STAT3612 Lecture10

Deep Neural Networks

Dr. Aijun Zhang

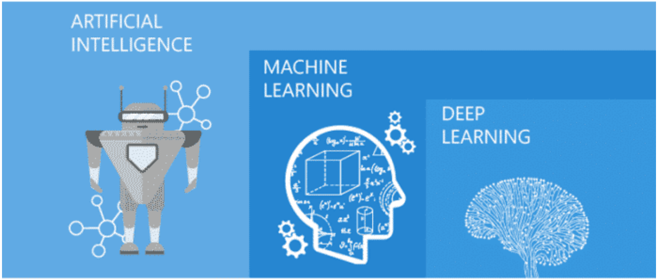
10 November 2020

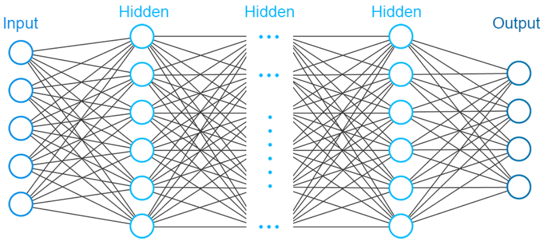
Department of 統計及精算學系
Statistics & Actuarial Science

Turing Award 2018

- News 27/3/2019: Fathers of the Deep Learning Revolution Receive ACM A.M. Turing Award (https://amturing.acm.org/)
- Deep Neural Networks → Major Breakthroughs in Artificial Intelligence

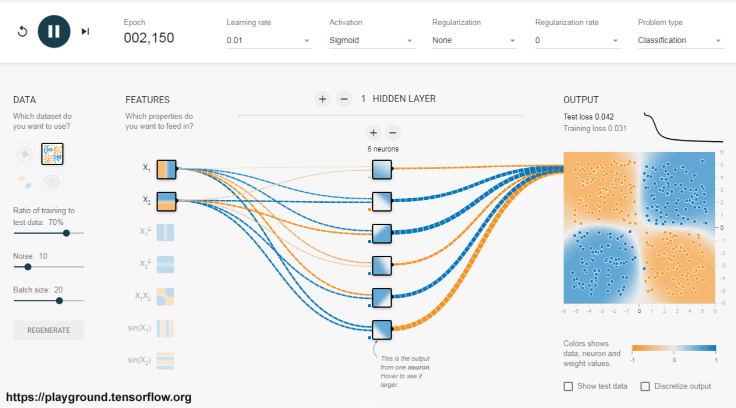
Deep Neural Networks



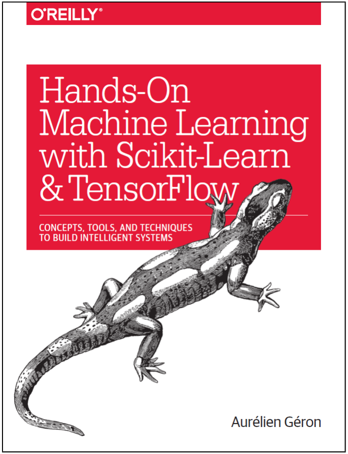


Interactive Visualization of Neural Networks

URL: https://playground.tensorflow.org/



Deep Learning with Python



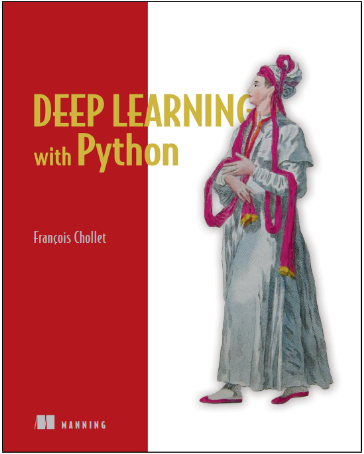
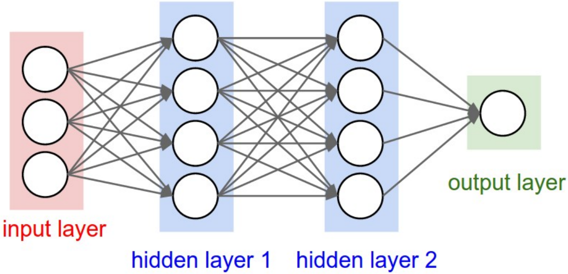


