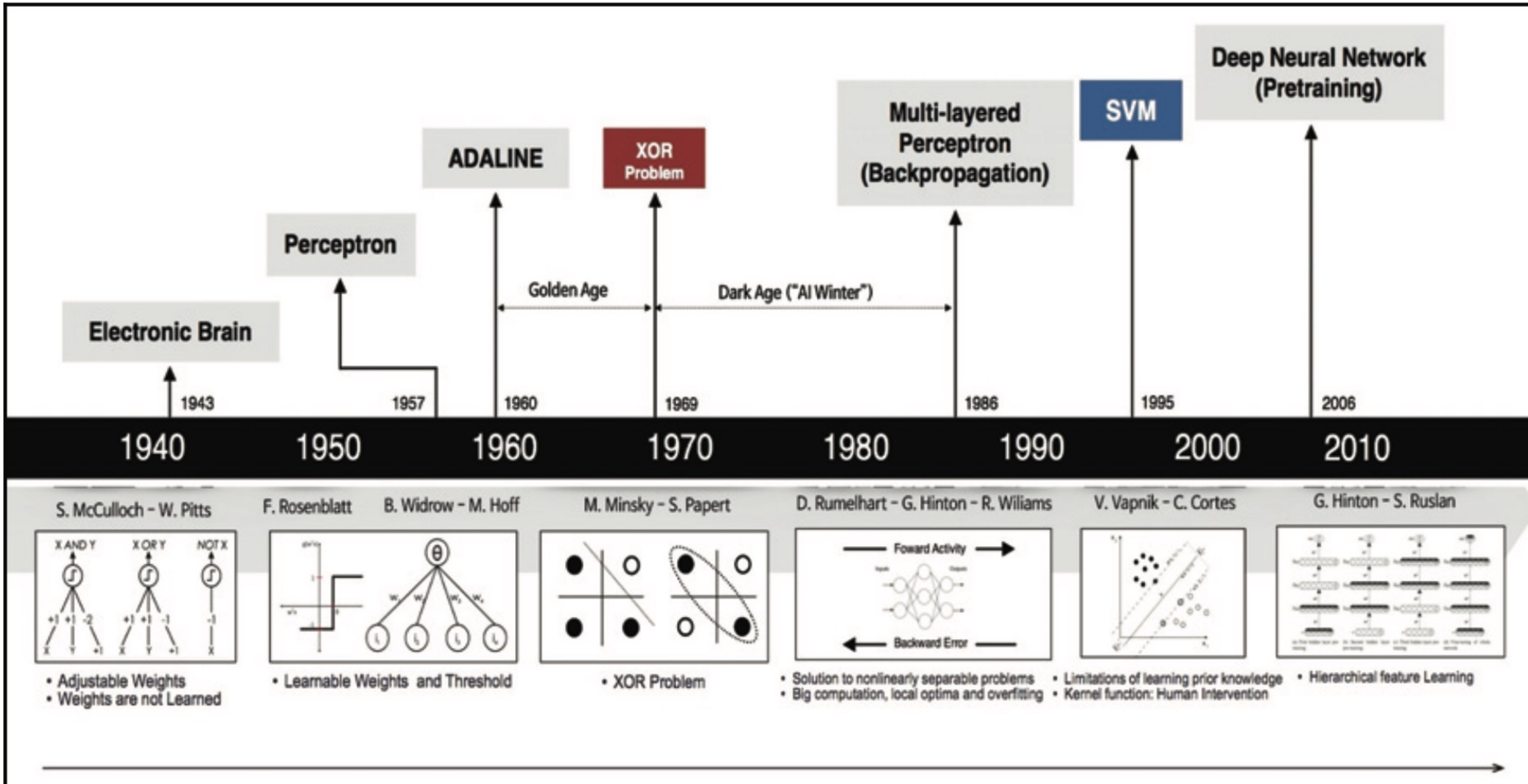
https://www.sli.do/
#073374

Fundamentals of Deep Learning (II)

Learning Objectives

- Learn the multi-layered neural network (MLP, or dense neural network DNN)
- Learn how to use Keras to solve MLP classification and regression problems.

