

What is Machine Learning

Jake Searcy

Assistant Research Professor of Data Science

Prompt: Why is it important for bench scientists to learn about machine learning?

GPT-3

There are many reasons why bench scientists should learn about machine learning. Machine learning can help bench scientists automate data analysis, find patterns in data more quickly, and make better predictions. Additionally, machine learning is a rapidly growing field, and bench scientists who are familiar with machine learning will be better positioned to take advantage of new opportunities as they arise.

Generative Pre-trained Transformer 3 (GPT-3) is an autoregressive language model that uses deep learning to produce human-like text.

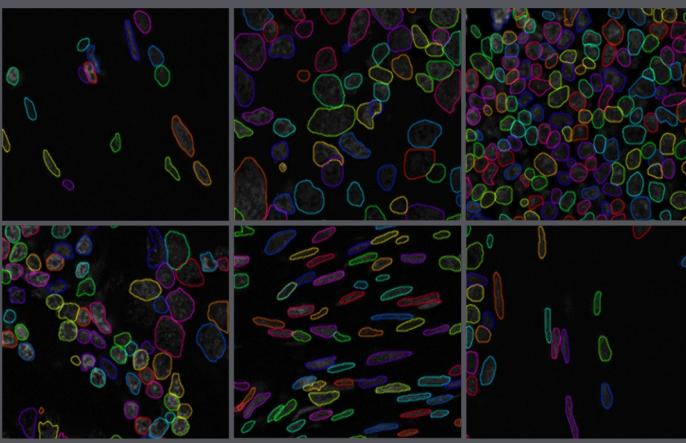
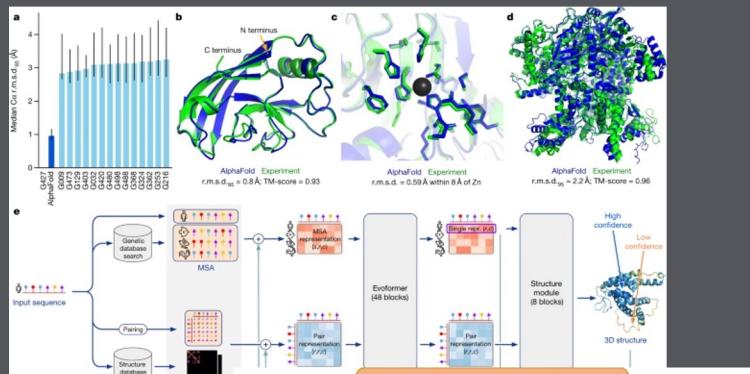
Prompt: Some of the most useful applications of Machine Learning in the Life Sciences are

GPT3

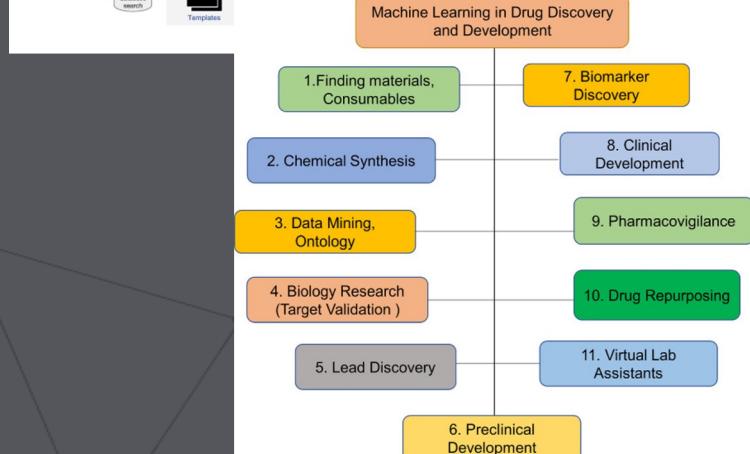
1. Diagnosis and treatment of disease
2. Development of new drugs and therapies
3. Personalized medicine
4. Predictive analytics for clinical trials
5. Population health management