Table of Contents

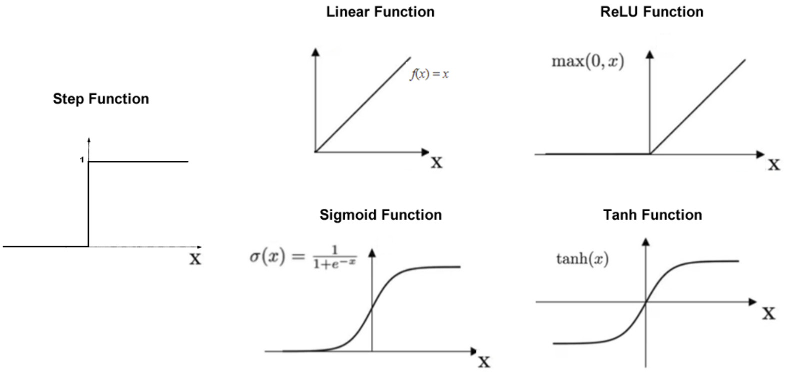
Feedforward Neural Networks

A feedforward neural network, also called a multi-layer perceptron (MLP), is an artificial neural network without cyclic connections.



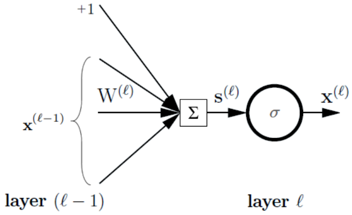
By the universal approximation theorem, a feedforward neural network with a single hidden layer containing a finite number of nodes can approximate any continuous function of n real variables with compact support.

Activation Functions



Forward Propagation

Let $\boldsymbol{W}^{(l)} = [w_{ij}]$ be the layer-l weight matrix of size $d^{(l-1)} \times d^{(l)}$, with $d^{(l)}$ denoting the number of nodes at layer l. Each node follows



The forward propagation from layer $l-1$ to layer l can be formulated as

$$s_j^{(l)} = w_{j0}^{(l)} + \sum_{i=1}^{d^{(l-1)}} w_{ji}^{(l)} x_i^{(l-1)}$$

$$x_j^{(l)} = \sigma(s_j^{(l)}), \quad j = 1, \ldots, d^{(l)}$$

Table of Contents

Network Training Techniques

- Define the training error (e.g. MSE for regression problem) as

$$E(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}(h(\mathbf{x}_i; \mathbf{w}) - y_i)^2$$

- Training with stochastic gradient descent

$$\mathbf{w}(t+1) = \mathbf{w}(t+1) - \eta \nabla E(\mathbf{w}(t))$$

  where the gradient $\nabla E(\mathbf{w}(t))$ is evaluated the backpropagation algorithm or automatic differentiation

- Other techniques: Momentum method, Regularization (ℓ_1 or ℓ_2), Dropout, Batch normalization, Weight initialization

Backpropagation Algorithm

- To compute $\nabla E(\mathbf{w}(t))$, let's introduce for each layer the sensitivity vector $\boldsymbol{\delta}^{(l)}$ to measure how sensitive the residual term $(h(x;w)-y)^2$ changes with input signal $\boldsymbol{s}^{(l)}$, i.e.,

$$\boldsymbol{\delta}^{(l)} = \frac{\partial (h(x;w)-y)^2}{\partial \boldsymbol{s}^{(l)}}$$

