks

Deep Learning frameworks

- Tensorflow (Python)
- Caffe (C++, Python, MATLAB)
- Theano (Python)
- Torch (Lua, C)
- Deeplearning4j (Java, Scala, Clojure)
- MXNet (C++, Python, R, Julia, Matlab, Scala, Go)

Working in the MXNet Deep Learning Framework

- Define computation graph
- Define initializations
- Define input

  (some parameters are infered by the framework)

Working in the MXNet Deep Learning Framework

- Define computation graph
- Define initializations
- Define input

  (some parameters are infered by the framework)
- Run the gradient descent optimizer

  (adjust gradient descent hyperparameters, repeat)

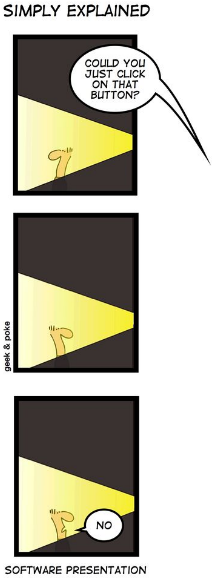
Working in the MXNet Deep Learning Framework

- Define computation graph
- Define initializations
- Define input

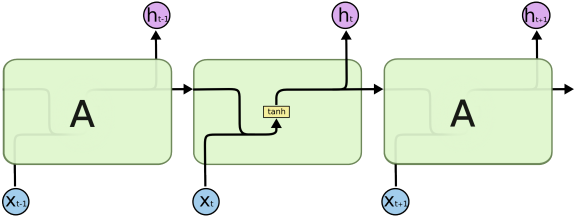
  (some parameters are infered by the framework)
- Run the gradient descent optimizer

  (adjust gradient descent hyperparameters, repeat)
- Evaluate model fit

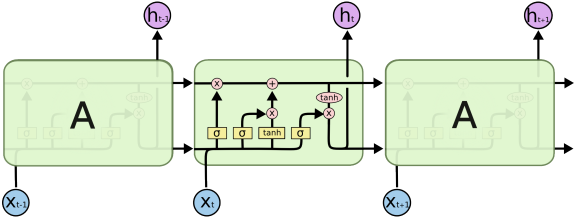
Let's see how this works. . .



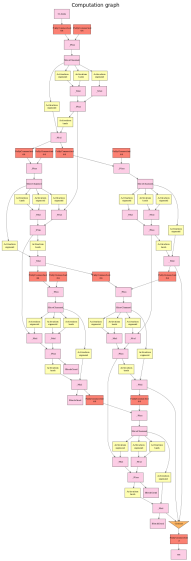
Recurrent Neural Networks



Recurrent Neural Networks - LSTM



LSTM with 2 layers. . .


Computation graph
t1.data
FullyConnected 64
FullyConnected 64
_Plus
SliceChannel
Activation sigmoid
Activation tanh
Activation sigmoid
_Mul
_Mul
Activation sigmoid
_Plus
Activation tanh
_Mul
BlockGrad
Concat
sm