HSV-A.I.
HUNTSVILLE ALABAMA

Intro to Artificial Intelligence, & Machine Learning

September 25, 2019

NASA INTERNATIONAL SPACE APPS CHALLENGE

Huntsville AI

What we cover:

- Application - how to solve a problem with a technology
- Theory - for those times when knowing “How” something works is necessary
- Social / Ethics - Human / AI interaction
- Brainstorming - new uses for existing solutions
- Hands on Code - for those times when you just have to run something for yourself
- Coworking Night - maybe have combined sessions with other groups to discuss application in their focus areas

https://www.facebook.com/groups/hsvai/

https://www.meetup.com/Huntsville-AI/

https://www.linkedin.com/groups/12177562/

HSV-A.I.

HUNTSVILLE ALABAMA

About me...

J. Langley

Chief Technical Officer at CohesionForce, Inc & Founder of SessionBoard, Huntsville AI

Involved in Open Source (Eclipse & Apache Foundations)

Started working with AI about 15 years ago when Intelligent Agents were all the rage.

Developed a Naive Bayes approach for text classification, a Neural Network for audio classification, heavily into NLP.

NASA INTERNATIONAL SPACE APPS CHALLENGE

Intro to a REALLY BIG TOPIC

So…

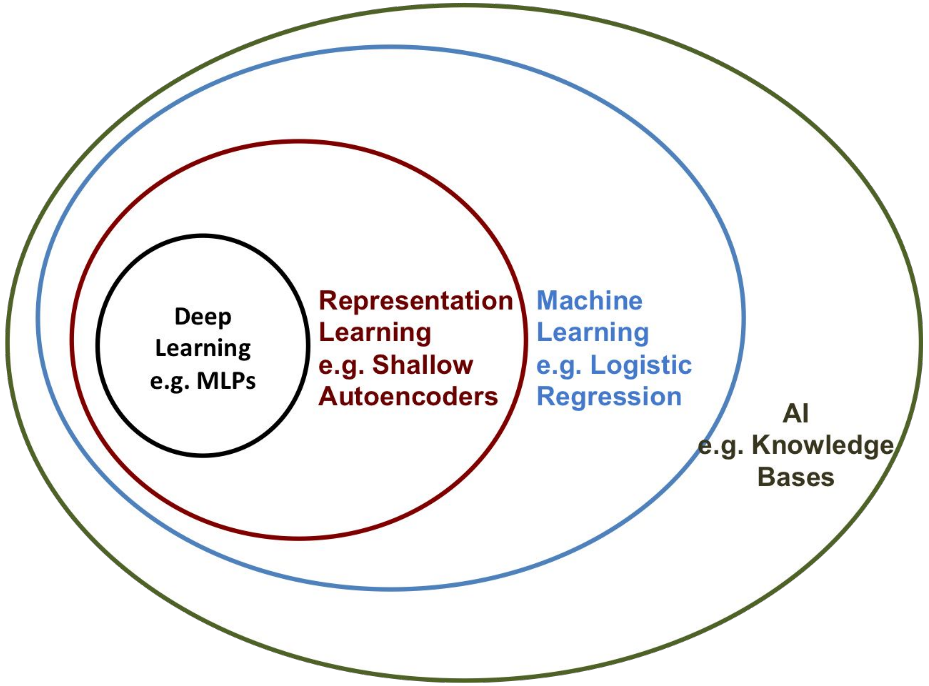
Artificial Intelligence and Machine Learning are pretty big topics to cover.

This session will attempt to help you categorize the types of problems that AI is best suited to solve.

We will cover a bit of history to give an idea of how fast things are changing.

We will also cover several of the tools and resources used by people working in this field.

Intro to a REALLY BIG TOPIC

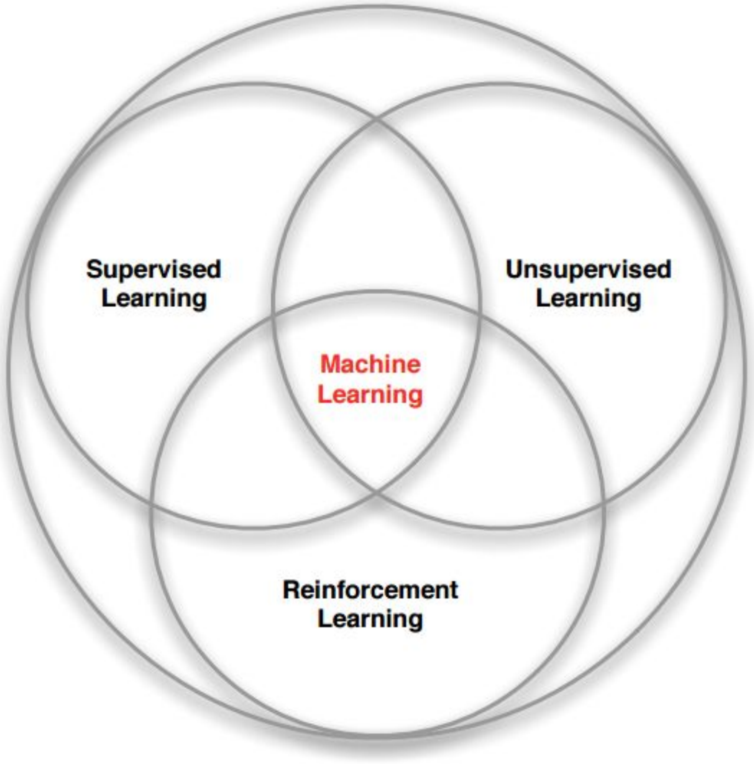


Here’s a useful way to think about the relationship between AI, ML, and Deep Learning.

This is adapted from Ian Goodfellows book:

Deep Learning : https://www.deeplearningbook.org/

Intro to a REALLY BIG TOPIC



Here's a another way to break down the topic of AI, ML, and Deep Learning.

Supervised Learning - Labelled data is used to create a model which is applied to similar data

Unsupervised Learning - used to draw inferences from datasets consisting of input data without labeled responses.