Alpha Fold - Protein Folding :

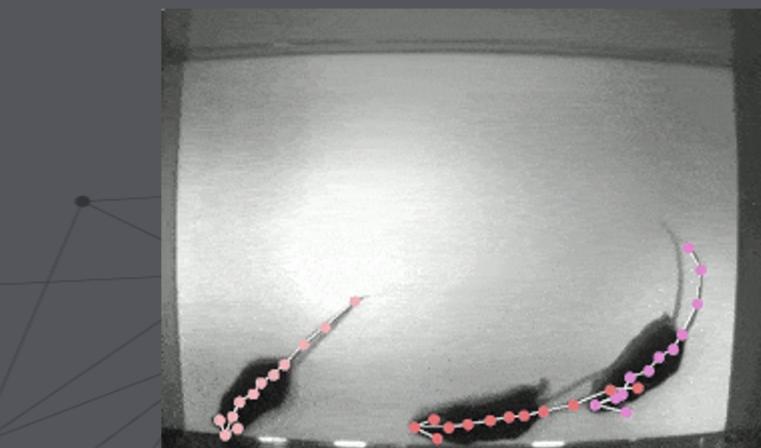
<https://www.nature.com/articles/s41586-021-03819-2>



<https://michaellee1.github.io/CellSegSite/index.html>



<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8356896/figure/Fig1/>



<http://www.mackenziemathislab.org/deeplabcut>

What is Machine Learning

- **Machine learning (ML) is a lot like programming**
 - It's a series of techniques for telling a computer how to do something
 - Instead of writing an algorithm with code you create a system that can learn an algorithm from data
- **Often is not the right choice**
 - Training an ML model to add numbers
 - Would take a lot of work
 - And would not be as accurate as a simple program
- **Perfect when you have data but no algorithm**

Besides quick coding fingers, look for a personality that can cope with failure.
You almost never know what you're doing, even if you think you do.

--[Cassie Kozyrkov](#) Chief Decision Intelligence Engineer at Google



Thought Experiment

How would you write an algorithm to tell the difference between cats and dogs



O

Still an Algorithm

- Algorithms can have bugs
 - Bugs in ML often occur in data



O

How Does it Work

THIS IS YOUR MACHINE LEARNING SYSTEM?

YUP! YOU POUR THE DATA INTO THIS BIG
PILE OF LINEAR ALGEBRA, THEN COLLECT
THE ANSWERS ON THE OTHER SIDE.

WHAT IF THE ANSWERS ARE WRONG?

JUST STIR THE PILE UNTIL
THEY START LOOKING RIGHT.



O

How Does it Work

- All ML Algorithms Require
 - Data
 - Goal
- Generally ‘learn’ through repeated trial and error
- Selecting the Goal and Data are often the most important parts of an ML project
- **Often understood the least by ML experts**



Data - Supervised

- Just a series of example inputs and a series of example outputs
- What are the inputs and outputs here?



O

Data - Supervised

Pixels



OpenPose

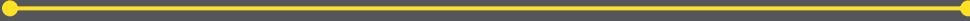
```
array([[[[ 59,   62,   63],  
        [ 43,   46,   45],  
        [ 50,   48,   43],  
        ...,  
        [158,  132,  108],  
        [152,  125,  102],  
        [148,  124,  103]],  
  
       [[ 16,   20,   20],  
        [  0,    0,    0],  
        [ 18,    8,    0],  
        ...,  
        [123,   88,   55],  
        [119,   83,   50],  
        [122,   87,   57]],  
  
       [[ 25,   24,   21],  
        [ 16,    7,    0],  
        [ 49,   27,   8],  
        ...,  
        [118,   84,   50],  
        [120,   84,   50],  
        [109,   73,   42]],  
  
       ...]
```

