Introduction to Deep Learning

ROHIT PADEBETTU

Course Assignments

Programming Assignments

Reading Assignments

Presentation Assignments

Technical Skills Assignments

Writing Assignments

Technical Assignment

Complete & Submit Code on GitHub for Mushroom Classification

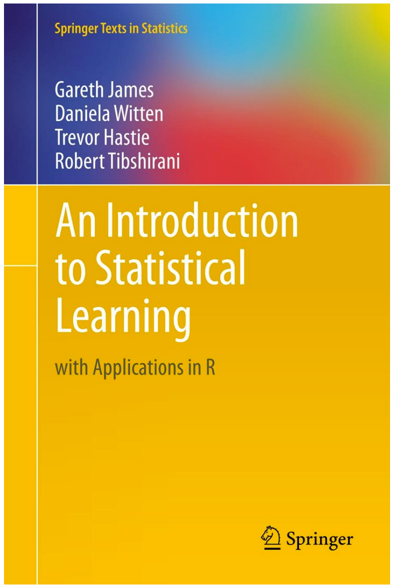
Programming Assignment

Install & Complete: Swirl – Regression Models

Install & Complete: Swirl – Getting & Cleaning Data

Reading Assignment

Read Chapter 4: Classification
Read Chapter 8: Tree Based Methods



Writing Assignment

No Writing Assignment this week !!!

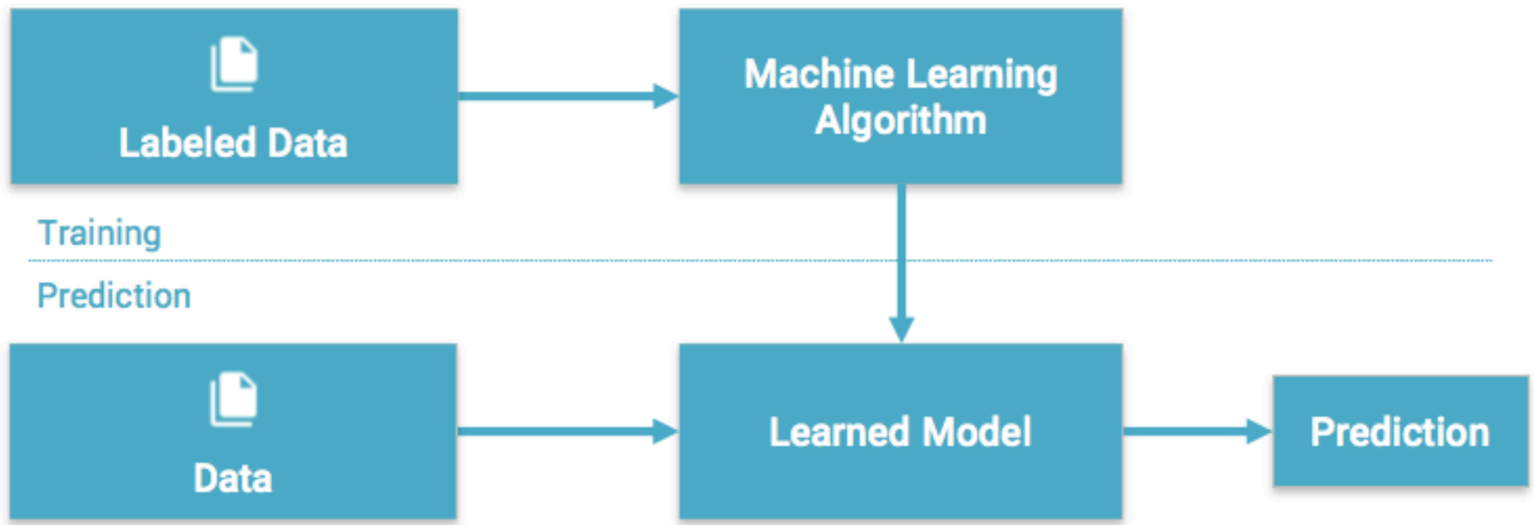
Presentation Assignment

By Saturday Submit

Your Presentations on Mushroom Classification Case

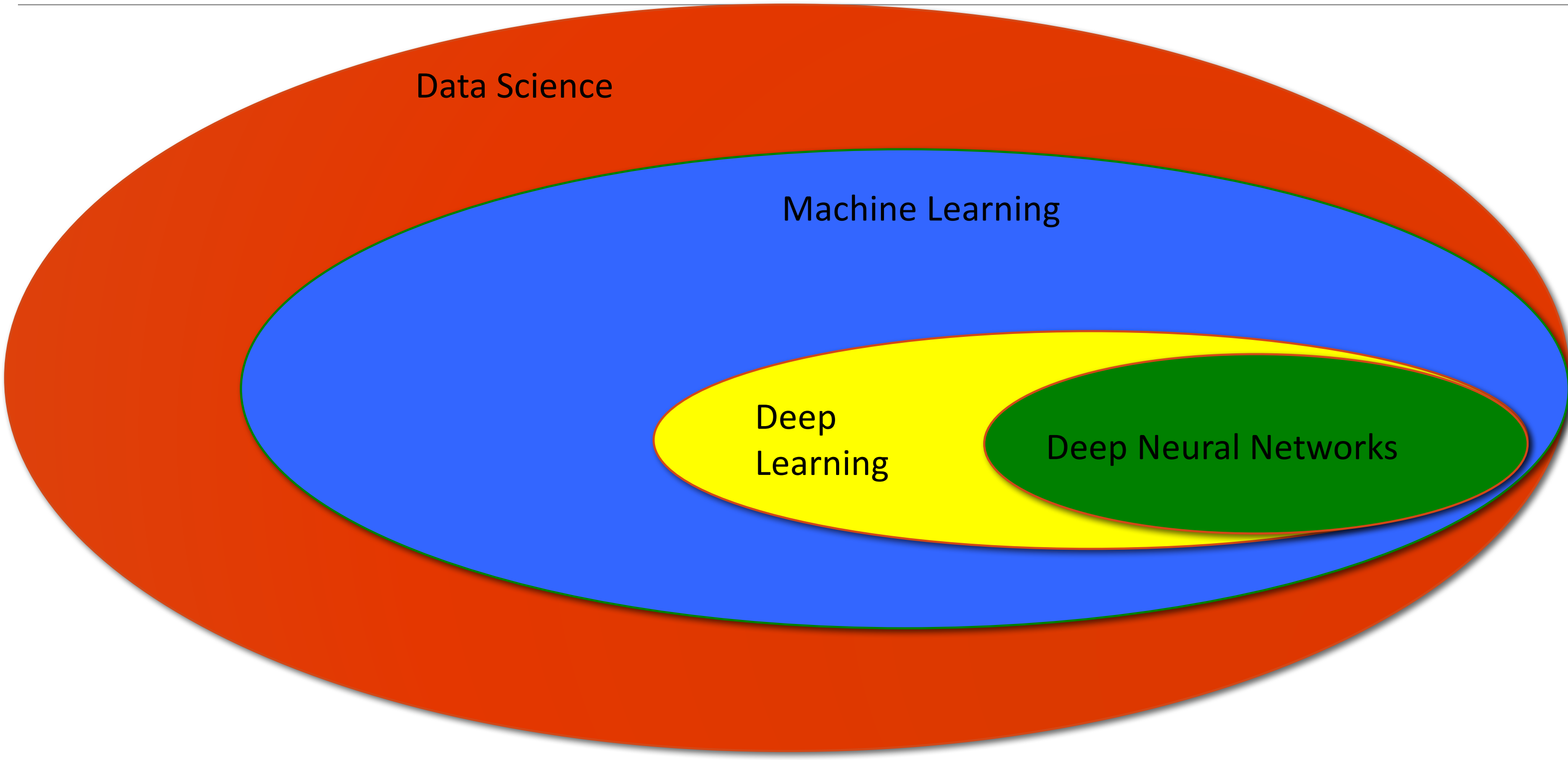
1. Technical Presentation
2. Business Presentation (Not to exceed 5 slides)

Machine Learning



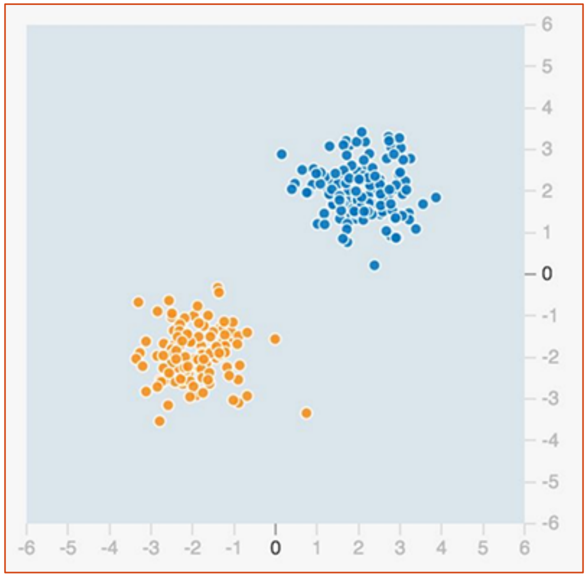
Provides various techniques that can learn from and make predictions on data

Where we are

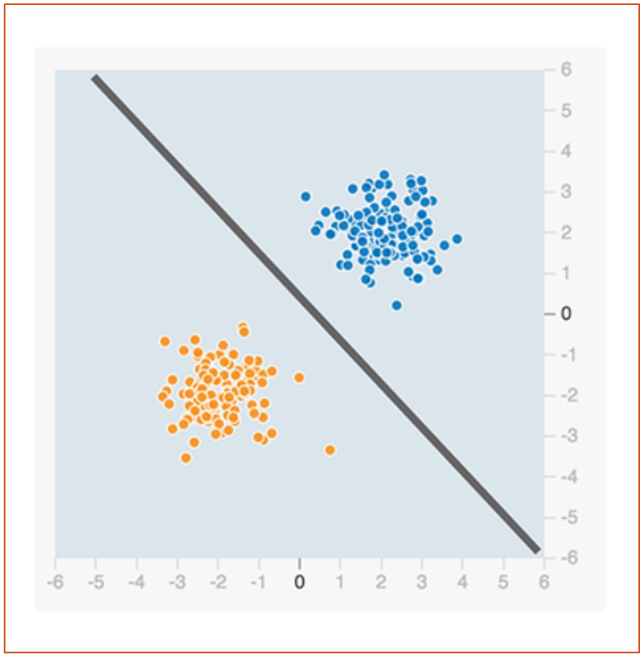


Data Science
Machine Learning
Deep Learning
Deep Neural Networks

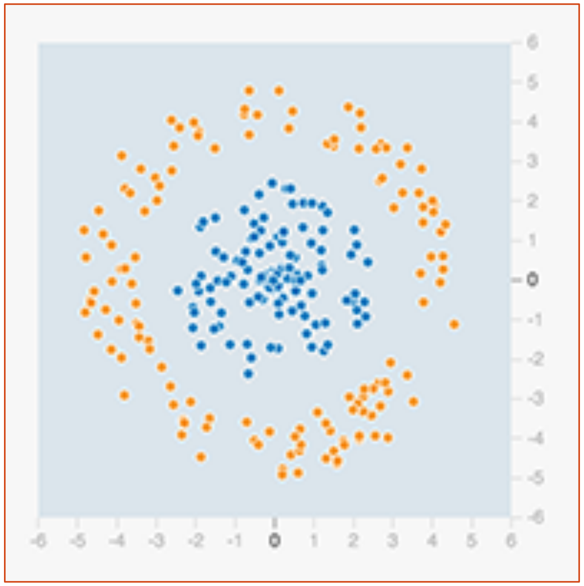
Why Neural Networks ?