Brief History of Neural Networks



Brief history of neural network

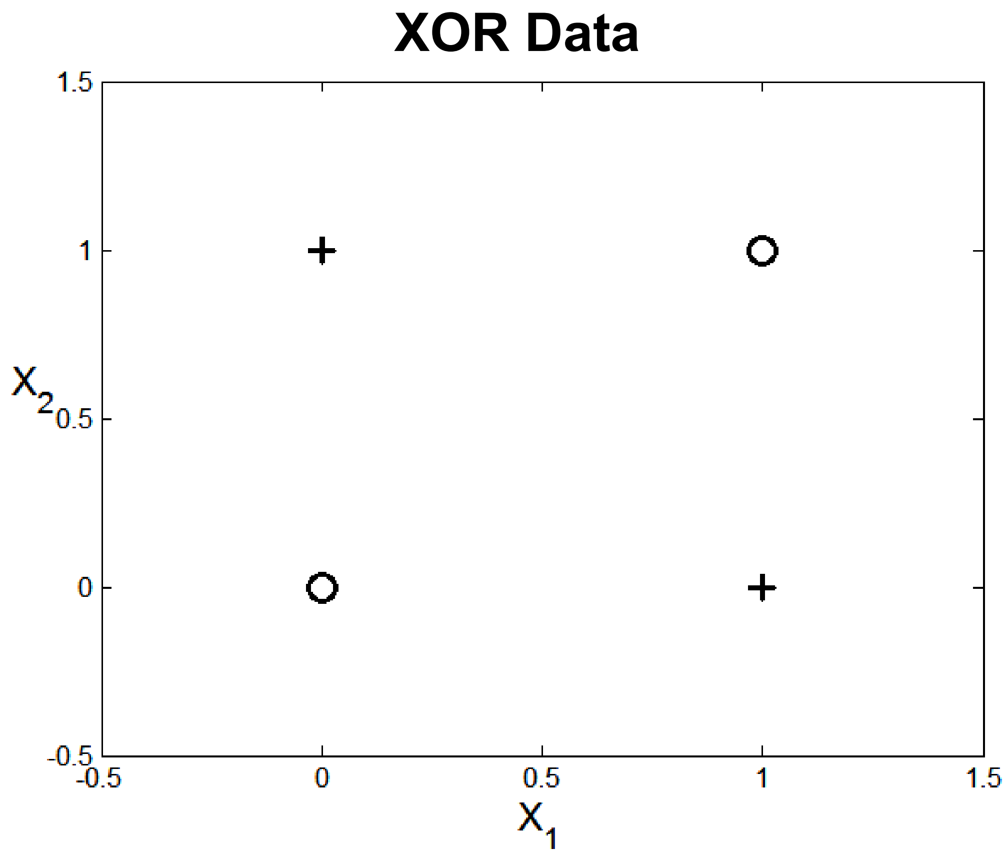
ANN (Artificial Neural Network): Multi-layer Neural Network

Nonlinearly Separable Data

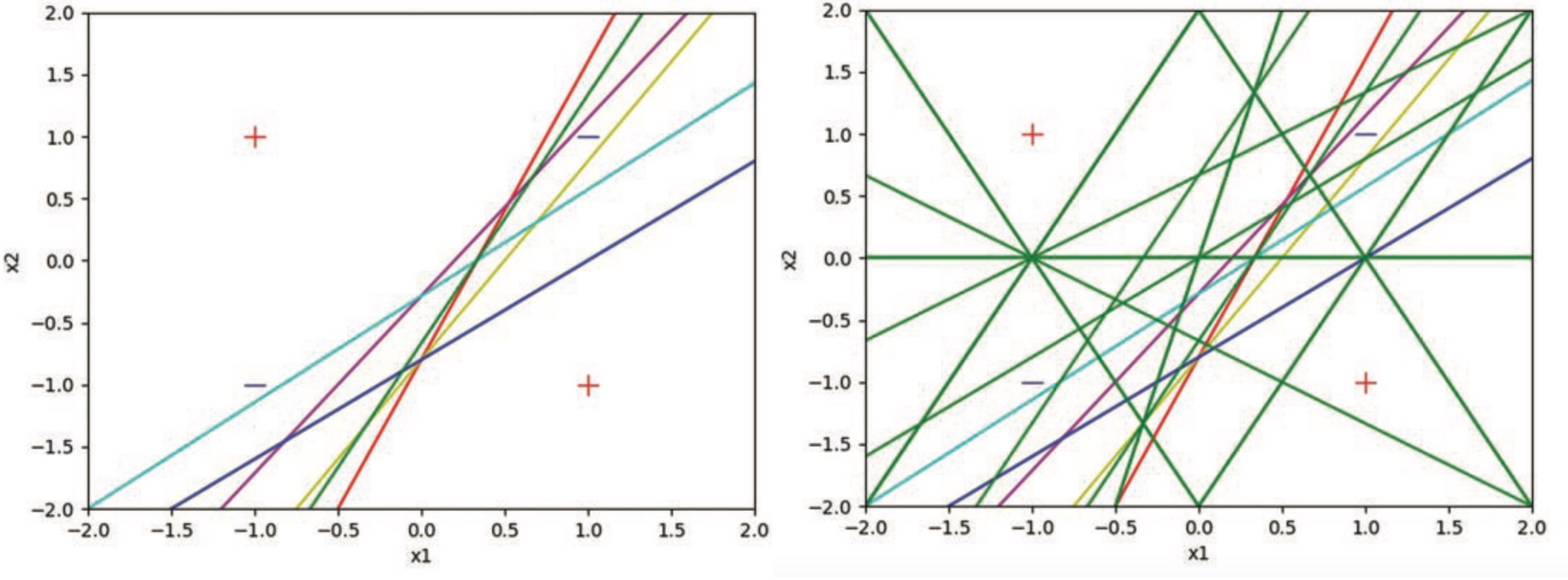
For nonlinearly separable problems, perceptron learning algorithm will fail because no linear hyperplane can separate the data perfectly