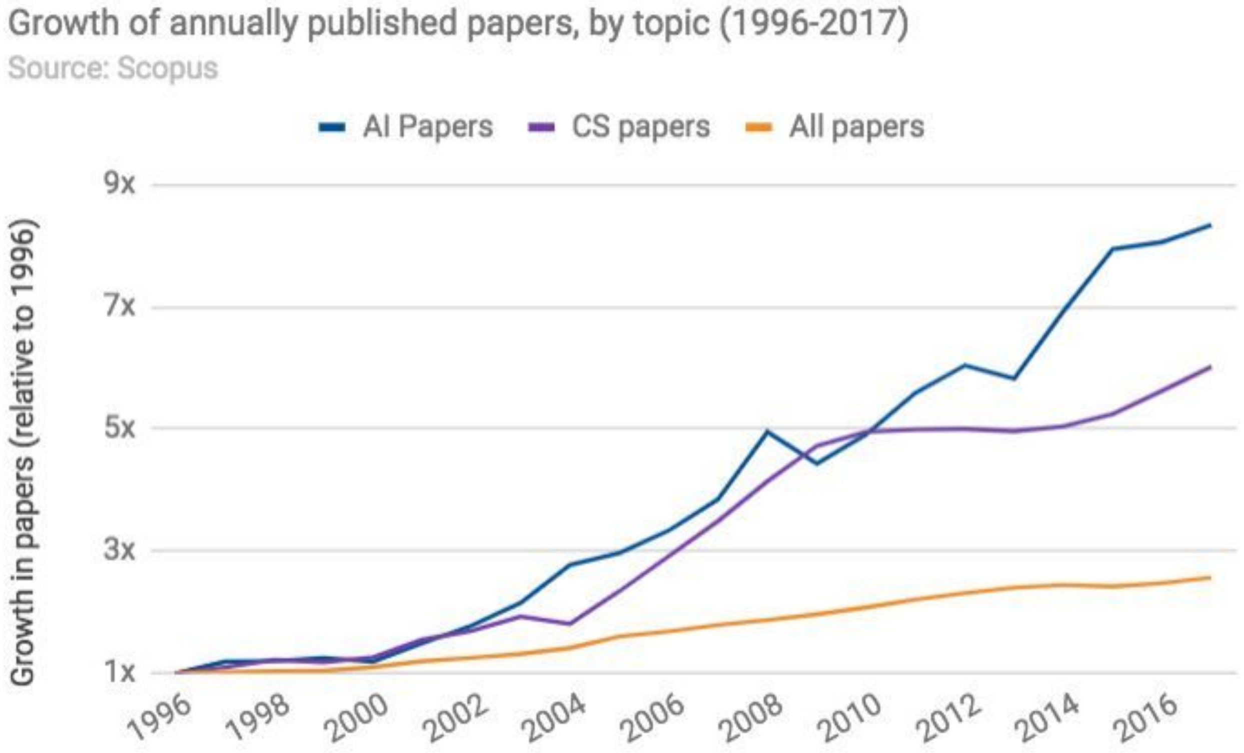
Reinforcement Learning - concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward.

Catching up with a moving target

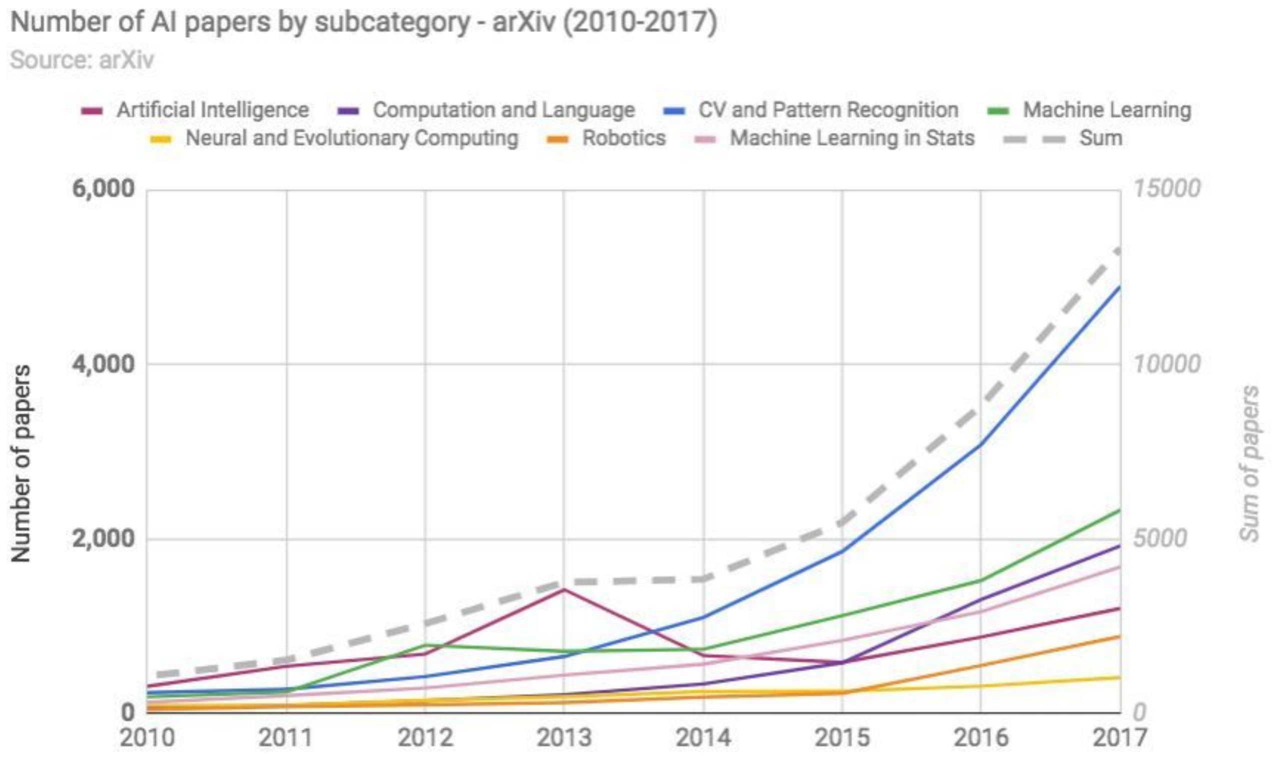
Another difficulty with AI/ML is the speed at which advancements are occuring.