Locations
X,Y

```
array([[ 0.25674411,  0.01283803],  
      [ 0.753803 ,  0.56980359],  
      [ 0.26282471,  0.40350747],  
      [ 0.47644531,  0.79709066],  
      [ 0.84001543,  0.94559807],  
      [ 0.07605252,  0.4048905 ],  
      [ 0.14939603,  0.27401423],  
      [ 0.82483375,  0.63122051],  
      [ 0.07184129,  0.00590745],  
      [ 0.99608272,  0.8684909 ],  
      [ 0.00902896,  0.34306335],  
      [ 0.85554453,  0.45699895],  
      [ 0.52146749,  0.66698339],  
      [ 0.55767337,  0.61783118],  
      [ 0.10308075,  0.4990765 ],  
      [ 0.50209006,  0.47997771],  
      [ 0.73032569,  0.75559176],  
      [ 0.46540034,  0.90332933],  
      [ 0.26248118,  0.59531095],  
      [ 0.47051964,  0.30168414]])
```

O

Example 2

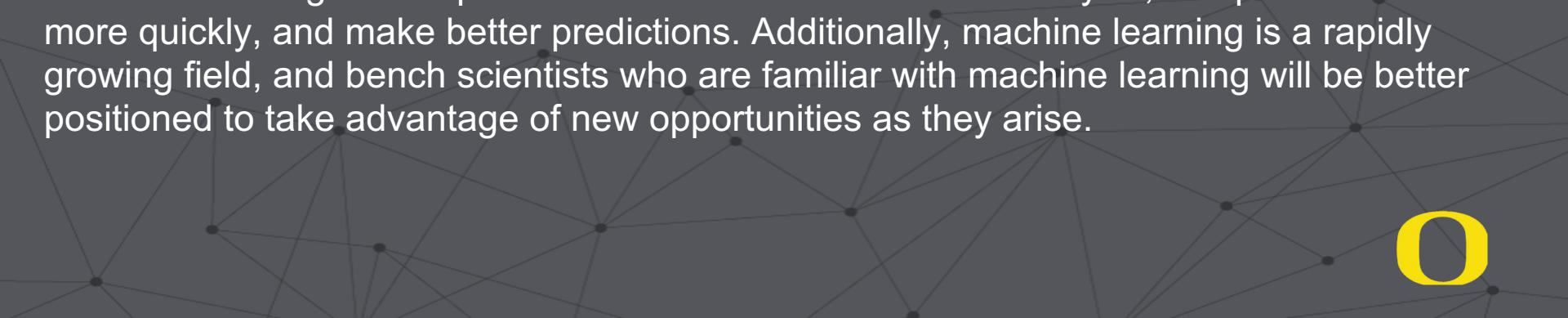


- What are the inputs and outputs here?

Prompt: Why is it important for bench scientists to learn about machine learning?

GPT-3

There are many reasons why bench scientists should learn about machine learning. Machine learning can help bench scientists automate data analysis, find patterns in data more quickly, and make better predictions. Additionally, machine learning is a rapidly growing field, and bench scientists who are familiar with machine learning will be better positioned to take advantage of new opportunities as they arise.



Example 3

An astronaut Teddy bears A bowl of soup

that is a portal to another dimension that looks like a monster as a planet in the universe

knitted out of wool spray-painted on a wall made out of plasticine



<https://openai.com/dall-e-2/>

Supervised Learning

- Lots of ML algorithms for utilizing all kinds of data in all kinds of combinations
- This can be fun, creative and useful and creating the training datasets is the first step.



Common Vocab

- x's - often used to represent input data
- y's - often used to represent target labels
- From x predict y ($P(y|x)$)
- Classification
 - y is a class label (cat,dog etc.)
- Regression
 - y is a continuous variable (location, miles per gallon etc.)