$$y = x_1 \oplus x_2$$

x_1	x_2	y
0	0	-1
1	0	1
0	1	1
1	1	-1

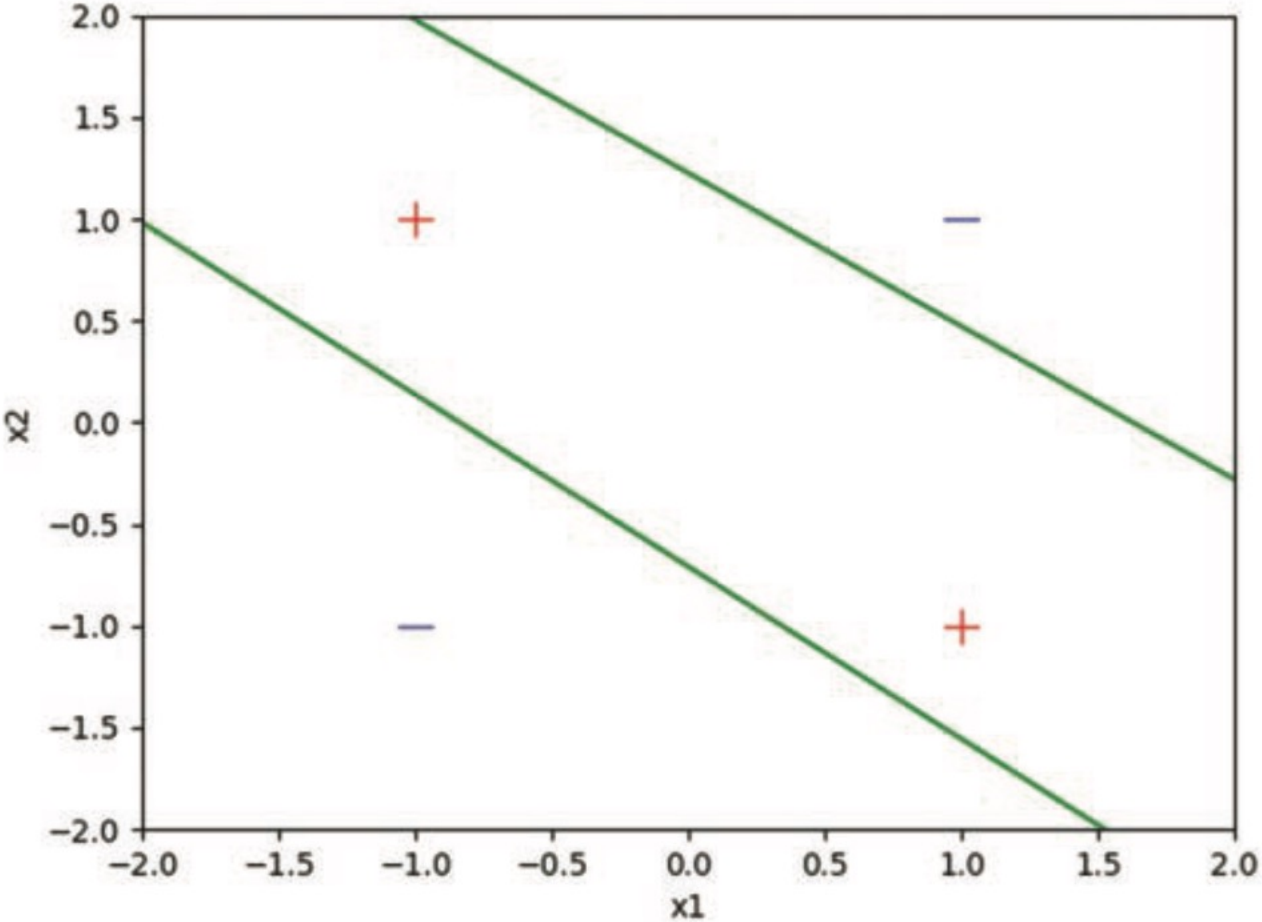


Nonlinearly Separable Data: XOR Problem



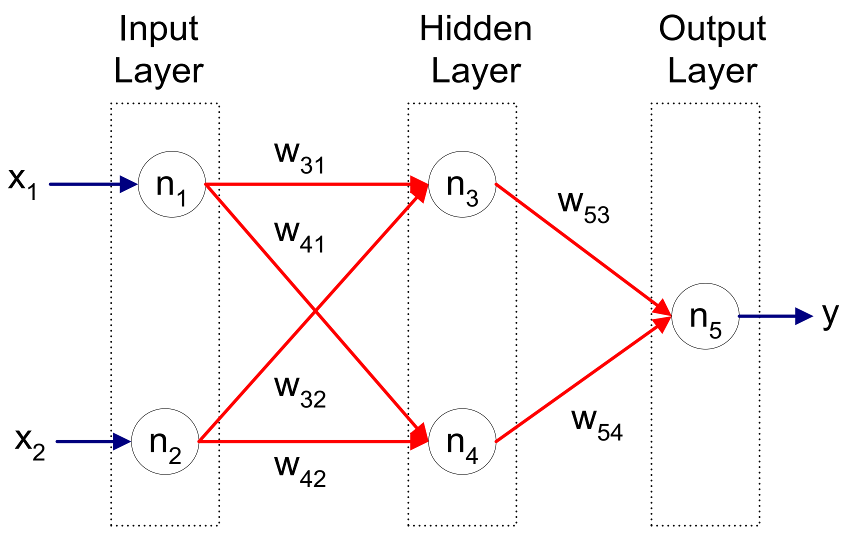
Perceptron attempting to learn XOR. Left: After 6 weight adjustments.
Right: After 30 weight adjustments.