Next, we'll look at some statistics from the AI Index 2018 Report: http://cdn.aiindex.org/2018/AI%20Index%202018%20Annual%20Report.pdf

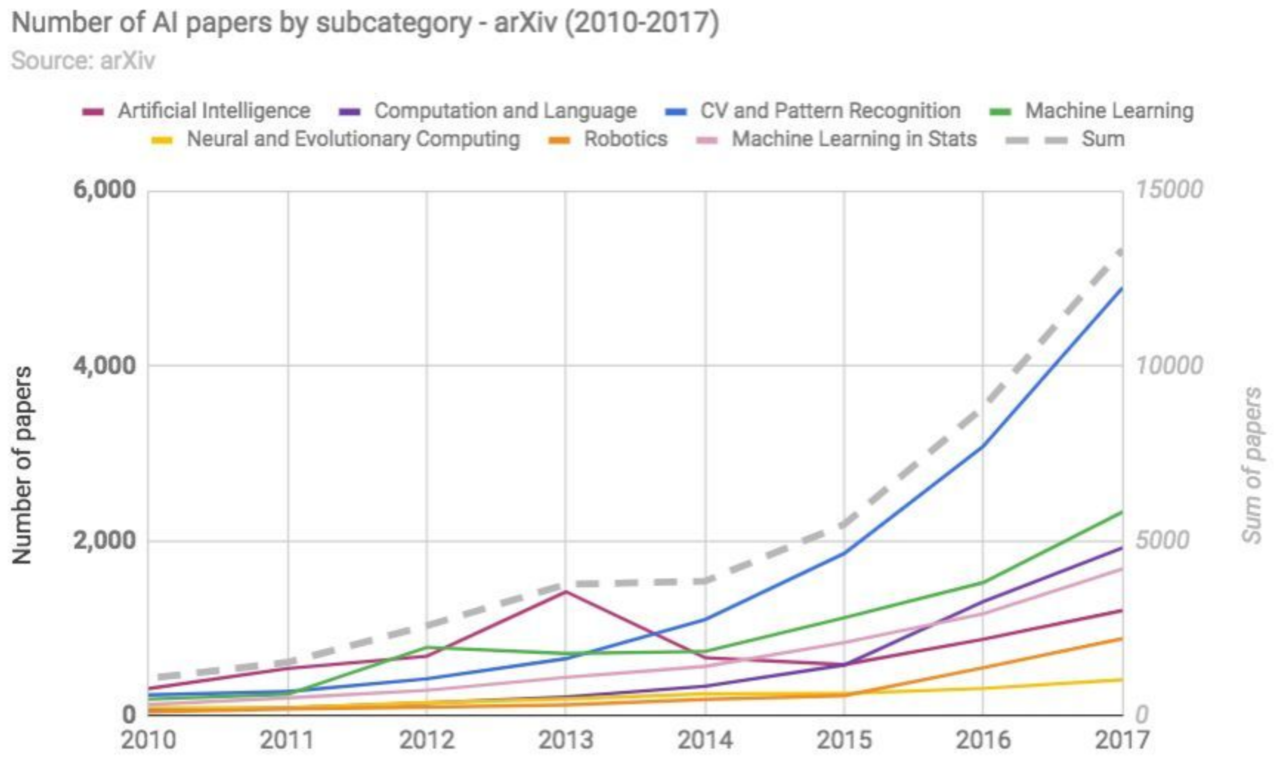
Catching up with a moving target



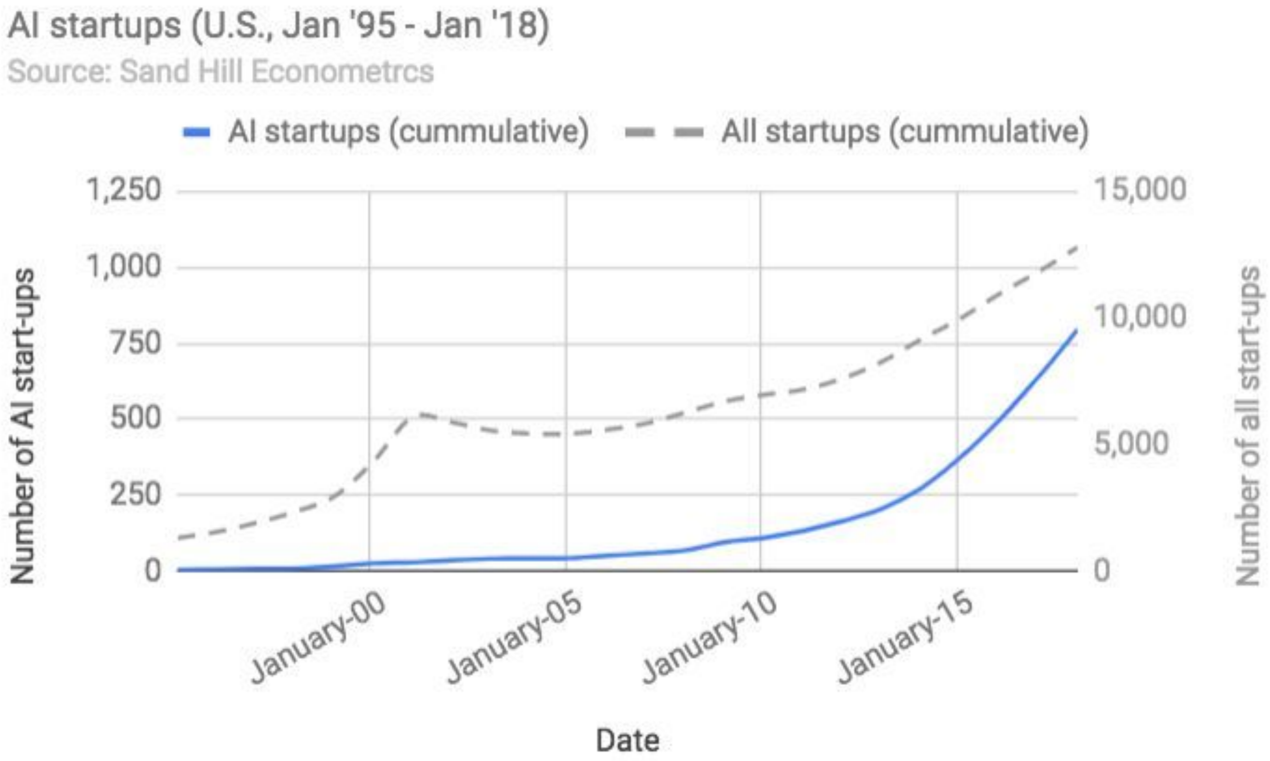
Catching up with a moving target



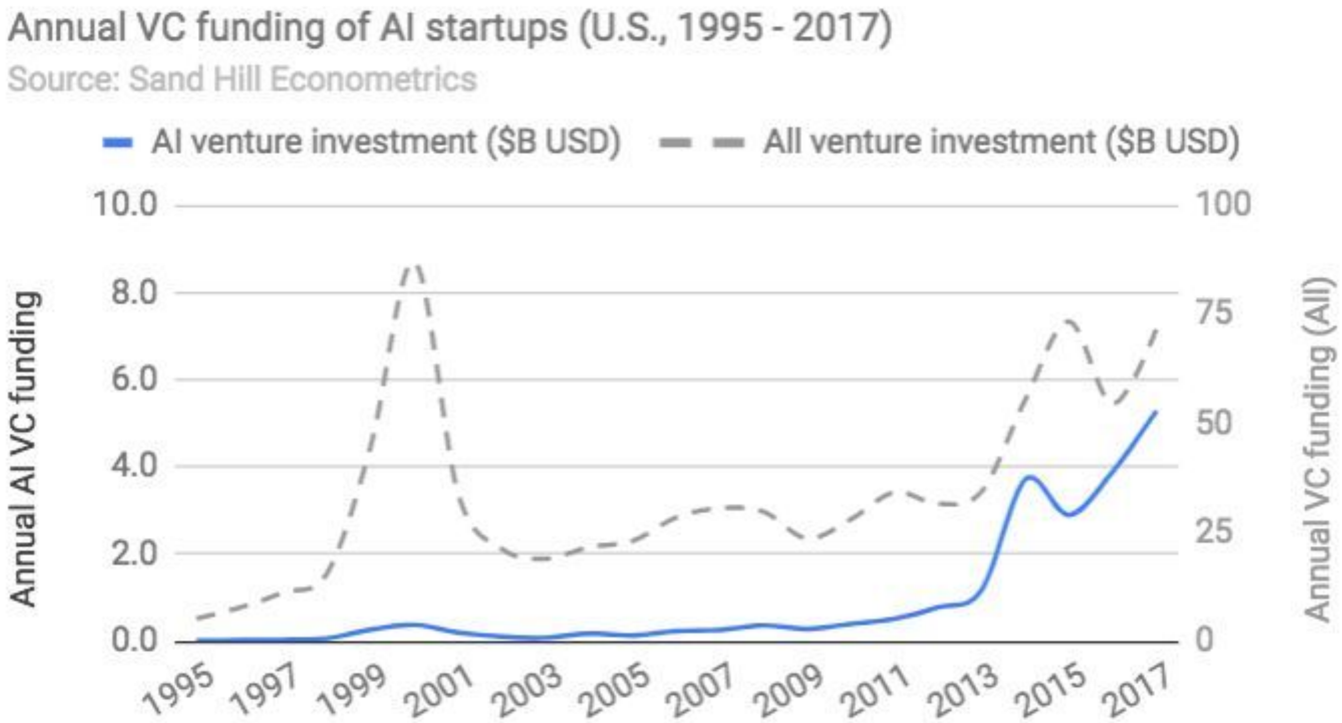
Catching up with a moving target



Catching up with a moving target



Catching up with a moving target



Keeping up with a moving target

So how do you learn and keep up with Artificial Intelligence and Machine Learning?

- Higher Education
- Local Communities - https://hsv-ai.com
- Global Communities - https://twimlai.com/
- Online Courses (see next slide)

Keeping up with a moving target

Ian Goodfellow - http://www.deeplearningbook.org/

Google - https://ai.google/education/

Machine Learning course from Stanford: https://www.coursera.org/learn/machine-learning

Intro to Tensorflow for Deep Learning from Google:
https://www.udacity.com/course/intro-to-tensorflow-for-deep-learning--ud187