Linear Discriminant Analysis



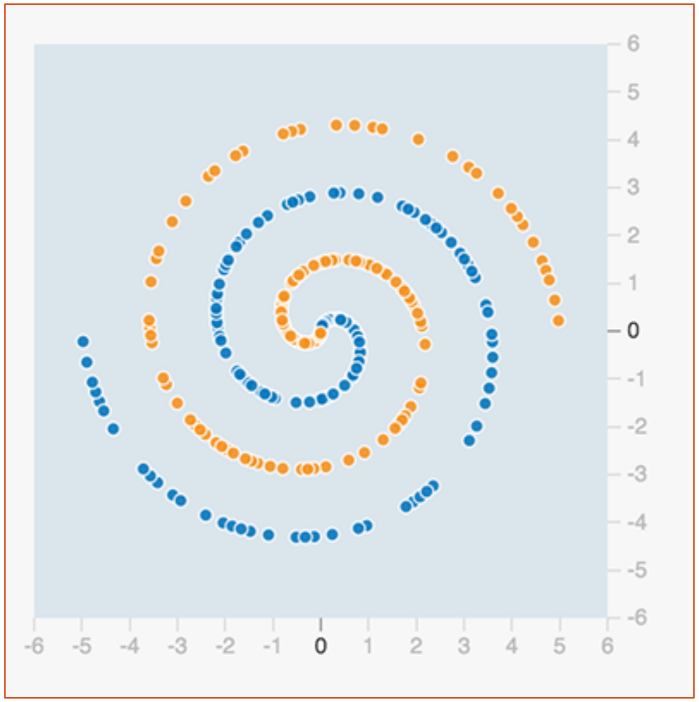
How to Classify this?



Line can't be used to classify this.

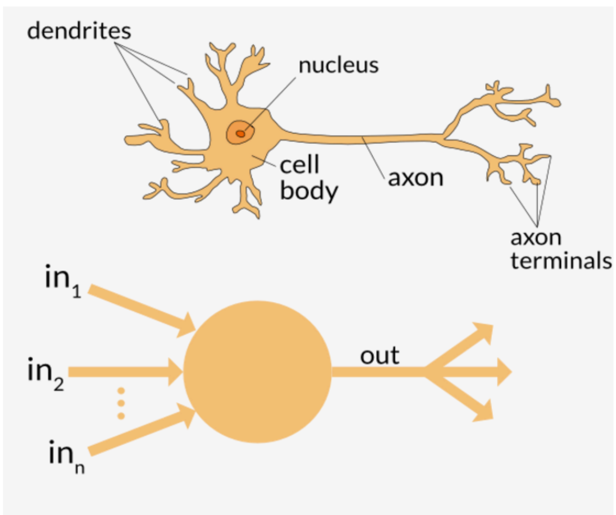
This is a so-called nonlinear classification problem.

And this?



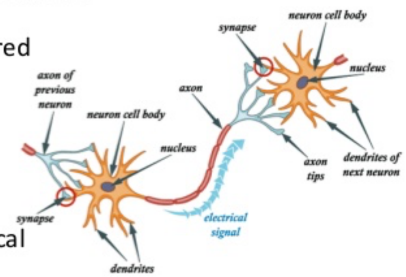
Neural Networks - Inspiration

Inspired by brain neural networks



Why Neural Networks?

- The human brain can be considered to be one of the best processors. (Estimated to contain ~10^{11} neurons.)
- Studies show that our brain can process the equivalent of ~20Mb/sec just through the optical nerves.
- If we can copy this design, maybe we can solve the “hard for a computer – easy for humans” problems.



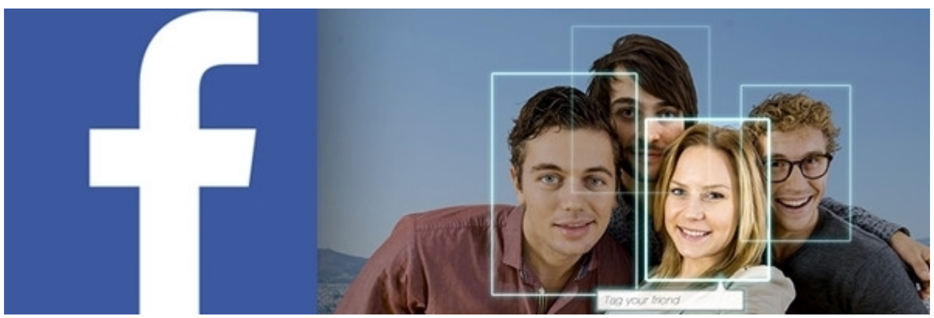
- Speech recognition
- Facial identification
- Reading emotions
- Recognizing images
- Sentiment analysis
- Driving a vehicle
- Disease diagnosis

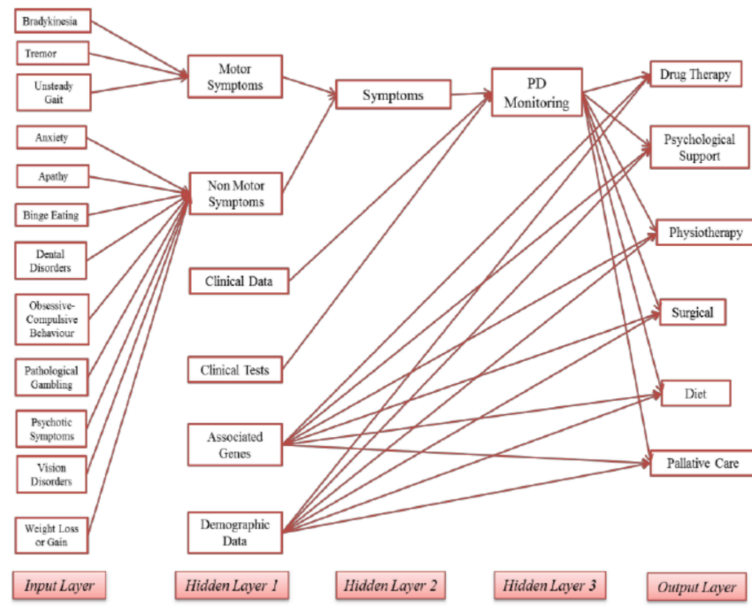
@nfmcclure

PayScale

Applications

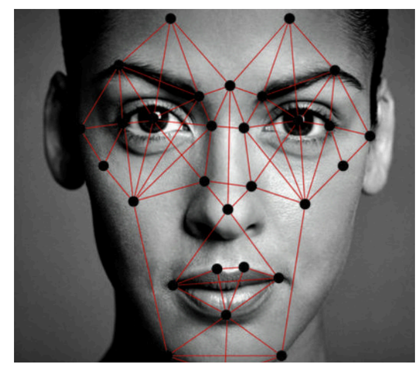
- Image and speech recognition
- Precision medicine





Applications

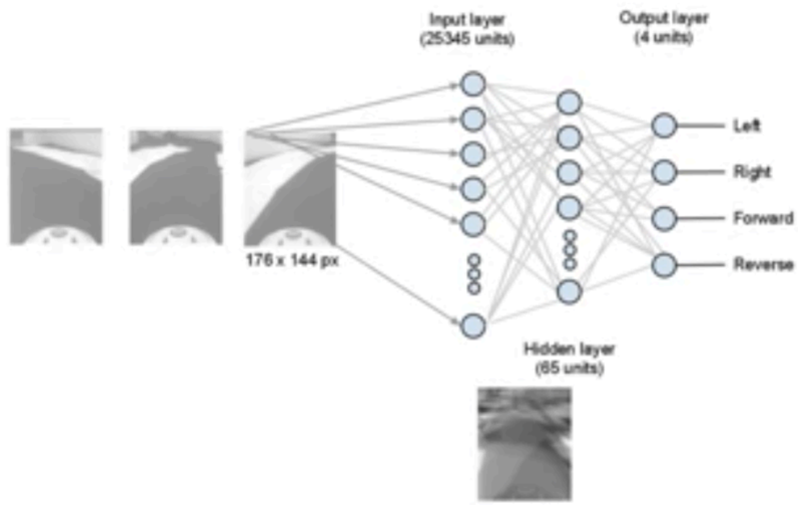
- Playing games
- Predicting court decisions
- Translating languages



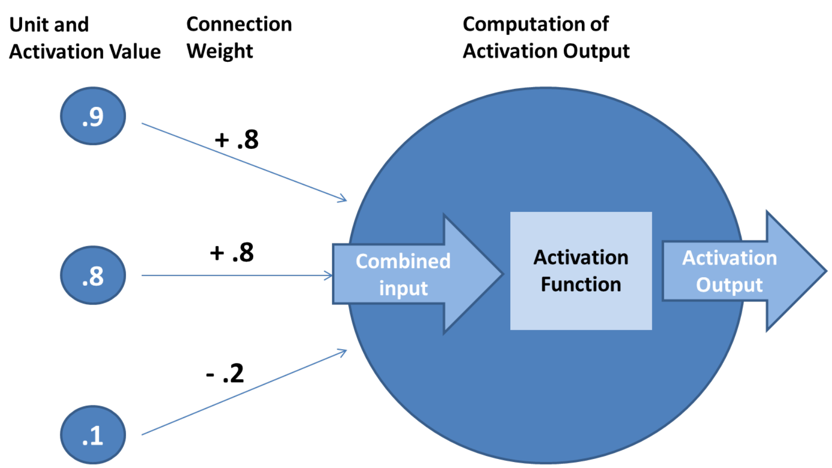
Applications

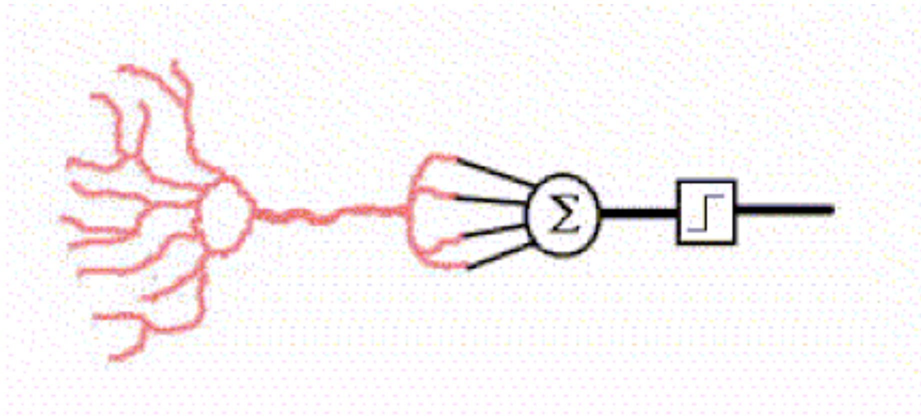
Self Driving Car

Google

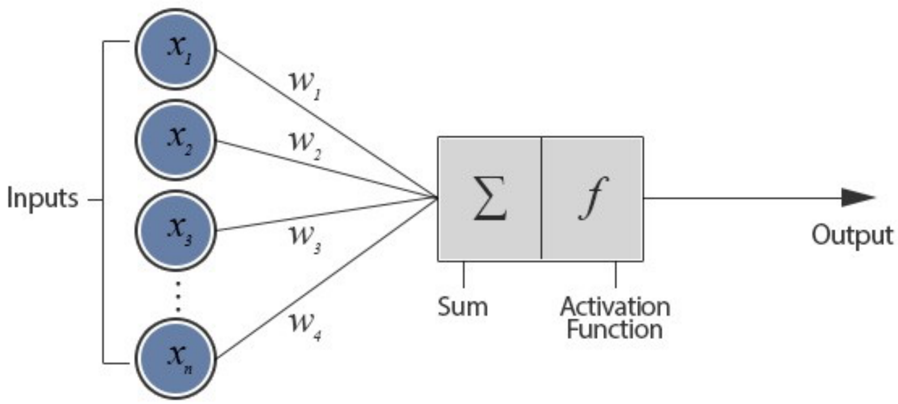


Neural Networks – How?





Neural Networks – How?