Combining Two Perceptrons: XOR Problem

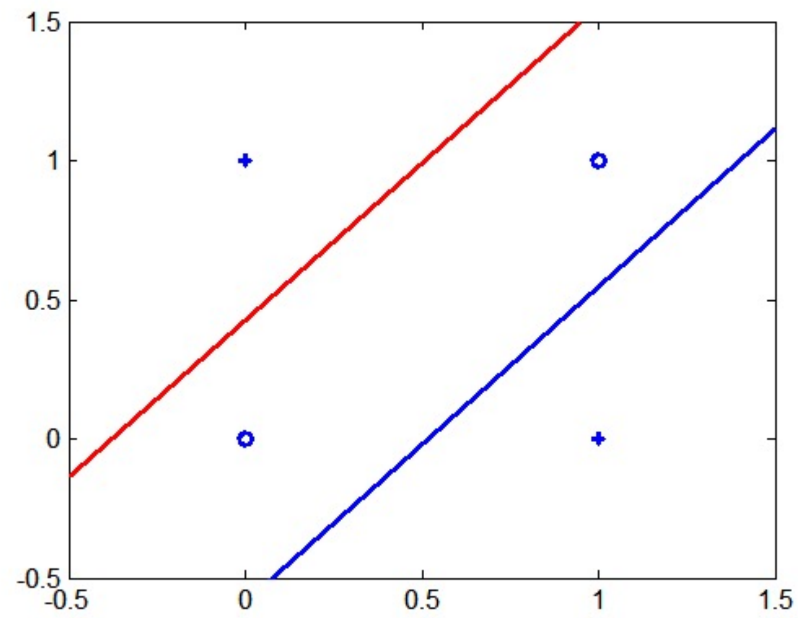


Multi-layer Neural Network

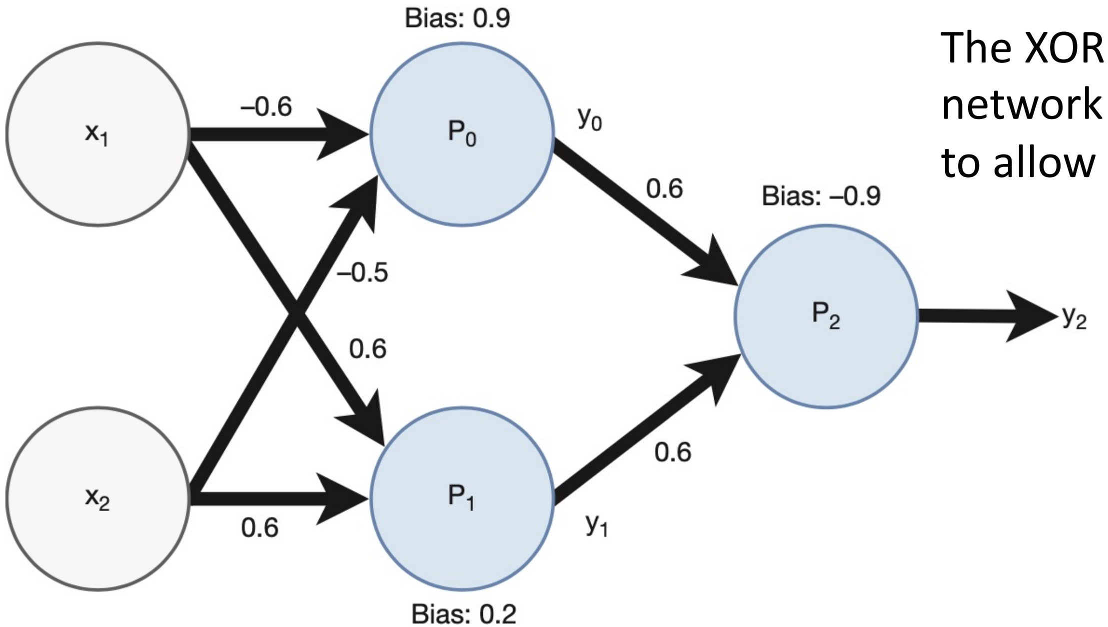
- The XOR can be overcome by allowing our network to have more than one layer, that is, to allow it to contain hidden units.



XOR Data



The XOR can be overcome by allowing our network to have more than one layer, that is, to allow it to contain hidden units.