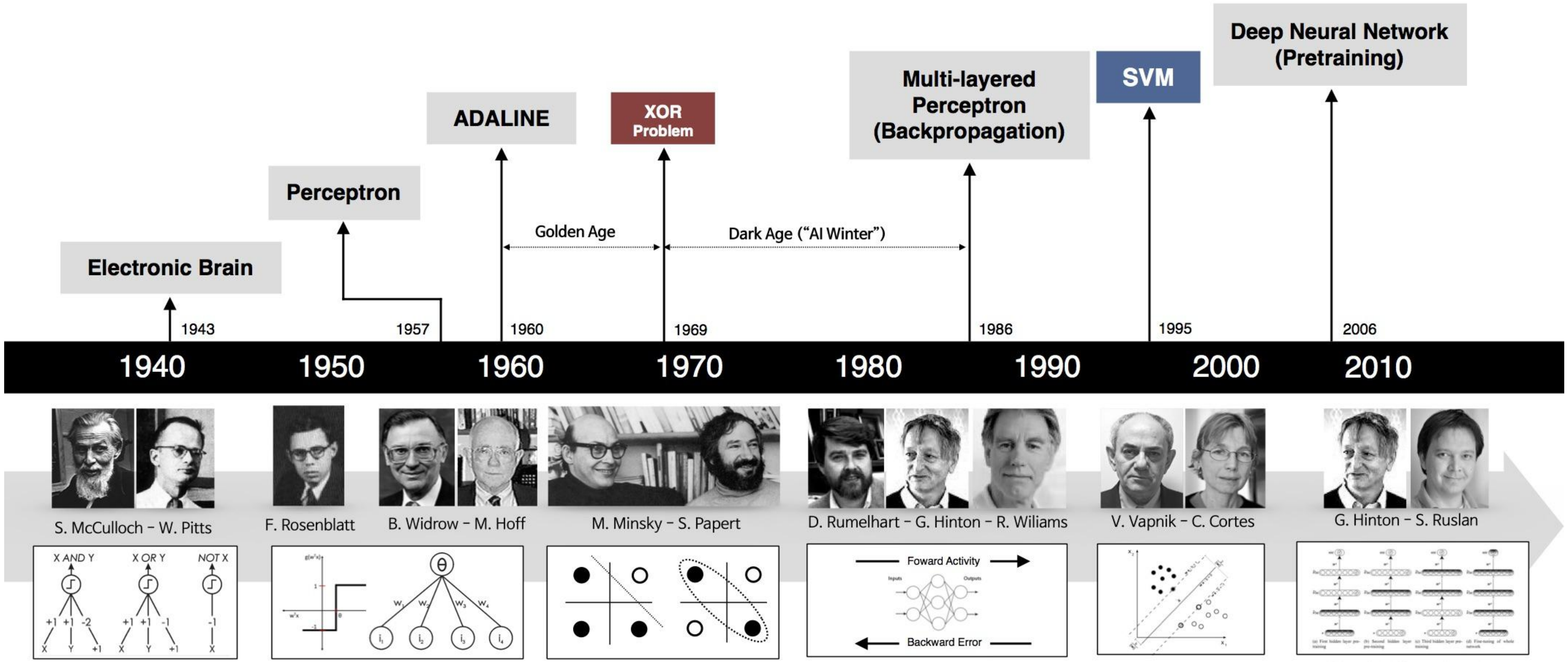
Machine Learning course from Columbia: https://www.edx.org/course/machine-learning

Fast.ai: https://www.fast.ai/

It’s time for time travel!

How about a shallow history of deep learning?

Deep Learning History



- Adjustable Weights
- Weights are not Learned

- Learnable Weights and Threshold

- XOR Problem

- Solution to nonlinearly separable problems
- Big computation, local optima and overfitting

- Limitations of learning prior knowledge
- Kernel function: Human Intervention

- Hierarchical feature Learning

Beginnings

Alan Turing - Computing Machinery and Intelligence (1950)

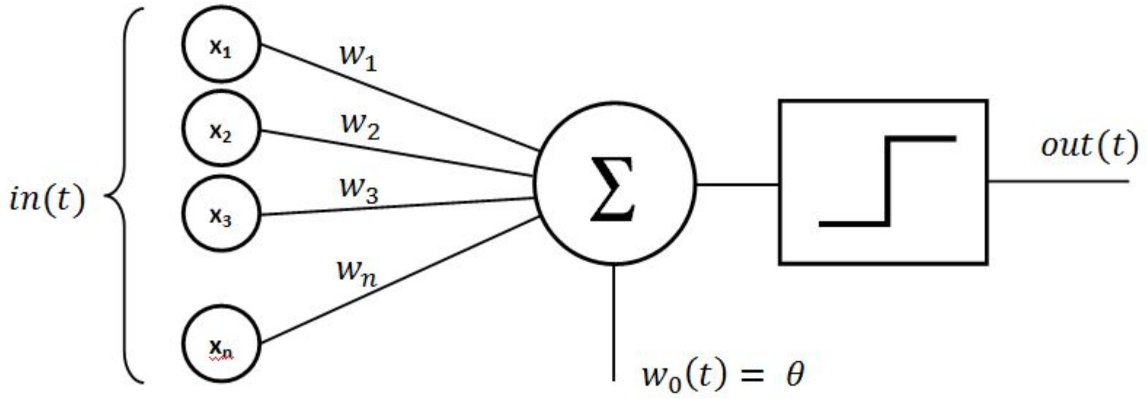
https://www.csee.umbc.edu/courses/471/papers/turing.pdf

This paper introduce the concept of what is now known as the "Turing Test"

The state of the art at the time was the Threshold Logic Unit - mostly based on current knowledge of how neurons were thought to work.

Perceptron

The perceptron algorithm was invented in 1957 at the Cornell Aeronautical Laboratory by Frank Rosenblatt,funded by the United States Office of Naval Research The perceptron was intended to be a machine, rather than a program, and while its first implementation was in software for the IBM 704, it was subsequently implemented in custom-built hardware as the "Mark 1 perceptron". This machine was designed for image recognition: it had an array of 400 photocells, randomly connected to the "neurons". Weights were encoded in potentiometers, and weight updates during learning were performed by electric motors



Perceptron

Frank Rosenblatt and the Perceptron

AI Winter (1969)

Cause of AI Winter

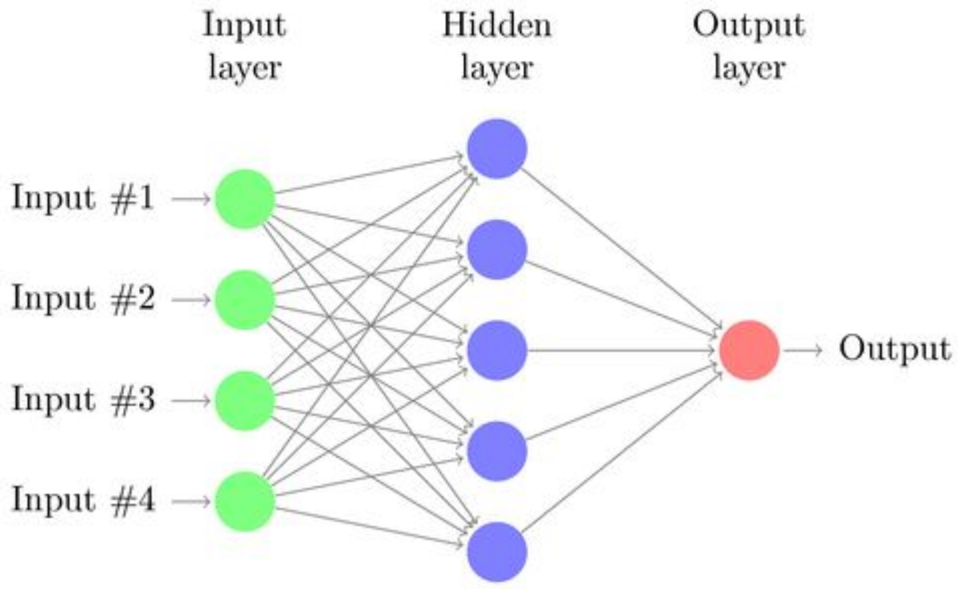
Perceptrons: an introduction to computational geometry is a book written by Marvin Minsky and Seymour Papert and published in 1969.