Input 1 (x1) = 0.6
Input 2 (x2) = 1.0

Weight 1 (w1) = 50%
Weight 2 (w2) = 80%

Sum

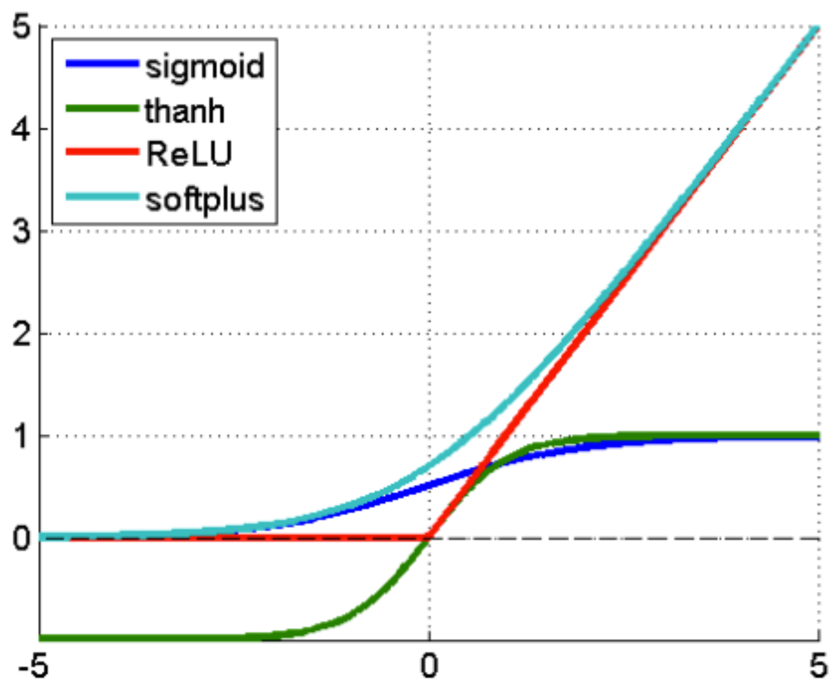
x1*w1 + x2*w2 = (0.6 x 0.5) + (1 x 0.8) = 1.1

Activation

Is Sum >= (Threshold = 1.0) ?

Neuron Fires

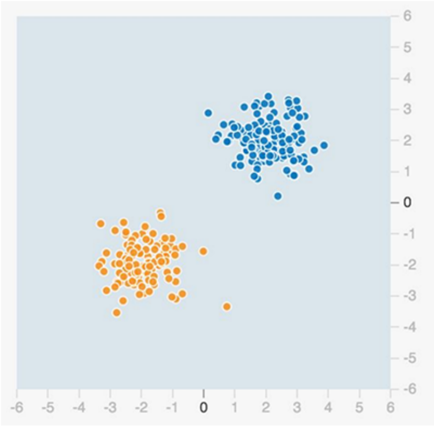
Activation Functions

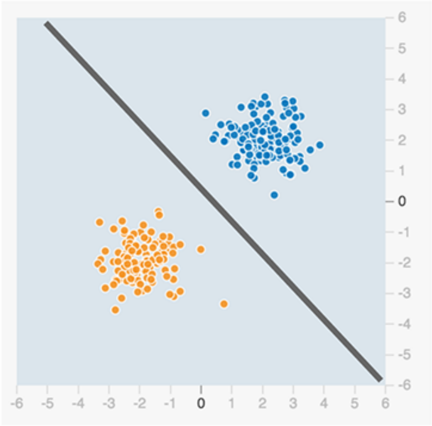


- Activation or transfer function

- Nonlinear Outputs

A Single Neuron

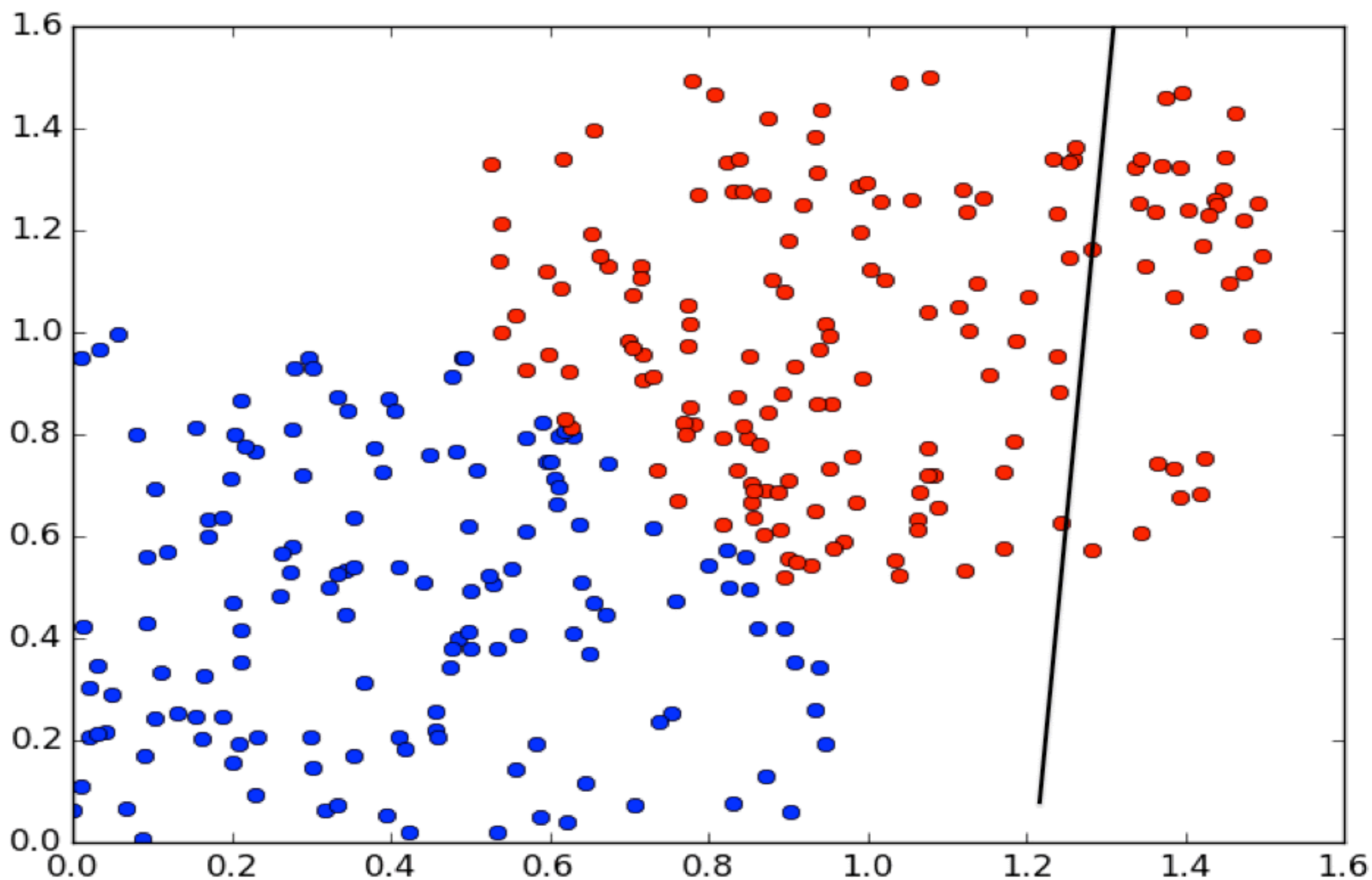




$$w_1x_1 + w_2x_2 > b$$

With two inputs, a neuron can **classify the data points in two-dimensional space into two kinds** with a straight line.

If you have three inputs, a neuron can classify data points in three-dimensional space into two parts with a flat plane, and so on. This is called "dividing n-dimensional space with a hyperplane."



1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.0 0.2 0.4 0.6 0.8 1.0 1.2 1.4 1.6

Deep Learning

Break Time

Classic Neural Networks

- Exceptional at recognizing patterns
- Impractical to have more than 1 hidden layer
- Features must be given in usable form
- Only work for supervised ML

What is Feature Engineering?

Manually choosing input features or Deriving New Features

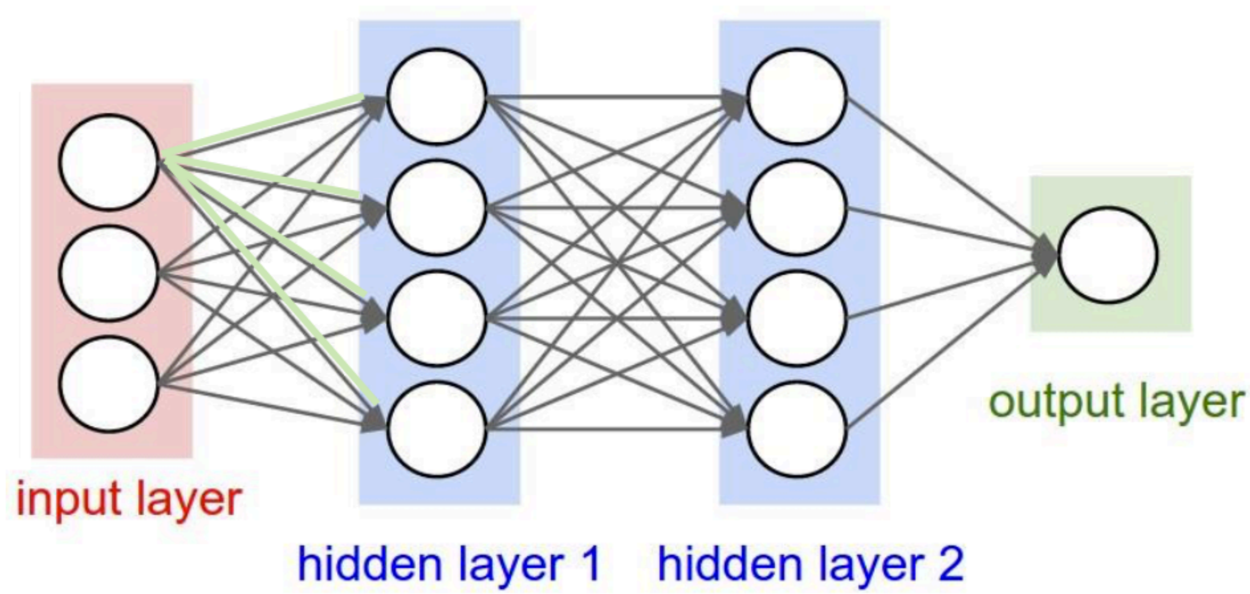
Pros
- Reduce complexity and computation
- Prevent over-fitting due to redundant features
- Prevent curse of dimensionality for small datasets

Cons
- Costly
- Requires domain knowledge
- Difficult

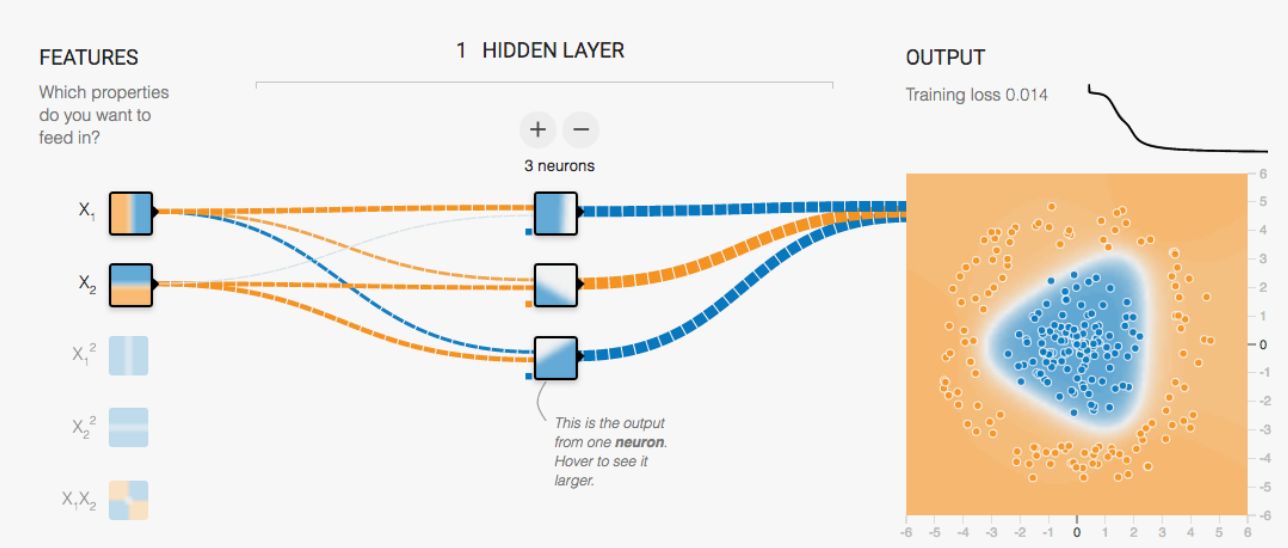
Deep Learning

- Set of techniques for learning in neural networks
- The study of neural networks
- Can be supervised or unsupervised