X_0	X_1	X_2	Y_0	Y_1	Y_2
1	–1 (False)	–1 (False)	1.0	–1.0	–1.0 (False)
1	1 (True)	–1 (False)	1.0	1.0	1.0 (True)
1	–1 (False)	1 (True)	1.0	1.0	1.0 (True)
1	1 (True)	1 (True)	–1.0	1.0	–1.0 (False)

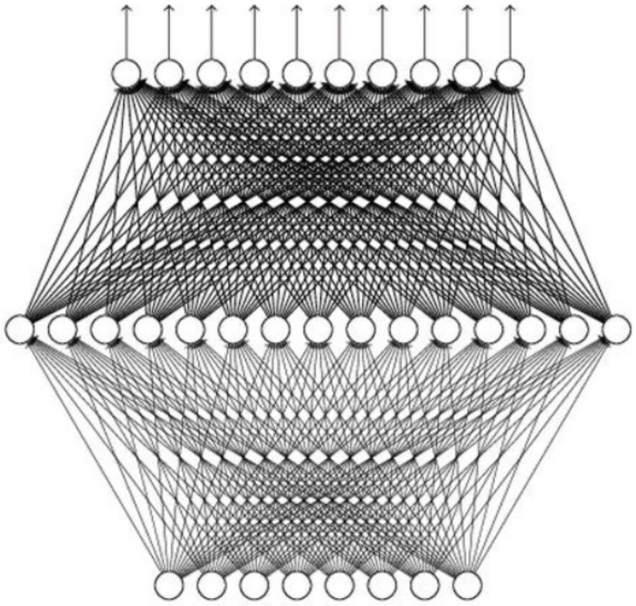
Universality Theorem

Any continuous function f

$$f : R^N \rightarrow R^M$$

Can be realized by a network with one hidden layer

(given **enough** hidden neurons)

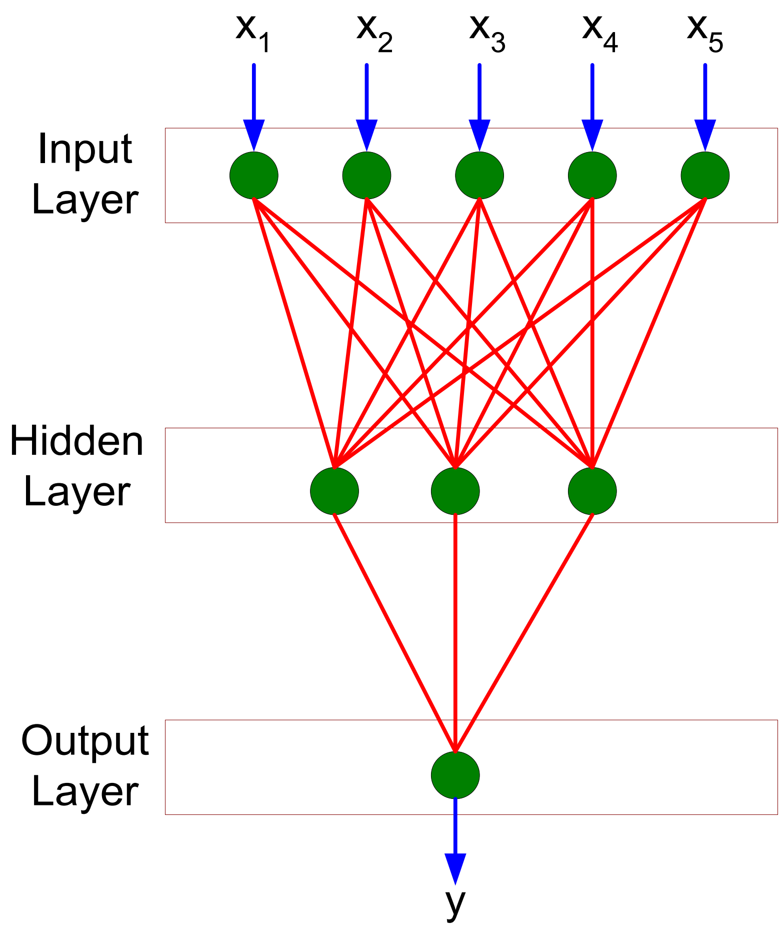


Reference for the reason: http://neuralnetworksanddeeplearning.com/chap4.html

Why "Deep" neural network not "Fat" neural network?