Supervised Goals

- Classification
 - y's - often used to represent target labels
 - From x predict y ($P(y|x)$)
- Classification
 - y is a class label (cat,dog etc.)
- Regression
 - y is a continuous variable (location, miles per gallon etc.)



Unsupervised Learning

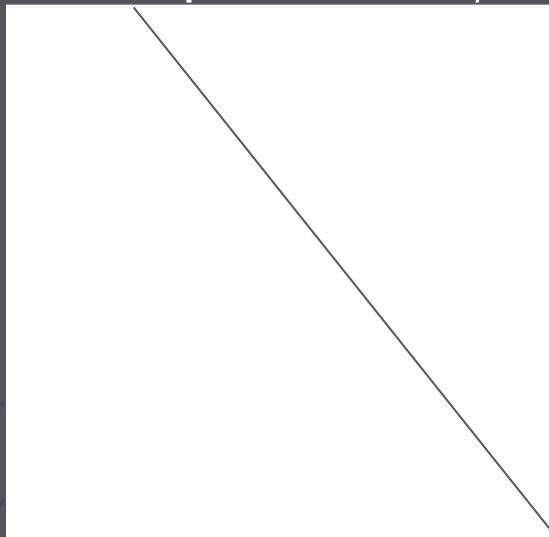
- Just examples **No-Labels**
 - Dimensionality Reduction
 - Your data has too many variables
 - Clustering
 - You want to group your data into categories



Dimensionality Reduction

Picture of a Line

(500x500 pixels=250,000 Numbers)



Equation:

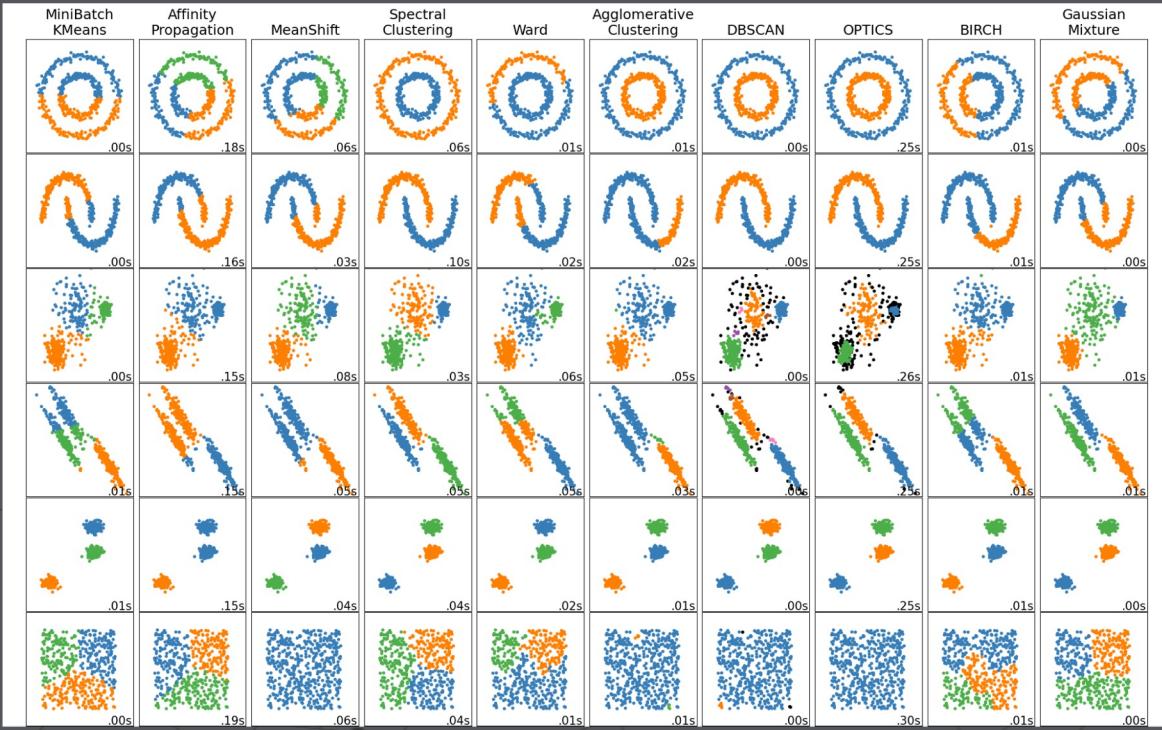
$y=m*x+b$ (2 numbers)