Neurons organized in layers



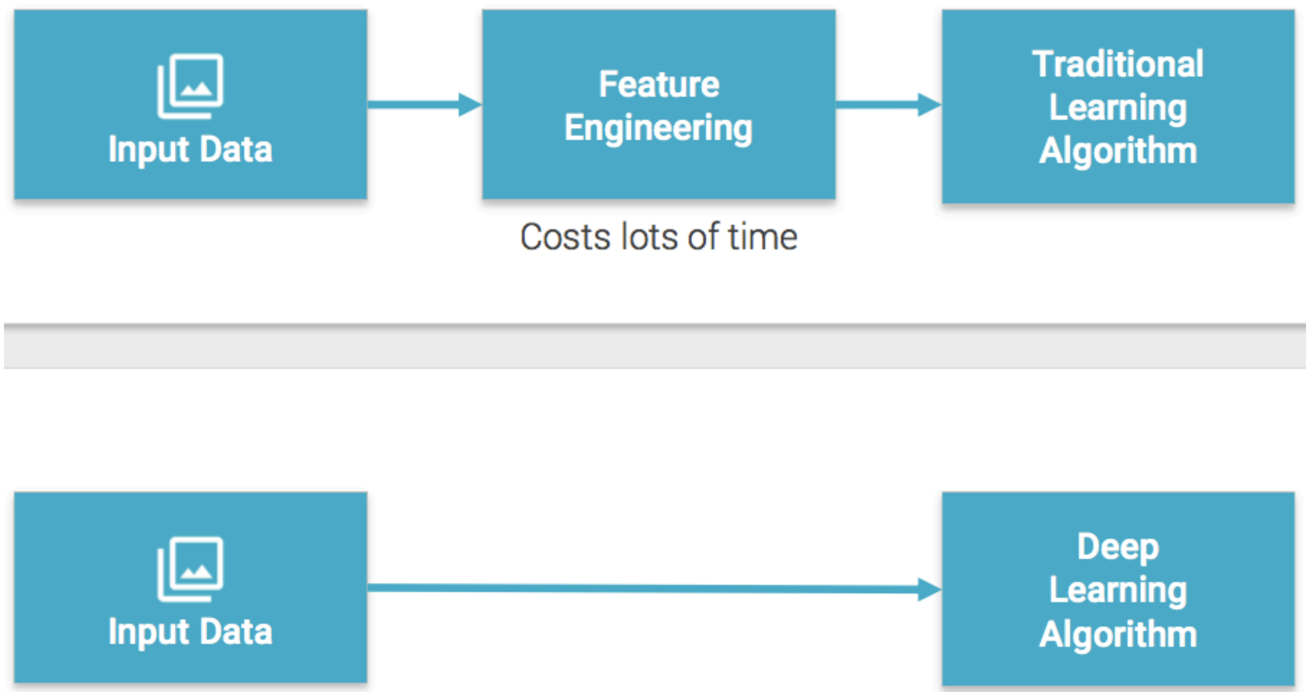
Multi Layer Perceptron (MLP)



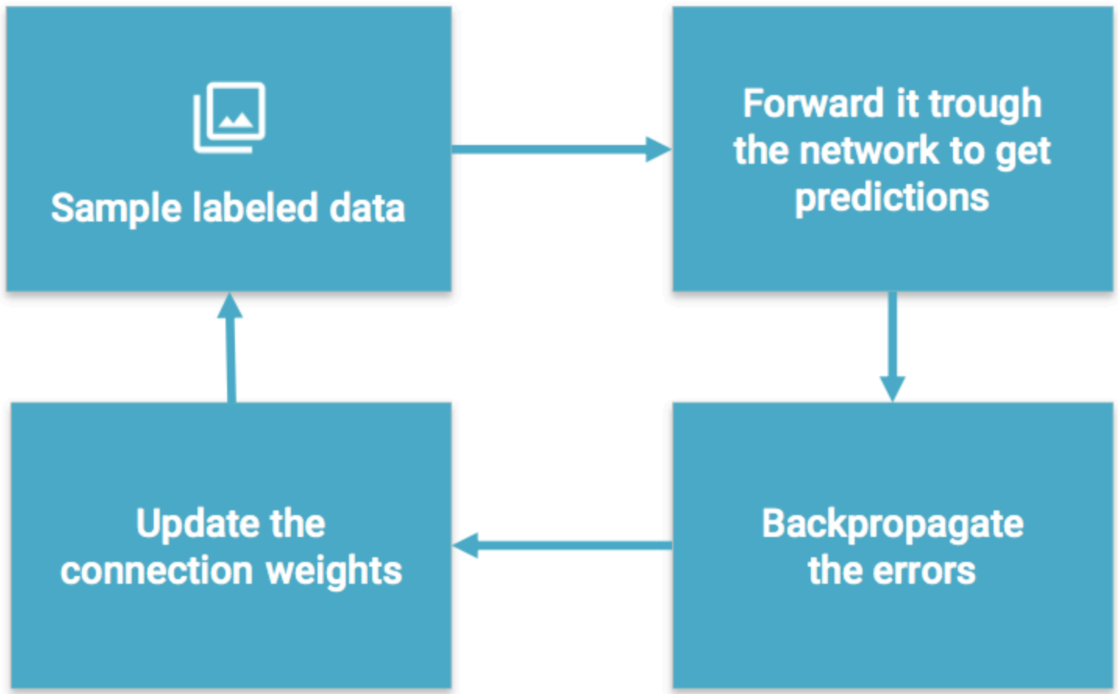
Deep Neural Networks

- Algorithms that have one or more hidden layers
- Exceptional at learning patterns
- Non-”deep” neural networks can only be supervised. Example: Support Vector Machines