credit: Hung-Yi Lee, http://speech.ee.ntu.edu.tw/~tlkagk/courses_ML17_2.html

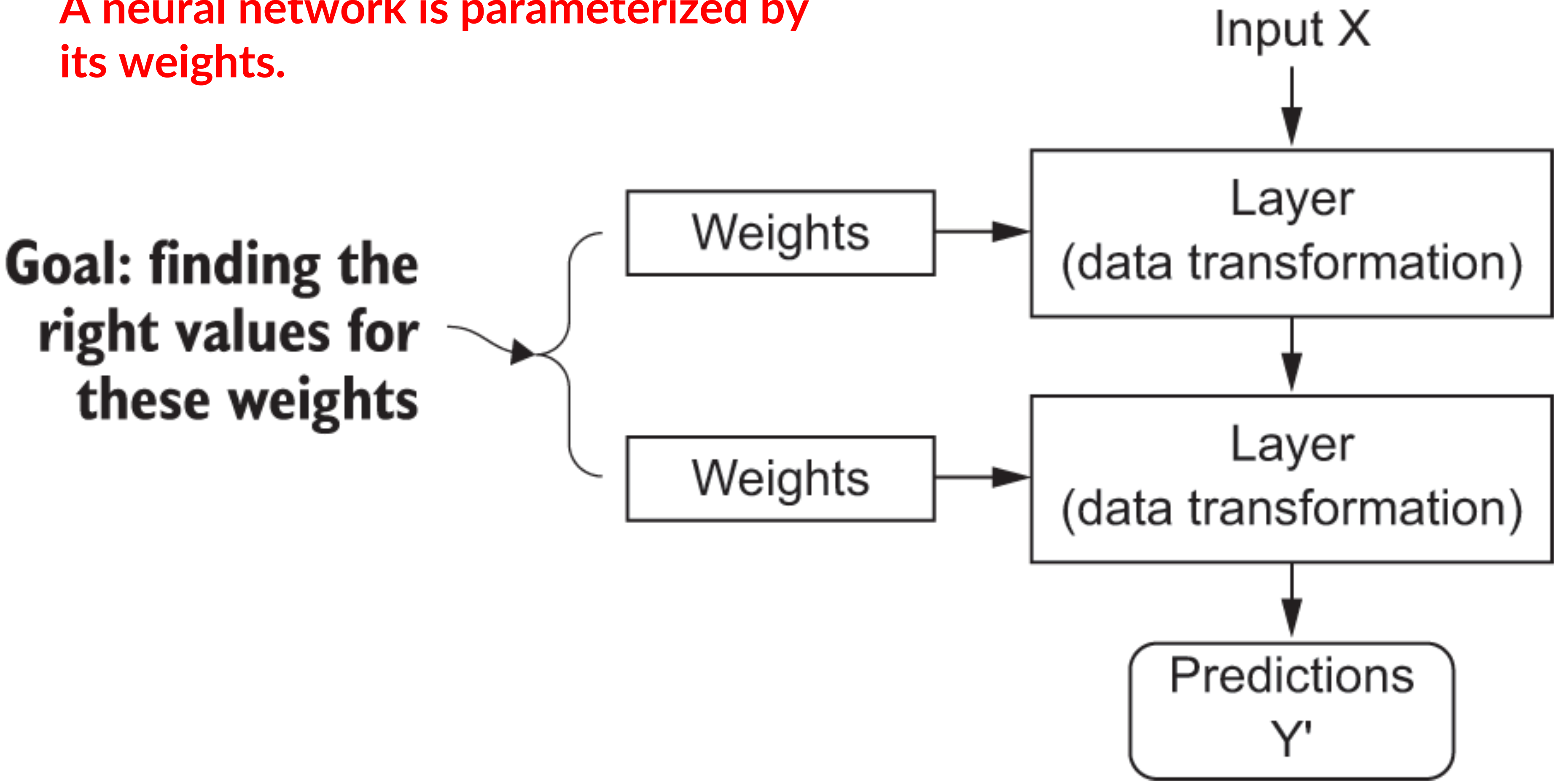
Multi-layer Neural Network



- More than one *hidden layer* of computing nodes
- Every node in a hidden layer operates on activations from preceding layer and transmits activations forward to nodes of next layer
- Also referred to as "feedforward neural networks" or "dense neural networks"