It offered a mathematical proof that the perceptron could not approximate an XOR function given an infinite training set.

I never liked XOR anyway.

Multilayer Perceptrons

By stacking several layers of perceptrons, researchers were able to overcome the XOR problem.



Backpropagation (1986)

Geoff Hinton, along with David Rumelhart and Ronald Williams, published a paper entitled “Learning representations by back-propagating errors”

https://www.iro.umontreal.ca/~vincentp/ift3395/lectures/backprop_old.pdf

This provided a mechanism for training multilayer perceptron networks.

Also about this time, the **universal approximation theorem** states that a feed-forward network with a single hidden layer containing a finite number of neurons (i.e., a multilayer perceptron), can approximate continuous functions on compact subsets of $\mathbf{R}^n$, under mild assumptions on the activation function.

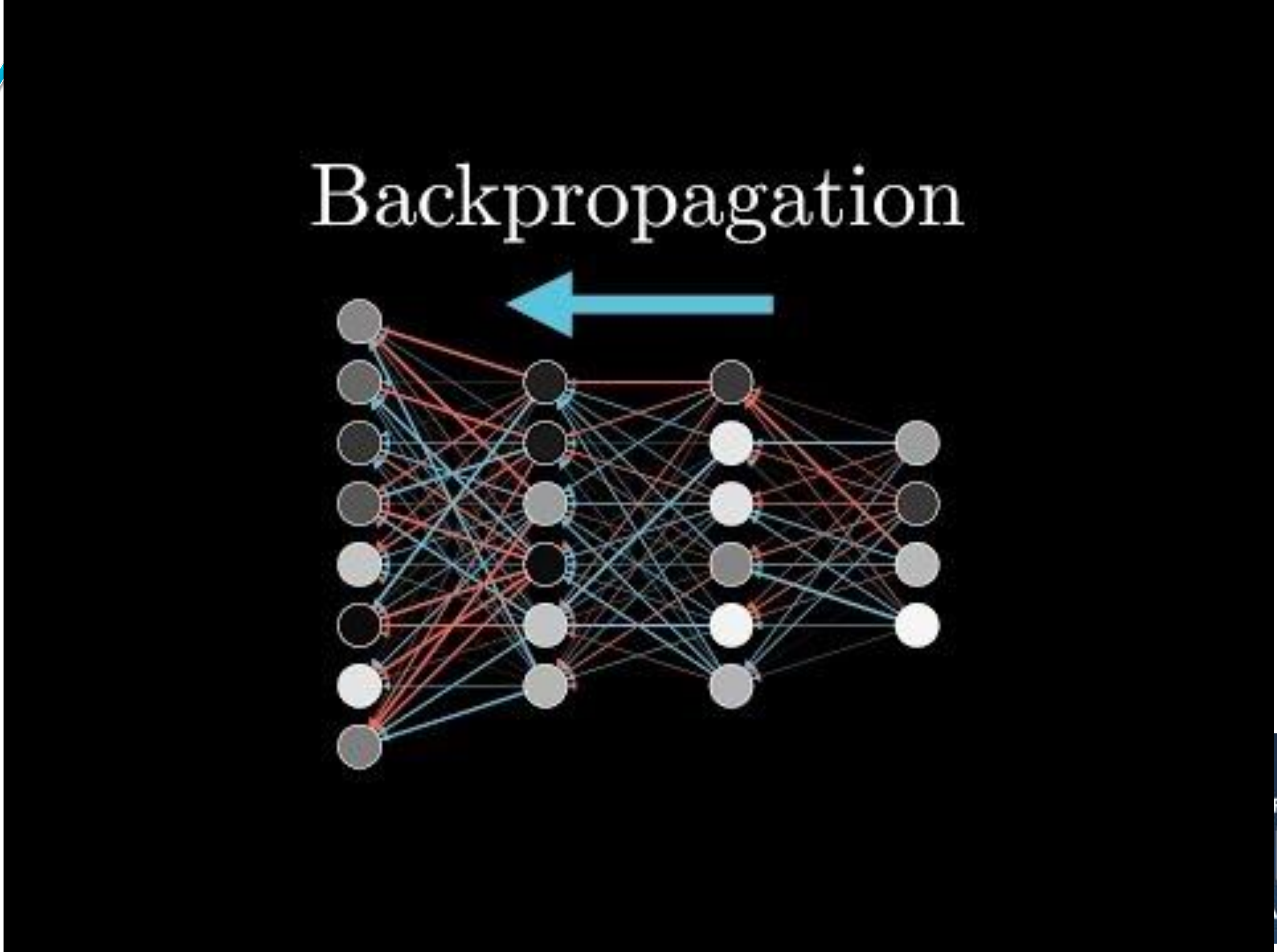
Backpropagation (1986)

Geoff Hinton, along with David Rumelhart and Ronald Williams, published a paper entitled "Learning representations by back-propagating errors"

https://www.iro.umontreal.ca/~vincentp/ift3395/lectures/backprop_old.pdf

This provided a mechanism for training multilayer perceptron networks.

Also about this time, the **universal approximation theorem** states that a feed-forward network with a single hidden layer containing a finite number of neurons (i.e., a multilayer perceptron), can approximate continuous functions on compact subsets of $\mathbf{R}^n$, under mild assumptions on the activation function.


Backpropagation

Learning before “Deep”

Gradient based learning (1998) - Yann Lecun - http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf

CNN from Yann Lecun (AT&T Bell Labs) could recognize handwritten digits.