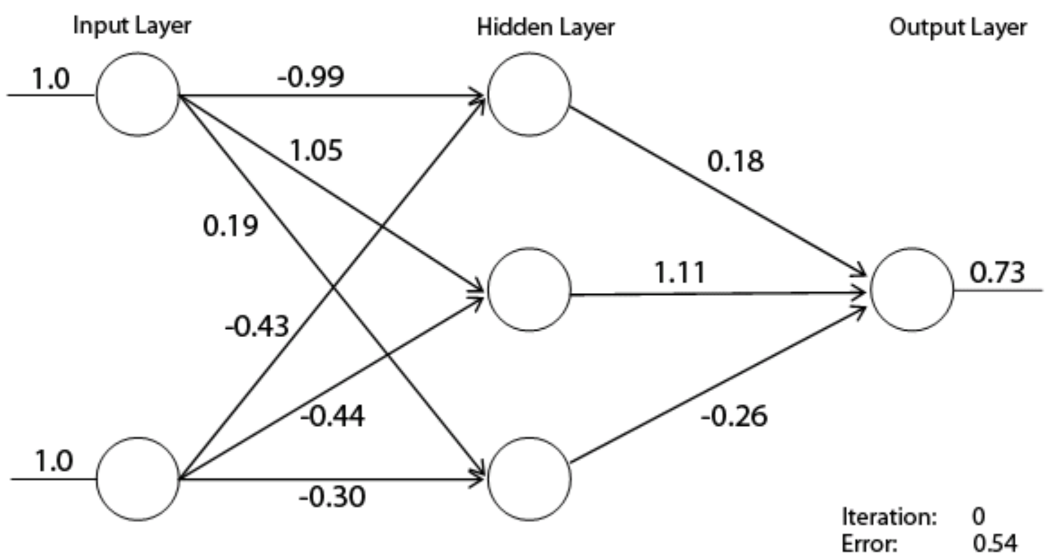
Feature engineering not needed



Training a Neural Network

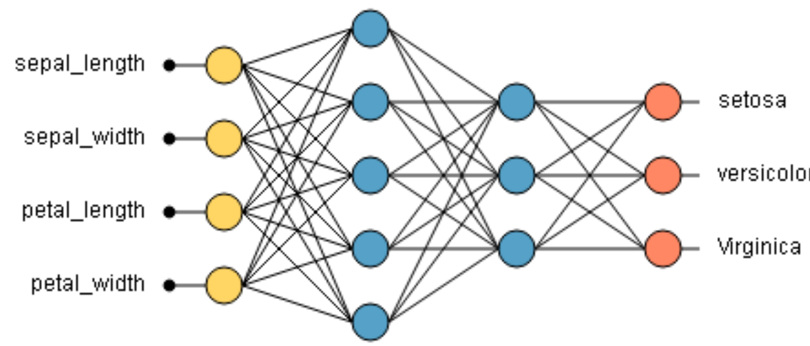


Neural Network - Learning

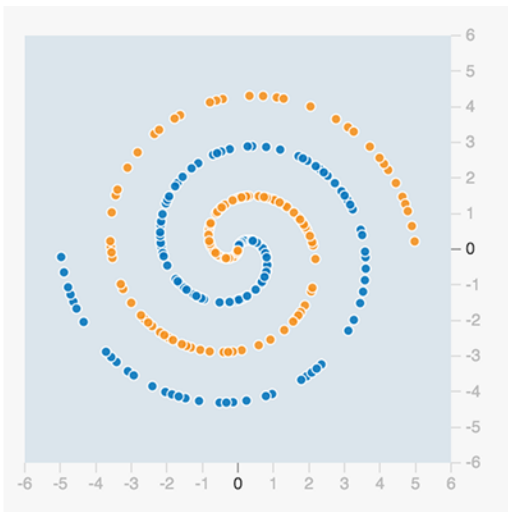


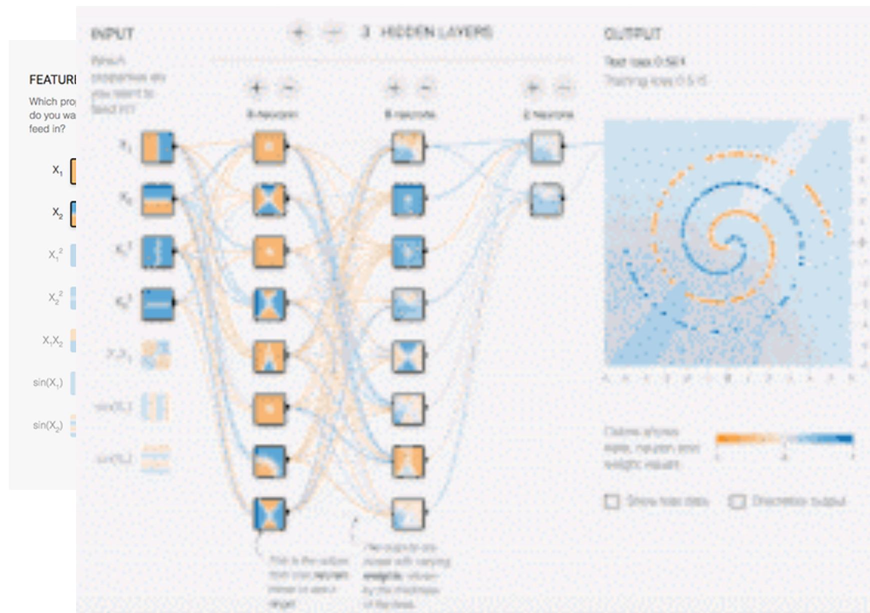
Deep Neural Networks

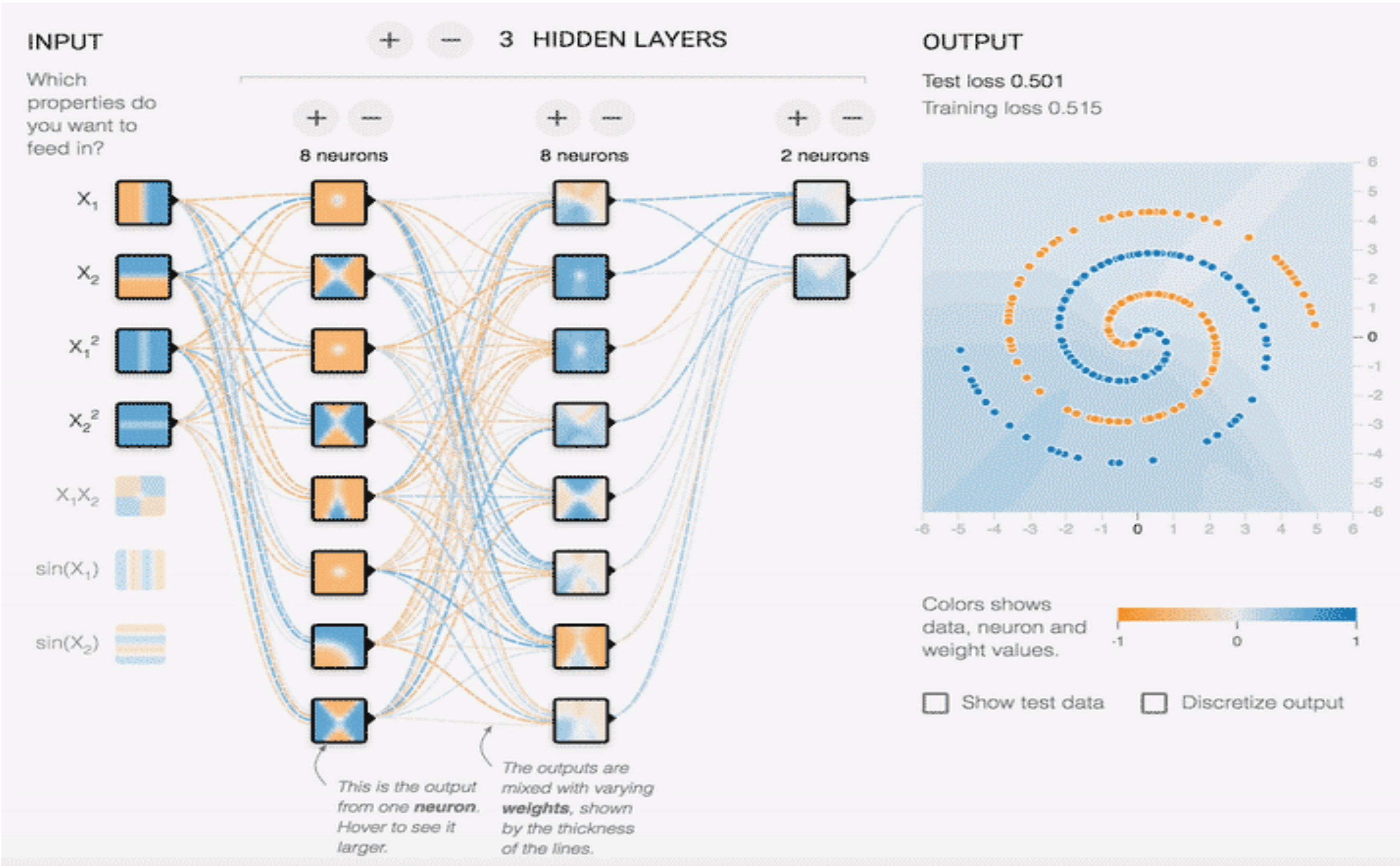
- Train on new data
- After a wrong classification, they can recalculate the importance of a feature
- Example: Leaf Dataset



Deep Neural Network






INPUT
Which properties do you want to feed in?
X_1
X_2
X_1^2
X_2^2
X_1X_2
$\sin(X_1)$
$\sin(X_2)$
3 HIDDEN LAYERS
8 neurons
8 neurons
2 neurons
This is the output from one **neuron**. Hover to see it larger.
The outputs are mixed with varying **weights**, shown by the thickness of the lines.
OUTPUT
Test loss 0.501
Training loss 0.515
Colors shows data, neuron and weight values.
-1
0
1
Show test data
Discretize output

Strengths of Deep Neural Networks

- Classifying based on combinations of features or units

- Things that human brains do automatically

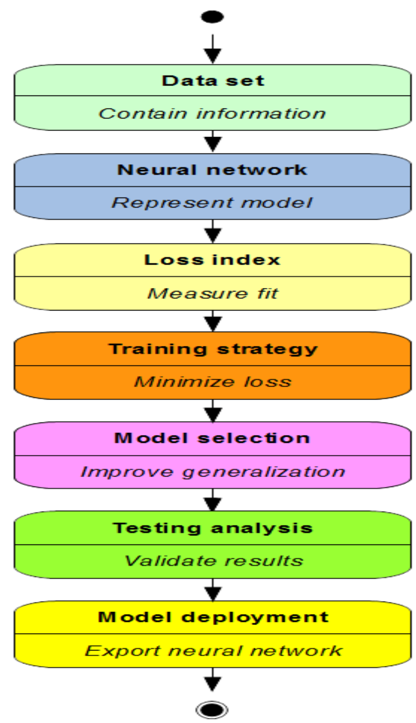


Image Recognition

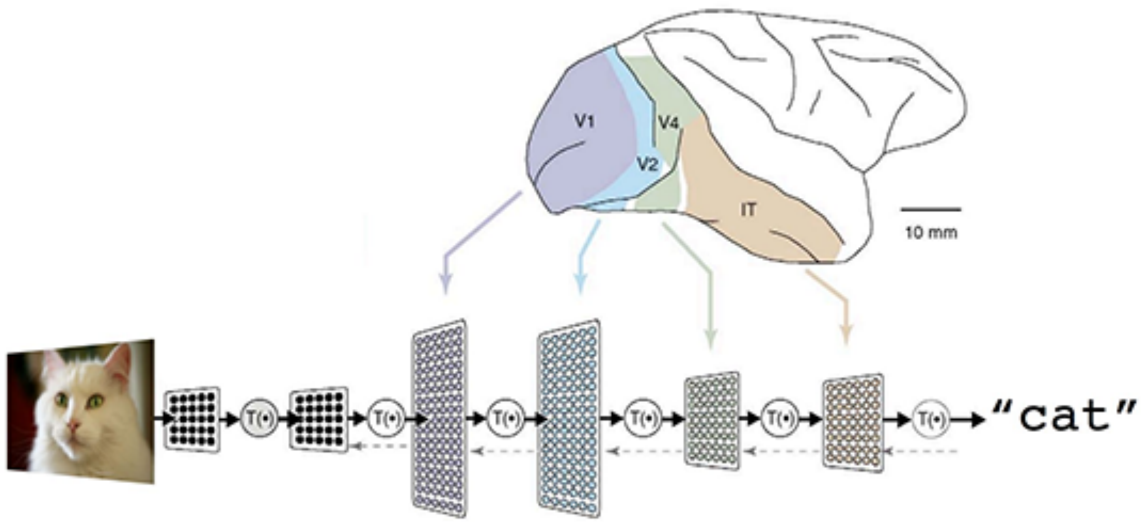
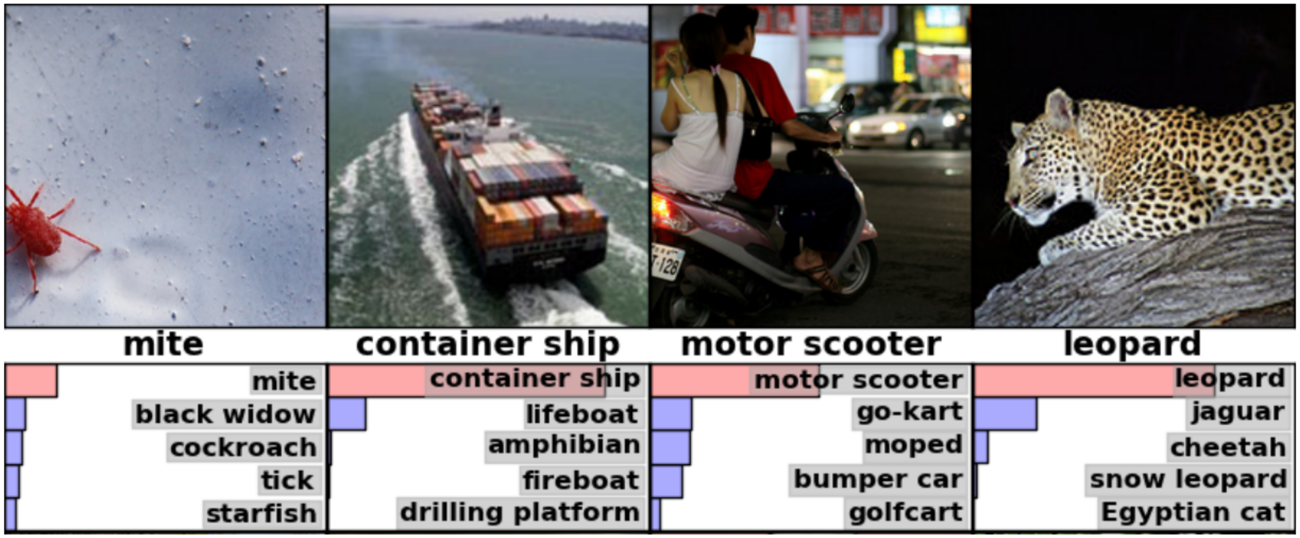


Image Recognition

ImageNet Challenge (1000 Classes)

How often the model fails to predict the correct answer as one of their top 5 guesses -- termed "top-5 error rate".



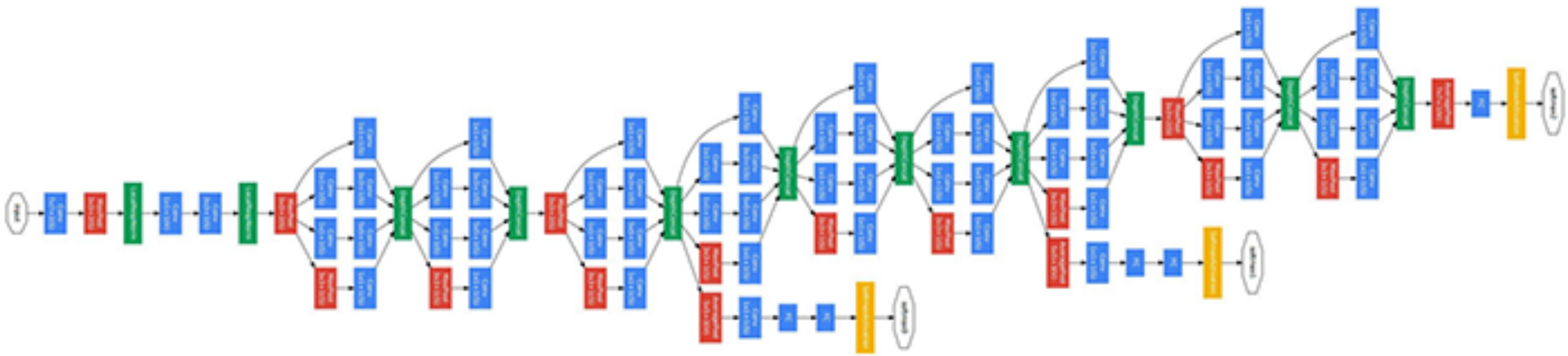
How well do humans do on ImageNet Challenge?

There's a blog post by Andrej Karpathy.

He reached 5.1% top-5 error rate.