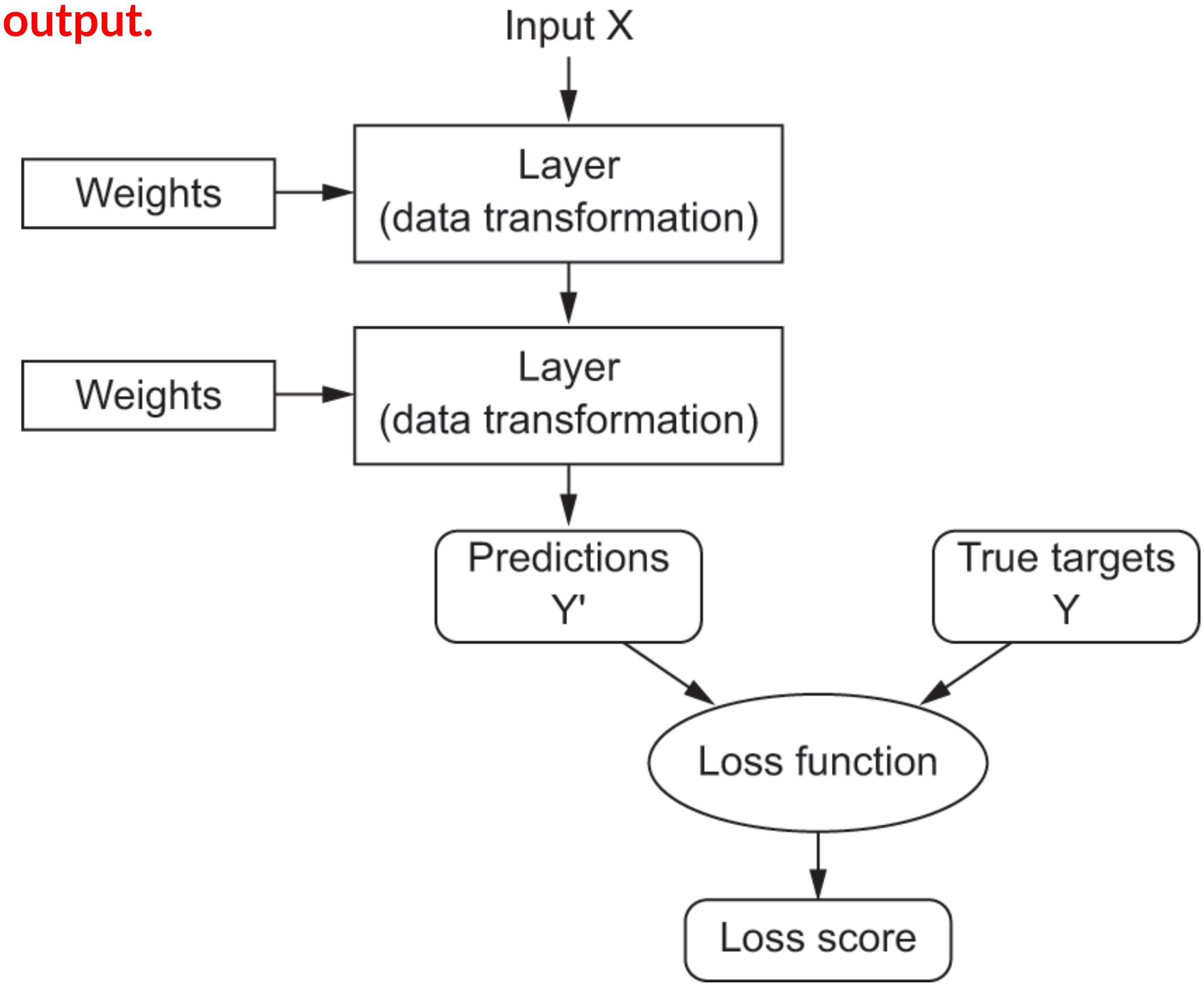
Understand How Neural Network Works in Three Figures (1/3)

A neural network is parameterized by its weights.



Understand How Neural Network Works in Three Figures (2/3)

A loss function measures the quality of the network's output.