- The backpropagation step for the sensitivity vector can be then derived as

$$\boldsymbol{\delta}^{(l)} = \sigma'\left(\boldsymbol{s}^{(l)}\right) \otimes \left[\boldsymbol{W}^{(l+1)}\boldsymbol{\delta}^{(l+1)}\right]_1^{d^{(l)}},$$

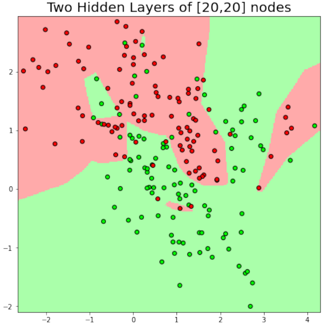
which gives an updated formula from layer $l+1$ to layer l (backward).

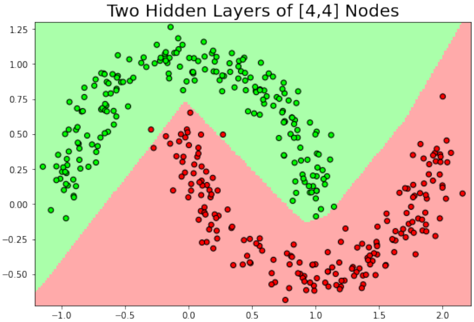
- The layer-wise gradient can be computed by

$$\frac{\partial E(\mathbf{w})}{\partial \mathbf{W}^{(l)}} = \frac{1}{n}\sum_{i=1}^{n}\frac{\partial (h(x;w)-y)^2}{\partial \boldsymbol{W}^{(l)}} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}^{(l-1)}\left(\boldsymbol{\delta}^{(l)}\right)^T.$$

Table of Contents

DNN Classification with Two Examples



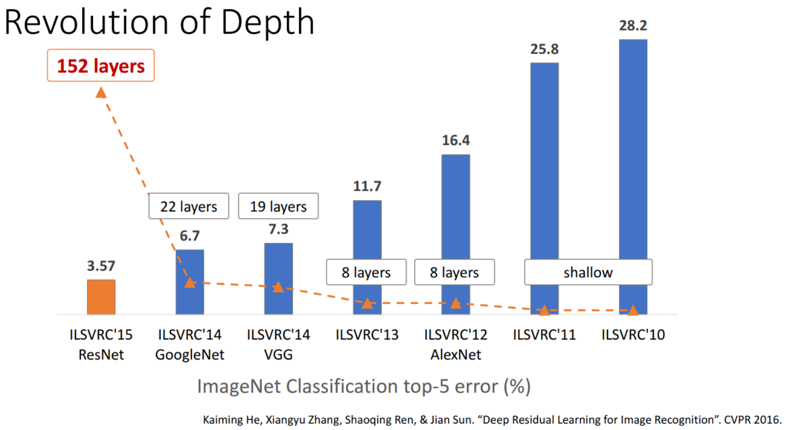


Try it out with the provided Python codes based on Scikit-Learn MLP ...

Table of Contents

ImageNet Challenge

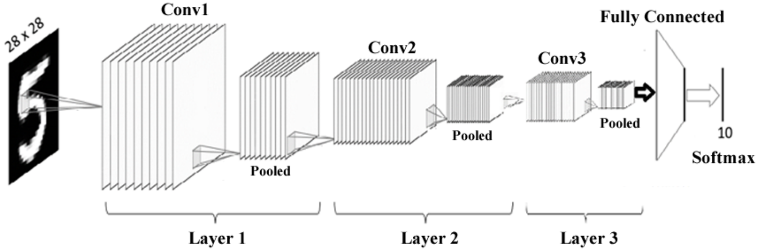
Problem Background: http://www.image-net.org/



Source: CVPR 2017 Tutorial by Kaiming He (FAIR)

Convolutional Neural Networks

MNIST Handwritten Digit Recognition:



Try it out with the provided Python codes based on TensorFlow Keras ...

Thank You!

Q&A or Email ajzhang@umich.edu