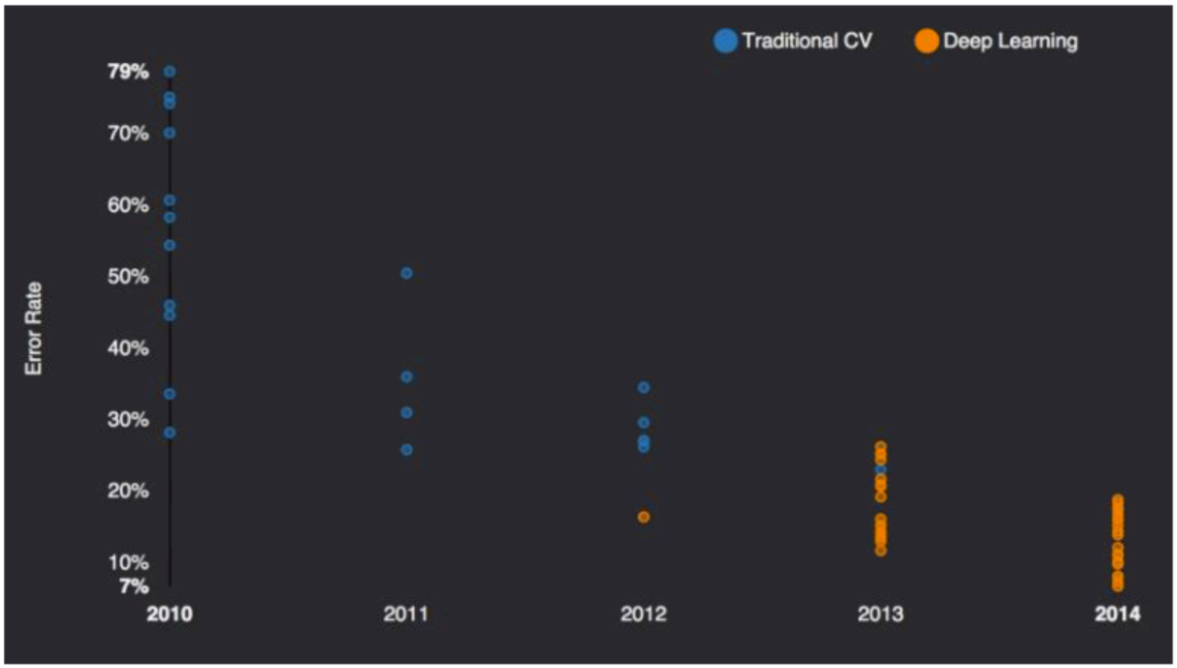
ImageNet Challenge

AlexNet achieved by setting a top-5 error rate of **15.3%** on the 2012 validation data set
Inception (GoogLeNet) achieved **6.67%**
BN-Inception-v2 achieved **4.9%**
Inception-v3 reaches **3.46%**



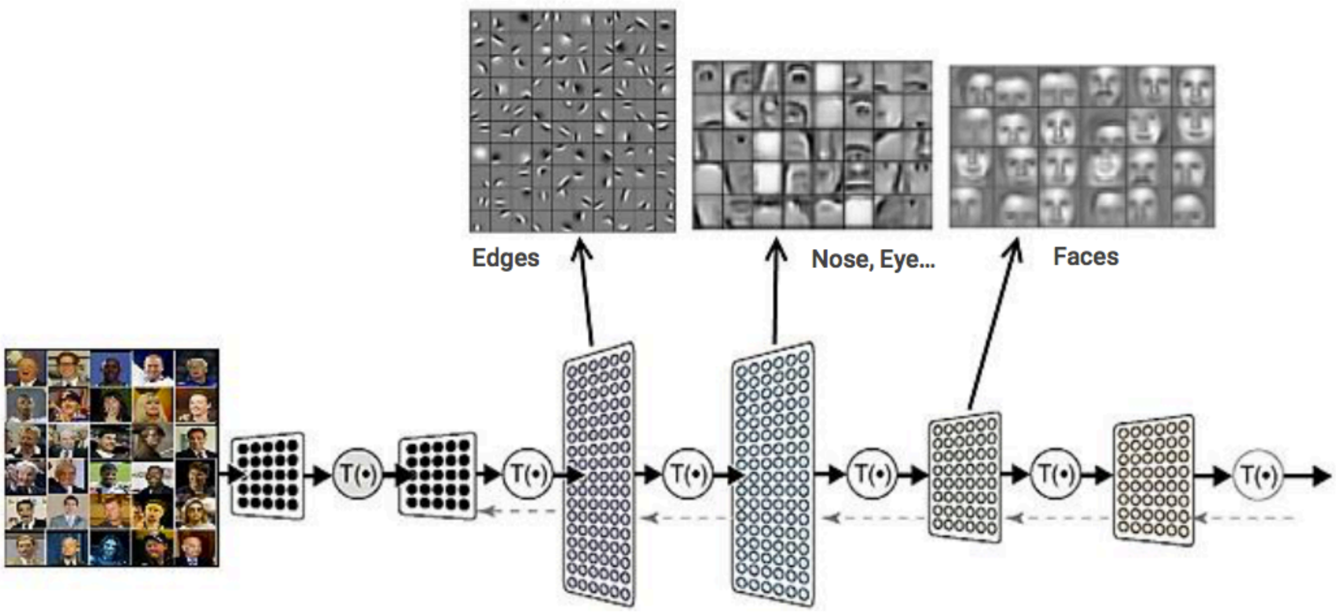
Inception V3 – by Google

Blowing away the competition



ImageNet: The "computer vision World Cup"

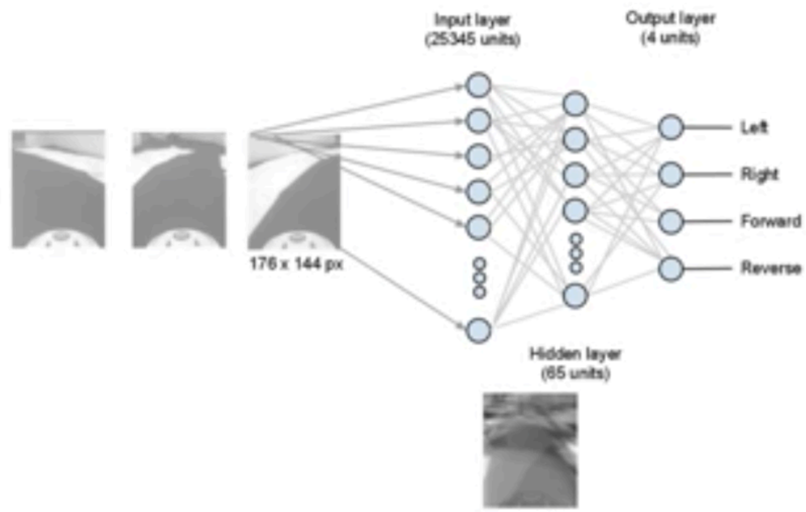
Facial Recognition



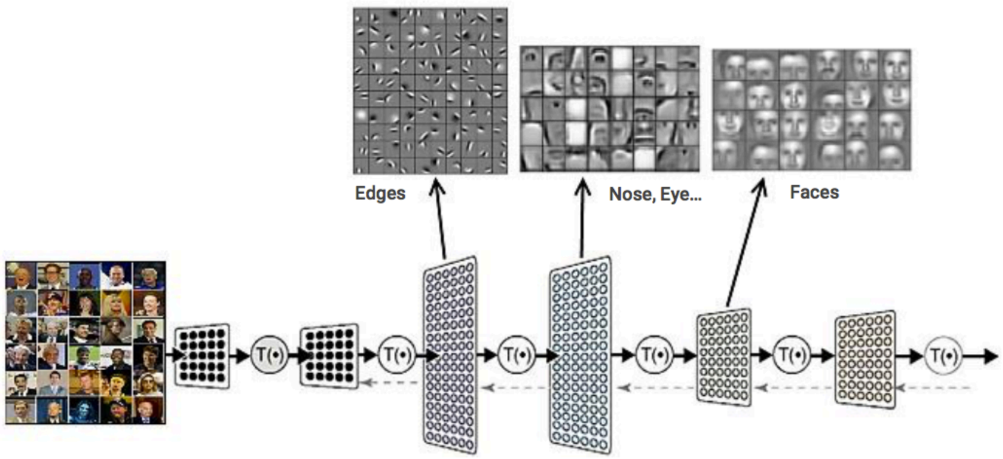
Hierarchy: Input layer Hidden Layers Output Layer

Automated Driving

Self Driving Car



How it works?

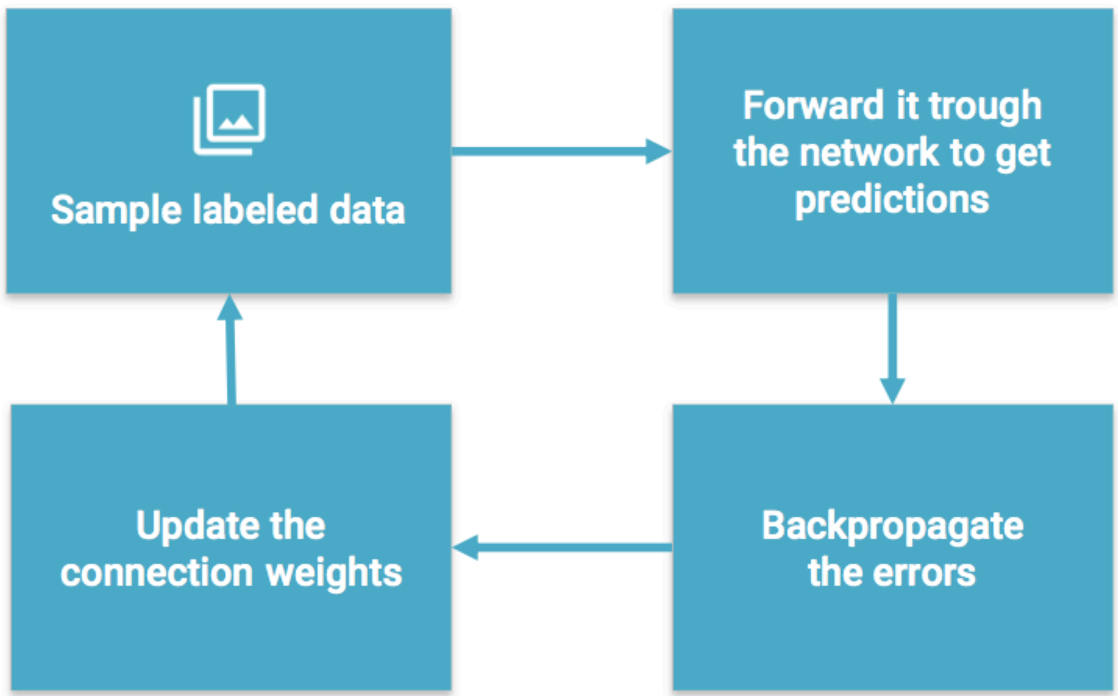


Each Layer detects a particular feature by scanning an image

Some layers pool features

Final layer classifies

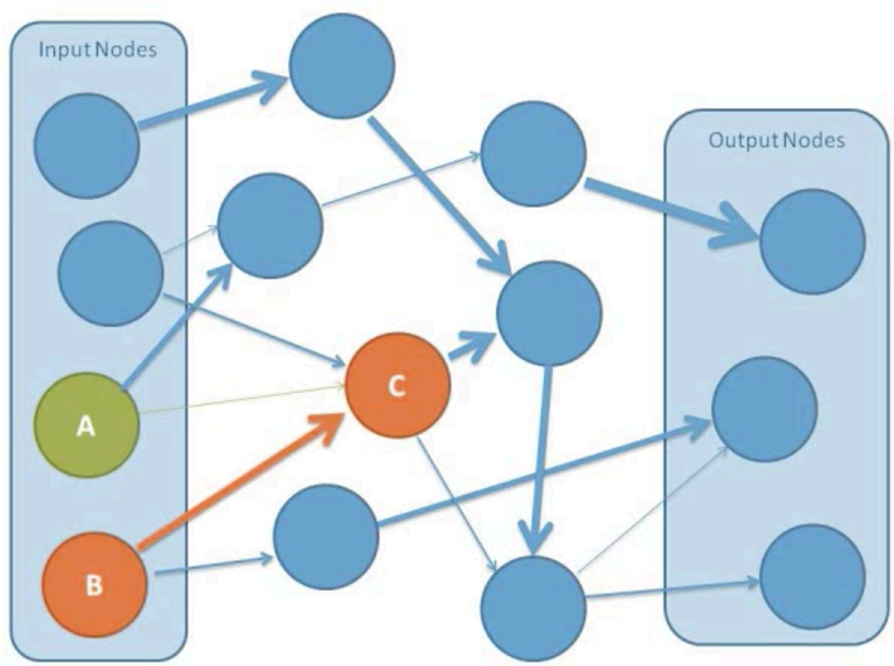
Training a Neural Network



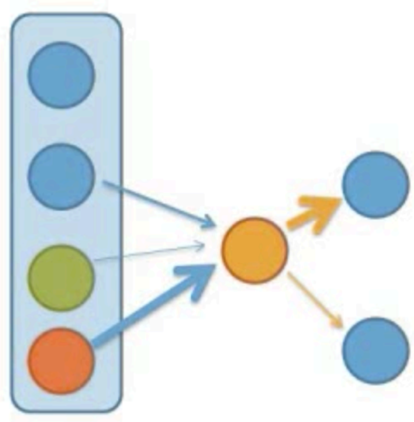
Requirements

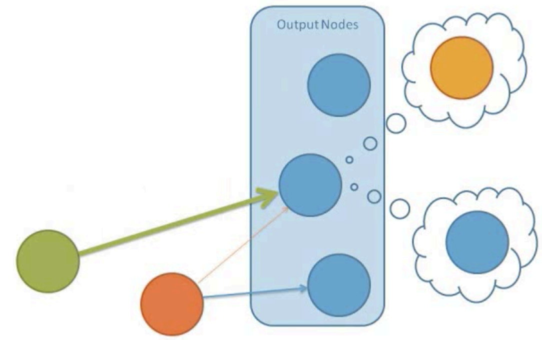
- Lots of labeled data of high quality
- Measurable goals and cost function
- Lots of computing power