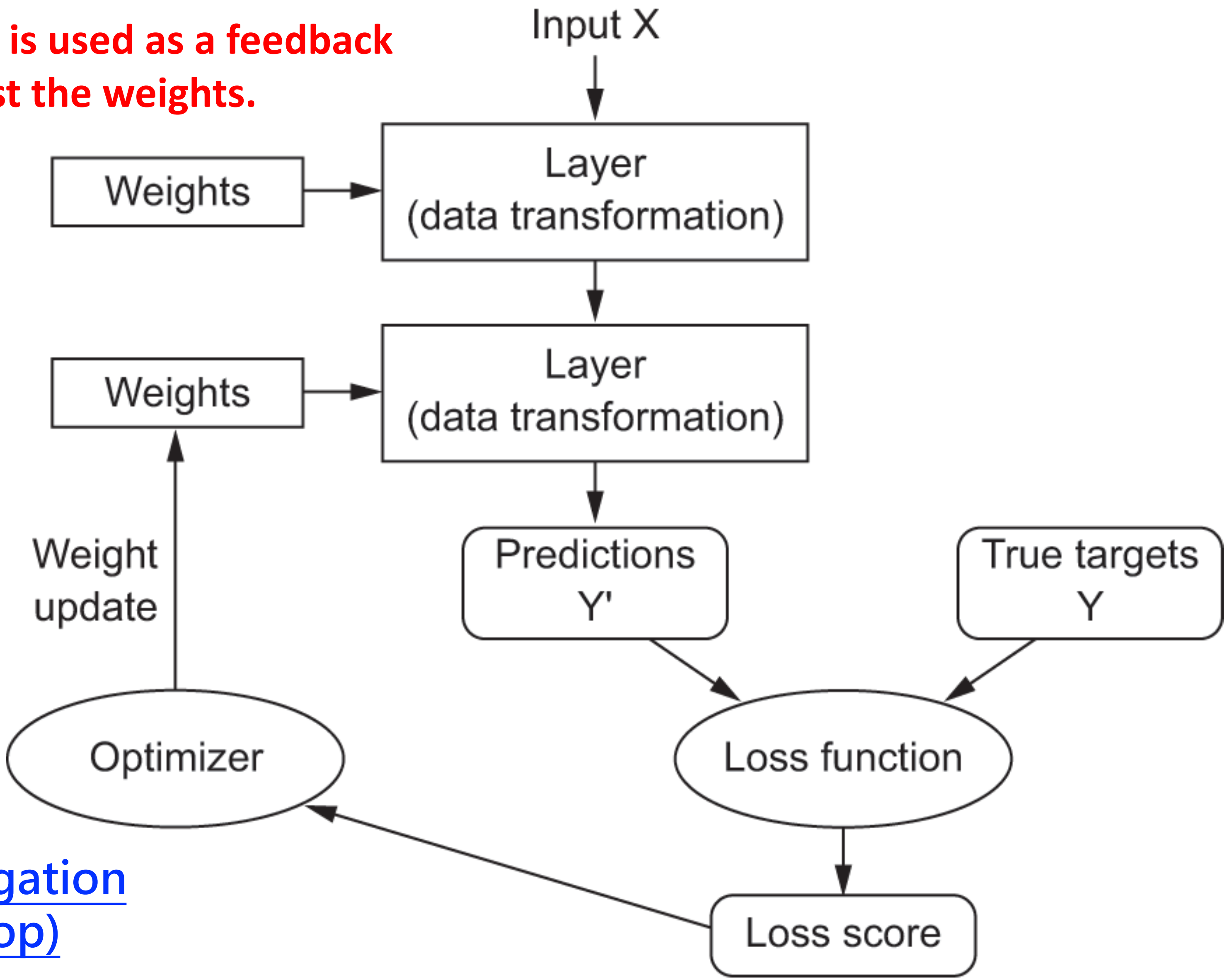


Neural networks: forward propagation

- The phrase “theoretical minimum” is taken from a very successful book series written by Leonard Susskind, a great physicist at Stanford University.
- “Theoretical minimum” means just the minimum theories and equations you need to know in order to proceed to the next level.
- See NN_forward.pdf

Understand How Neural Network Works in Three Figures (3/3)

The loss score is used as a feedback signal to adjust the weights.



Backpropagation
(Backprop)

Neural networks: backprop

- The phrase "theoretical minimum" is taken from a very successful book series written by Leonard Susskind, a great physicist at Stanford University.

- "Theoretical minimum" means just the minimum theories and equations you need to know in order to proceed to the next level.

- See NN_backprop.pdf

Summary of learning methods for neural networks

- For small datasets (e.g. 10,000 cases) or bigger datasets without much redundancy, use a full-batch method.
  - Conjugate gradient, LBFGS ...
  - adaptive learning rates, rprop ...
- For big, redundant datasets use mini-batches.
  - Try gradient descent with momentum.
  - Try rmsprop (with momentum)
  - Try LeCun's latest recipe.

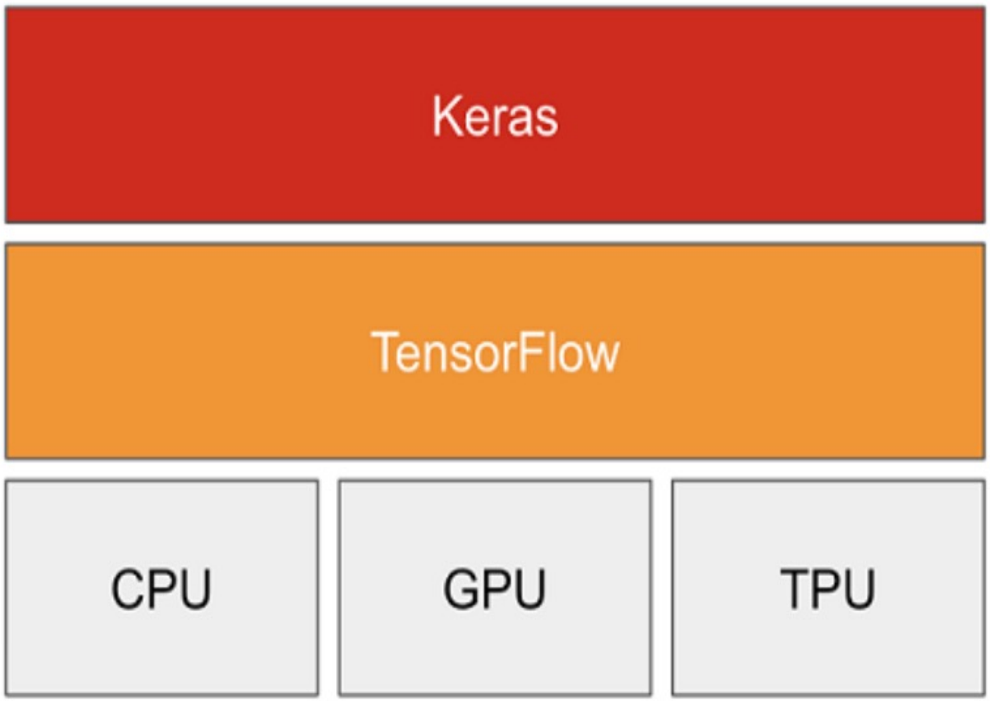
- Why there is no simple recipe:

  Neural nets differ a lot:
  - Very deep nets (especially ones with narrow bottlenecks).
  - Recurrent nets.
  - Wide shallow nets.

  Tasks differ a lot:
  - Some require very accurate weights, some don't.
  - Some have many very rare cases (e.g. words).

credit: G. Hinton, 2012 COURSERA COURSE LECTURES: Neural Networks for Machine Learning
https://www.cs.toronto.edu/~hinton/coursera_lectures.html

Neural Networks with Keras and TensorFlow



Deep learning development: layers, models, optimizers, losses, metrics...

Tensor manipulation infrastructure: tensors, variables, automatic differentiation, distribution...

Hardware: execution

See NN_Keras.pdf