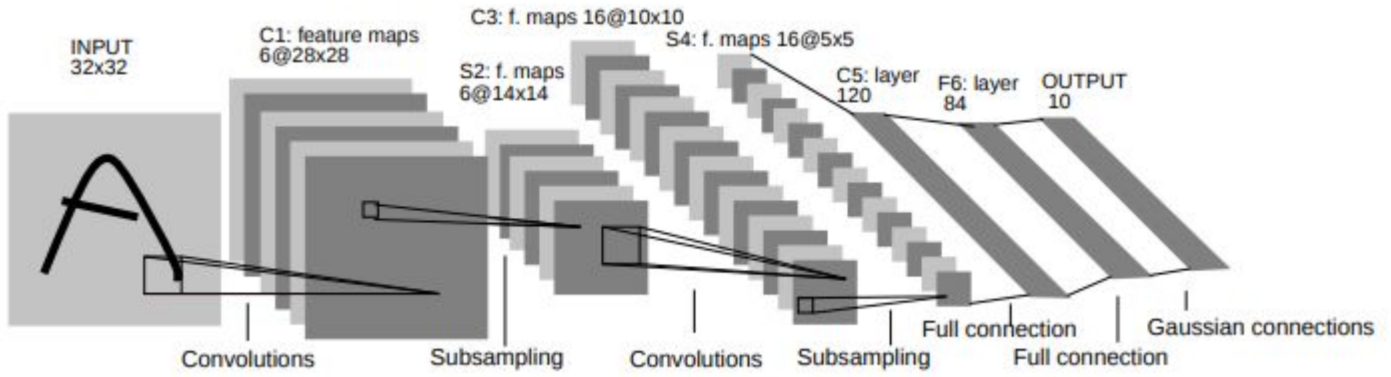


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Rolling in the Deep

Deep Learning (2006) - Again with Geoff HInton. The idea was to train a simple 2-layer unsupervised model like a restricted boltzman machine, freeze all the parameters, stick on a new layer on top and train just the parameters for the new layer.

Using this strategy, people were able to train networks that were deeper than previous attempts, prompting a rebranding of ‘neural networks’ to ‘deep learning’.

Filling the void with Hardware and Data

Imagenet (2009) - millions of labeled images created and published by

Fei-Fei Li at Stanford

MNIST - Handwritten digits

Google House Numbers from street view

Flickr 30k Image dataset

GPU - used for multi-core floating point calculation

Custom chipsets from Intel (Nervana) https://ai.intel.com/ and NVIDIA

Exponential Improvements

Alexnet (2012) - Won the Large Scale Visual Recognition Challenge(LSVRC) with an error rate 10% lower than the previous year. Used dropout to reduce overfitting and a rectified linear activation unit (ReLU)

Generative Adversarial Networks (2014) - Ian Goodfellow

Gated Recurrent Unit (2014) - Kyunghyun Cho et al

Speech recognition:

https://venturebeat.com/2017/05/17/googles-speech-recognition-technology-now-has-a-4-9-word-error-rate/

https://techcrunch.com/2017/08/20/microsofts-speech-recognition-system-hits-a-new-accuracy-milestone/

https://hacks.mozilla.org/2017/11/a-journey-to-10-word-error-rate/

Machine Learning Overview

We’ll cover several areas of Machine Learning algorithms / techniques along with some sample applications.

- Regression / Prediction
- Classification
- Clustering

Regression

Regression is a way to predict an outcome based on a set of variables, after training using a set of labeled data.

It is often the first ML method that people learn - also may be the one fallback used.

Regression

Regression is used to answer questions like:

“Given current income, savings, # of late payments, # of kids, marital status, etc—will this person default on a loan of $$$ amount?”

“Given square footage, age, school district, zip code, price of nearby homes—what is the value of a house?”

Regression

There are two parts of the question that will determine which type of regression would be the most useful:

1. The expected answers—Is this a yes/no question? If you plotted the actual answers from historical data, do they look like a straight line or some type of curve? The type of regression method is generally dependent on the type of answer expected.

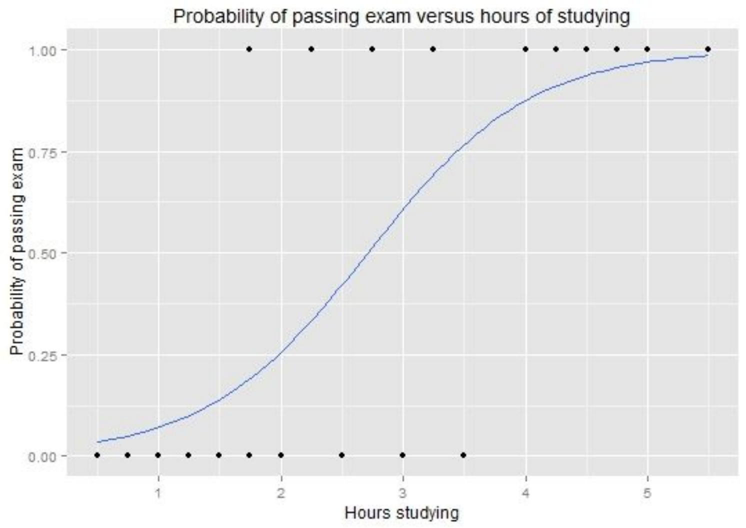
2. The input values—Are they independent of each other (square footage and age)? Are they highly related (zip code and school district)? Based on the dependence between input values, special cases of regression methods can take this into account.