Connections have different weights

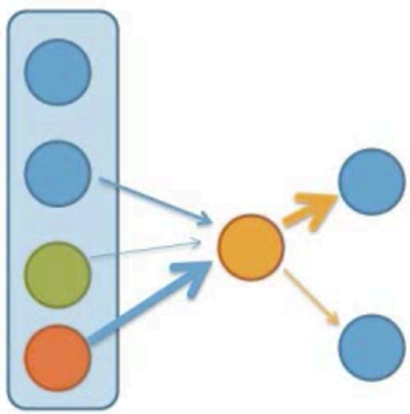


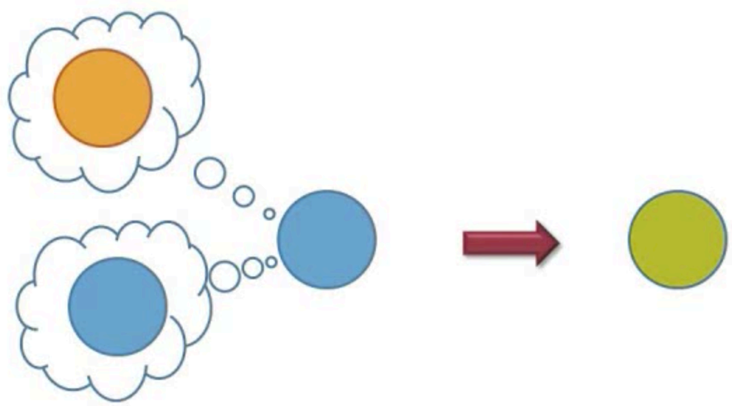
Node sets its value, triggers next node



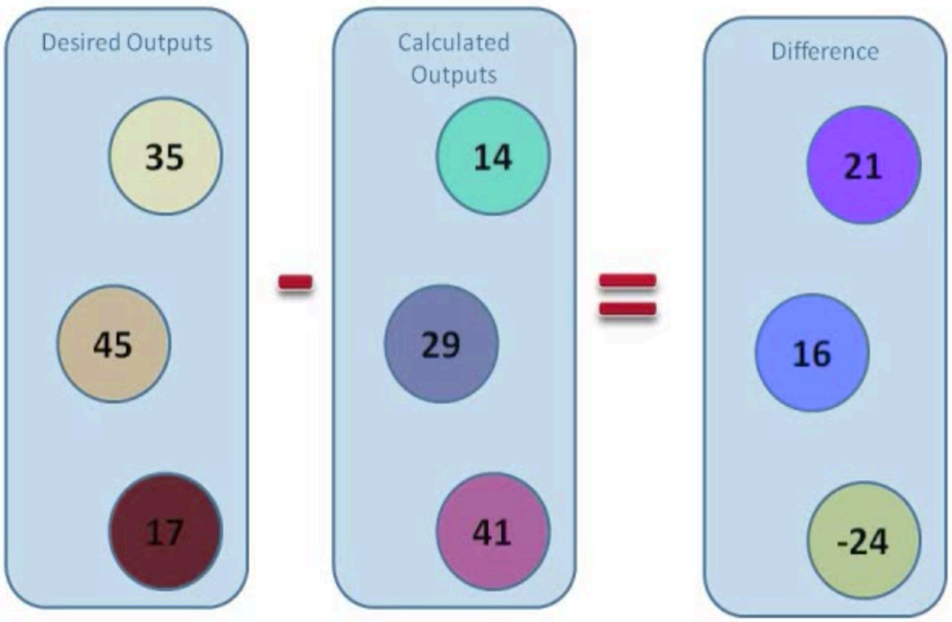


New value = old value + input

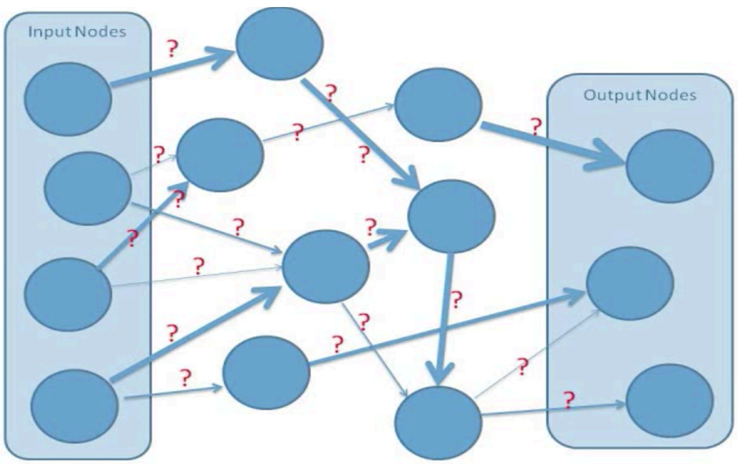




Error in the network

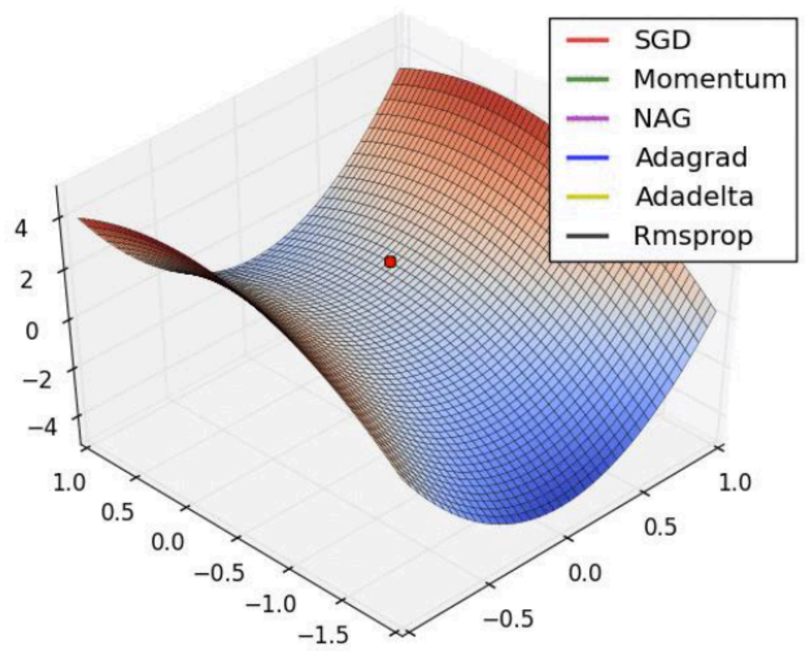


Backprropagation



The process to learn the correct weights starts with random weights

Gradient Descent



Find the minimum of the cost function

Use that to adjust the weights

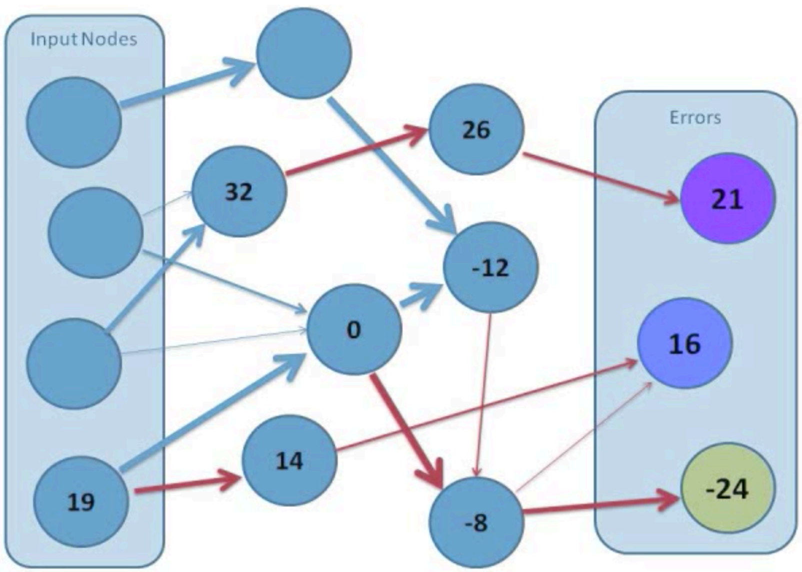
Learning Rate

Learning Rate

$$W_{j,i} \leftarrow W_{j,i} + \alpha \times a_j \times \Delta_i,$$

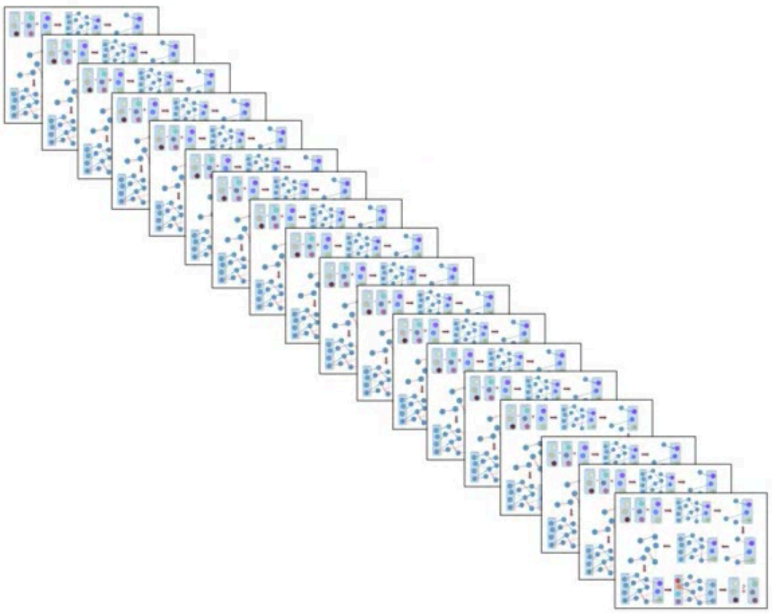
$$\Delta_i = (T_i - O_i) \times g'\left(\sum_j W_{j,i} a_j\right).$$

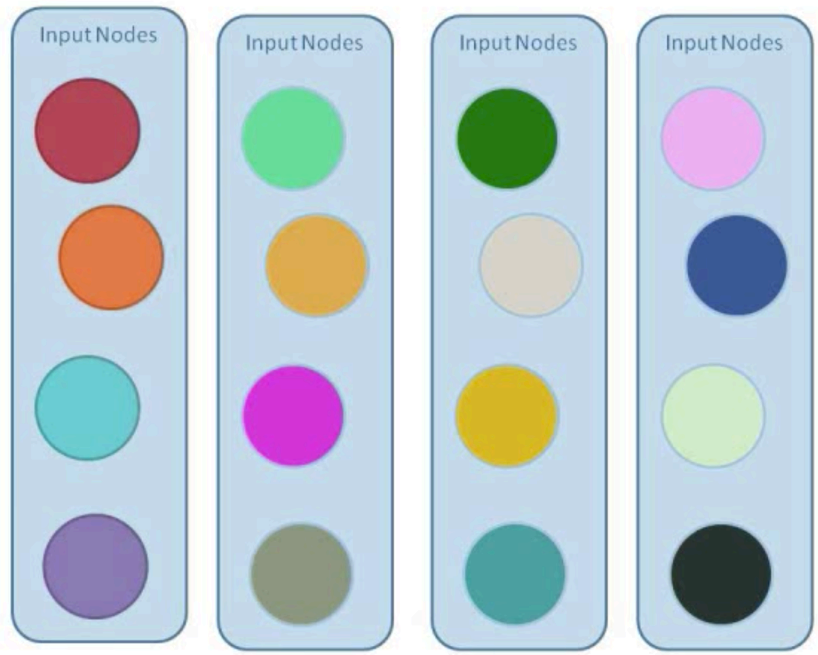
Backpropagation



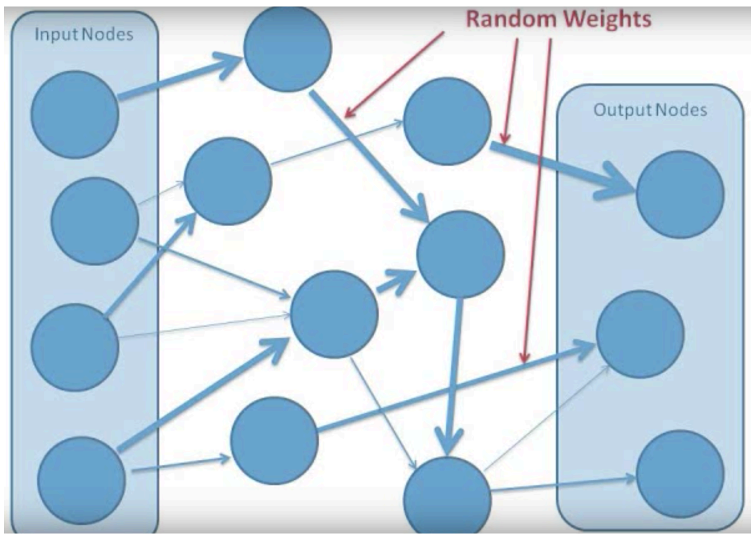
Improved connection weights compared to the random ones

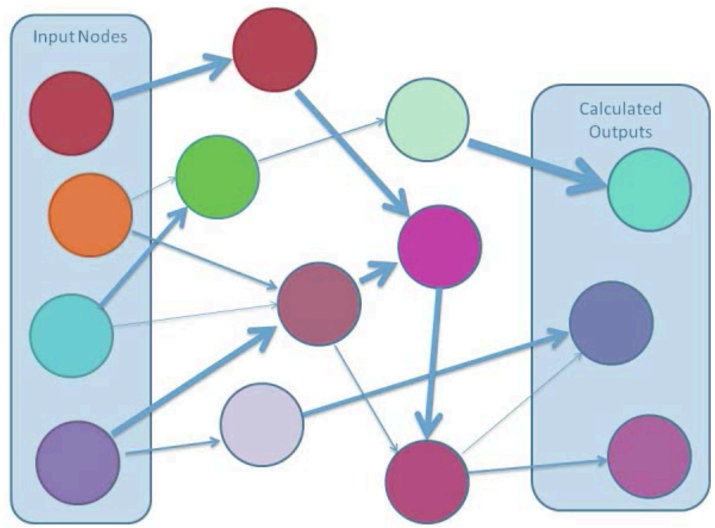
Repeated for each input set



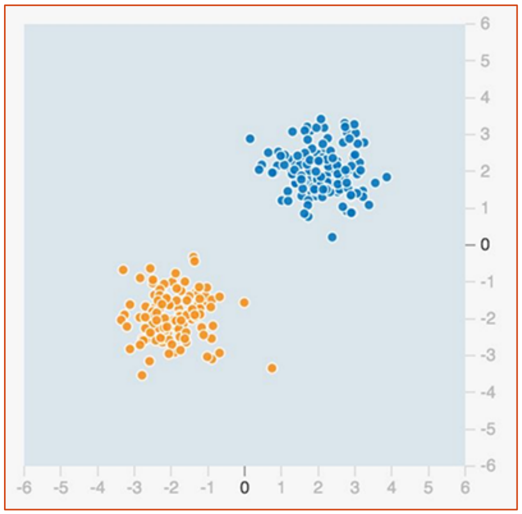


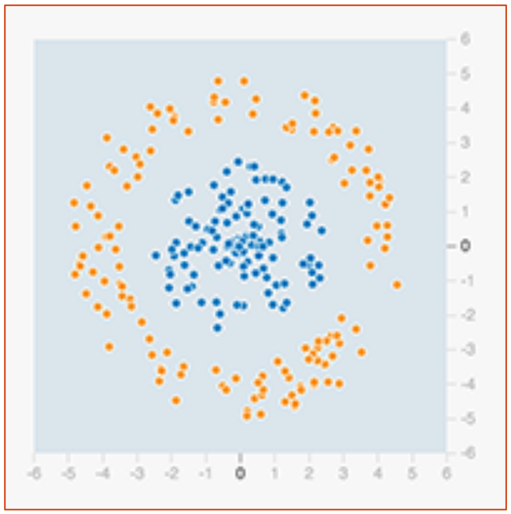
Determining connection weights

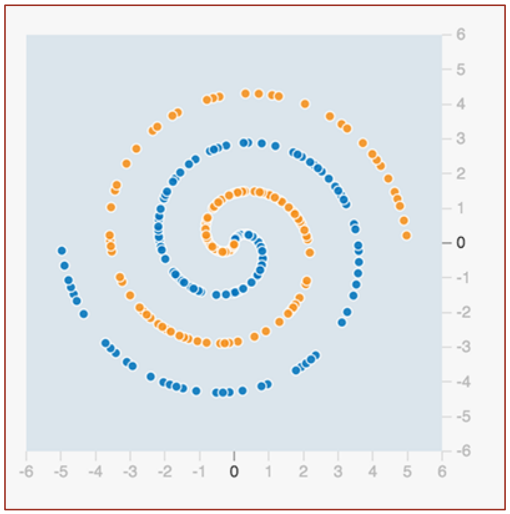




Neural Networks - Playground







Challenges

- Training on less data
- Natural Language Processing
- Understanding videos
- Deep reinforcement
- Unsupervised learning
- Quantum computing

Digit Classification Case

Lunch

Course Assignments

Programming Assignments

Reading Assignments

Presentation Assignments

Technical Skills Assignments

Writing Assignments

Technical Assignment

Install & Setup RStudio on AWS

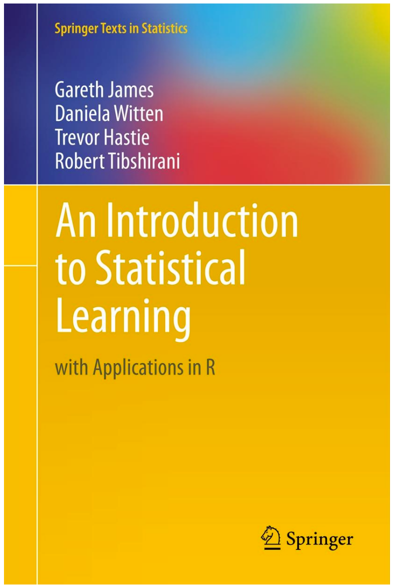
Programming Assignment

Install & Complete: Swirl – Exploratory Data Analysis

Install & Complete: Swirl – Statistical Inference

Reading Assignment

Read Chapter 5: Resampling Methods



Writing Assignment

Submit by Saturday

Written Report (not to exceed 15 pages) on Mushroom Classification Case

Presentation Assignment

By Saturday Submit

Your Presentations on Mushroom Classification Case

1. Technical Presentation
2. Business Presentation (Not to exceed 5 slides)

Instructor Case – MNIST Digit Classification

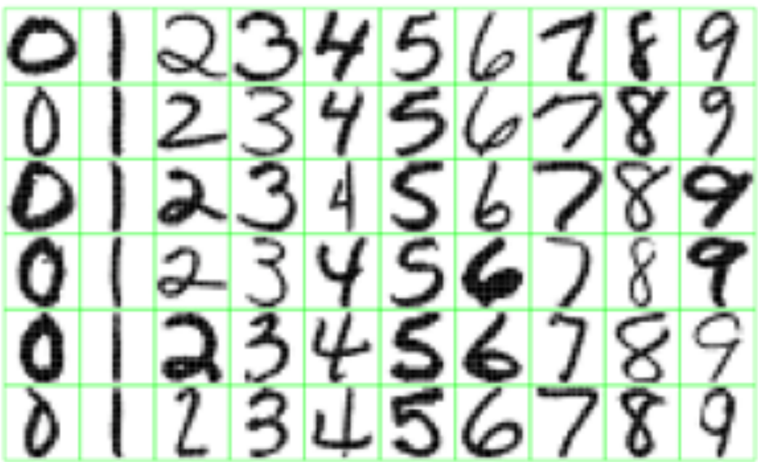
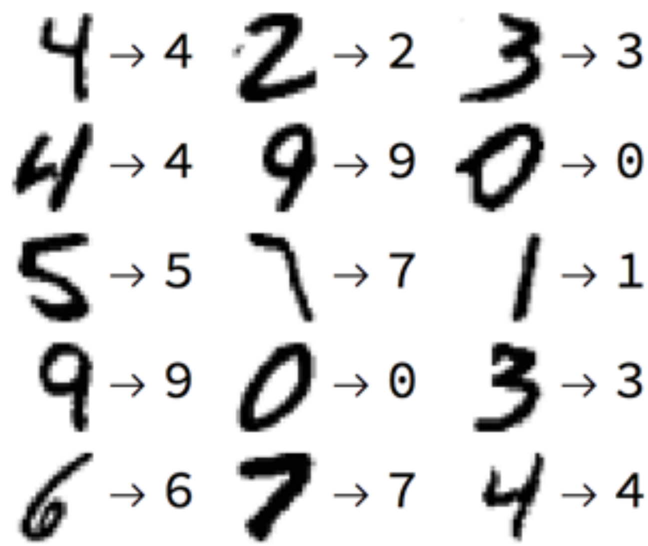
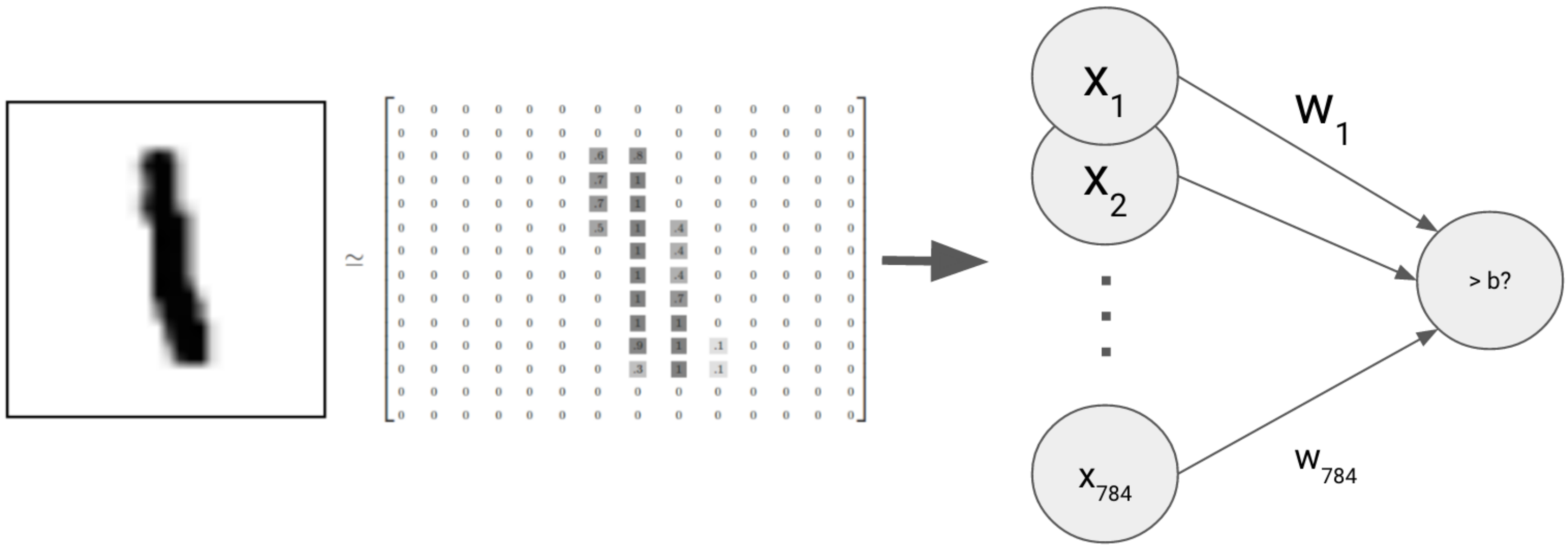


Figure 1.2: *Examples of handwritten digits from U.S. postal envelopes.*



Instructor Case - MNIST Digit Classification



Digit Classification Case

Demo Time

Digit Classification Case

Introduction to AWS

Summary

Neural Networks

Multi Layer Perceptron

Deep Neural Networks

- Examples
- Hidden layers
- Backpropagation
  - Weights
  - Errors
  - Learning coefficients
- Requirements

Gradient Descent