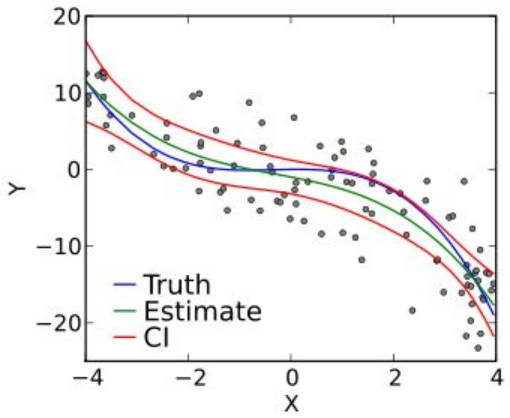
Regression

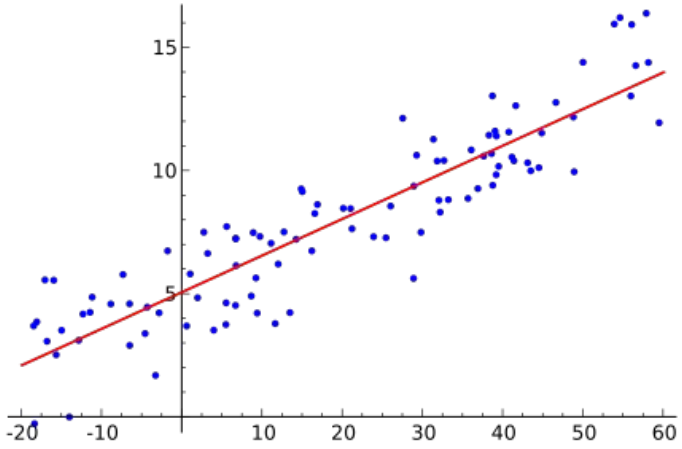
Logistic Regression is used for yes/no answers.

Linear Regression is used when answers follow a straight line if you put them on a graph.

Polynomial Regression is used when answers follow a curve if you put them on a graph.







Regression

Regression Tips:

Always check the data you're training on to see if it is a singular set of data and not actually separate groupings.

For example, a collection of housing data that includes both Madison City and rural Madison County will provide two distinct groupings for price given the same square footage.

Classification

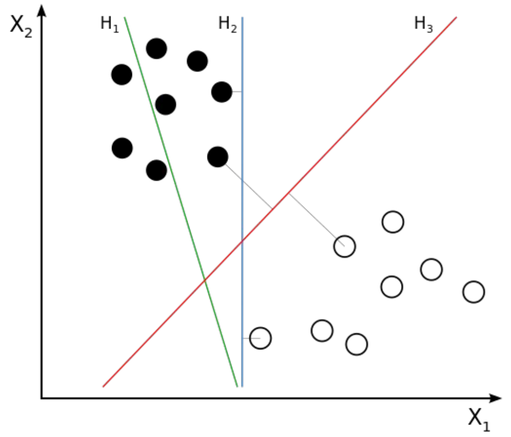
Classification is used to take a piece of data and determine which classification it should belong to, from a set of known classifications.

Typically, there are two types of classification problems:

- Binary - there are only two classes - like “Spam or Not Spam”. This is the same as a yes or no answer. This can usually be solved through Logistic Regression.
- Multiple Classes - potentially a large number of classes to choose from. This is used to answer questions like “What part of speech is the word ‘blue’ in the sentence ‘Roses are red, violets are blue’ ?”

Classification

- **Linear Regression** - if the classifications can be grouped separately along some arbitrary line, then linear regression may be the best and most efficient way to solve the problem.
- **Support Vector Machine** - attempts to find a line between two groups in training data.
- **Decision Tree** - used to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). Thank you Wikipedia!
- **Naive Bayes** - looks a lot like probability. Actually, it is probability - make sure variables are independent when using it though.



Classification

Let’s take a walk through an example of classification using several different methods:

https://colab.research.google.com/github/HSV-AI/presentations/blob/master/2019/190213/MUSK%20Classification.ipynb

Clustering

Clustering can be thought of as the opposite of classification. The classifications (or groupings) are not known ahead of time, and are discovered by using machine learning.

Some algorithms used here need to be given the number of clusters to break the data into, while others take other parameters and identify some set of clusters based on them.

Clustering

Here are a few of the many algorithms used for clustering:

- **K-Means** - takes a number of clusters as an input, and computes the cluster boundaries based on the distance between points.
- **Affinity Propagation -** also known as “Nearest Neighbor”. Takes a set of parameters and computes the cluster boundaries based on the nearest neighbor graph distance.

There are a lot of other algorithms as well, but all seem to use either the distance between points or nearest neighbor graph distance.

Deep Learning Techniques

The two techniques that we’ll cover today are the Convolutional Neural Network (CNN) and the Recurrent Neural Network (RNN).

Convolutional Neural Net

A great way to get started in Deep Learning is through the free Fast.ai courses. We’ll walk through an example of their first course that covers image recognition with a CNN.

Use this to set up colab to use fast.ai:

```
!curl -s https://course.fast.ai/setup/colab | bash
```

Fast.ai Lesson 1

https://colab.research.google.com/github/fastai/course-v3/blob/master/nbs/dl1/lesson1-pets.ipynb

Convolutional Neural Net

While the Fast.ai libraries are very useful at getting something to work rather quickly, there are times when you want to go deeper and do things by hand.

Tensorflow Lesson

https://colab.research.google.com/github/tensorflow/examples/blob/master/courses/udacity_intro_to_tensorflow_for_deep_learning/l03c01_classifying_images_of_clothing.ipynb

Huntsville AI

https://hsv-ai.com/meetups/190227_amazon_rekognition/

Deep Neural Net Benchmarks

If you noticed, the fast.ai course downloaded a pre-trained model for the ResNet. Here is a benchmark that measured current training time and cost for various Deep Learning challenges:

https://dawn.cs.stanford.edu/benchmark/

Recurrent Neural Networks

From WikiPedia:

A recurrent neural network (RNN) is a class of artificial neural network where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition.

Let’s walk through a discussion we had earlier this year with Huntsville AI:

https://colab.research.google.com/github/HSV-AI/presentations/blob/master/2019/190612_Recurrent_Neural_Networks.ipynb

https://colab.research.google.com/github/HSV-AI/presentations/blob/master/2019/190626_RNN_TimeSeries_Part3.ipynb

Other Cool Stuff!

Check out page 60 of the 2018 AI Index Report for more.
http://cdn.aiindex.org/2018/AI%20Index%202018%20Annual%20Report.pdf

Robot learning to run:
https://twitter.com/eron_gj/status/967672260147470336

Other Cool Stuff!