Both have the same information in different formats

It's often a lot easier to analyze 2 numbers than 250,000



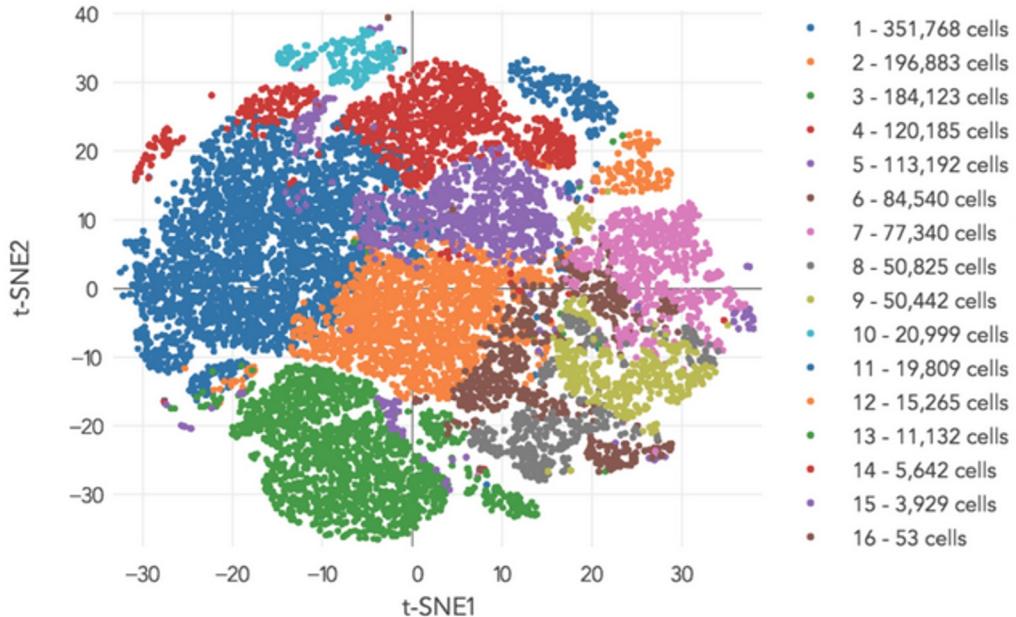
Clustering



Group your data points together



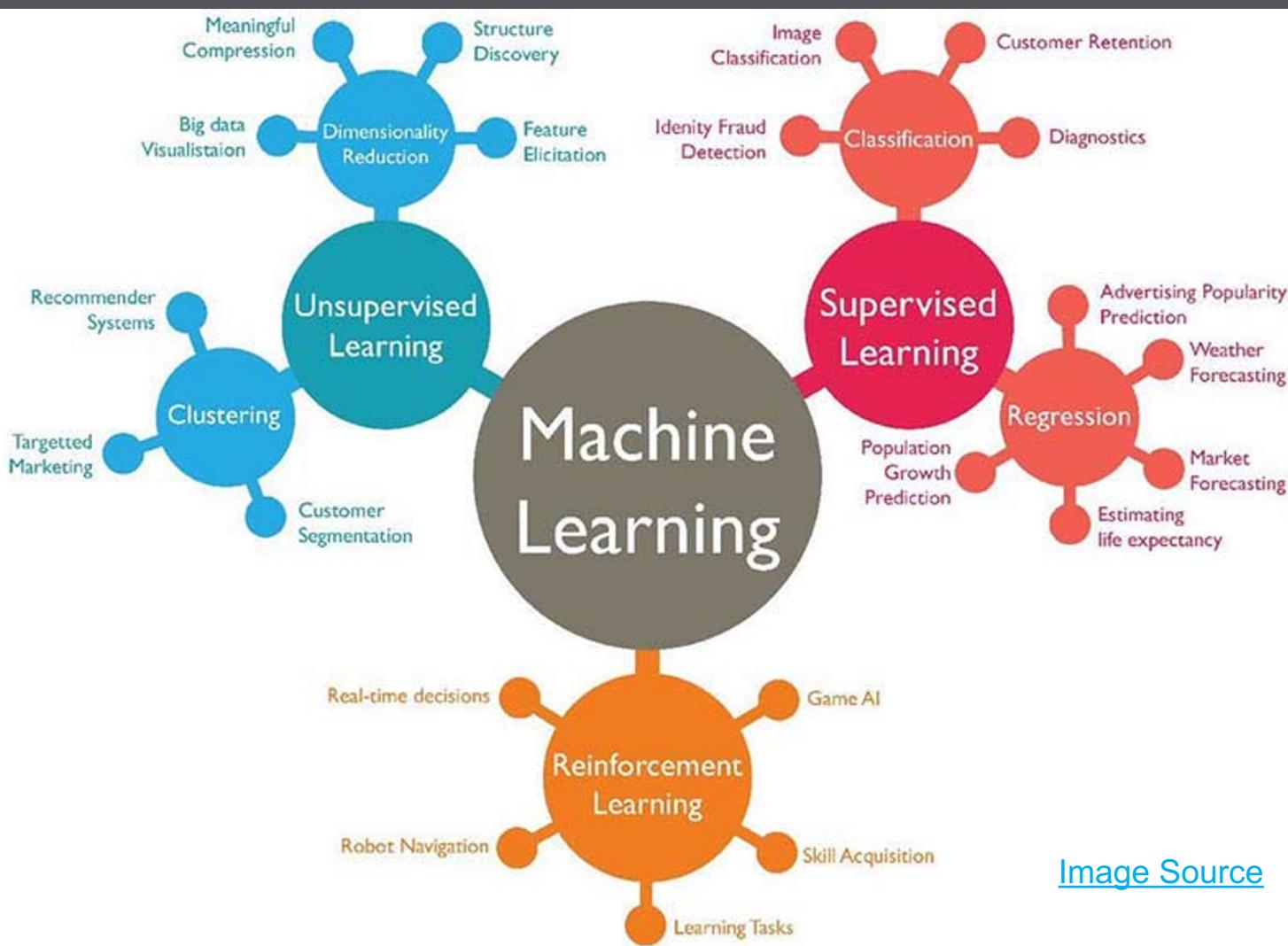
Both at once



The 10x Genomics 1.3 million cell dataset visualized in 2 dimensions using t-stochastic neighbor embedding and colored based on computed clusters.

<http://data-science-sequencing.github.io/Win2018/lectures/lecture19/>





[Image Source](#)

O

Algorithms

- Algorithms vary based on there goal, but also based on the kind of data they can use
- Structured Data
 - Fixed number of features
- Unstructured Data
 - Text
 - Images
 - Movies



Traditional ML

- Commonly used algorithms primarily using structured data
 - Random Forests
 - Support Vector Machines (SVM)
 - K-Means Clustering
 - Principal Component Analysis



Feature Engineering

- A lot of traditional ML relies on extracting structured data from unstructured data
- Example Bag of words
 - Raw: “This is a sentence for a test”
 - Structured word counts:

[This : 1, is : 1 , a : 2 , sentence: 1, dog: 0, other :2]



Deep Learning

- Primarily used with unstructured data
- No Feature Engineering
- Built with Layers
 - Think lego blocks that can be used to utilize different types of data

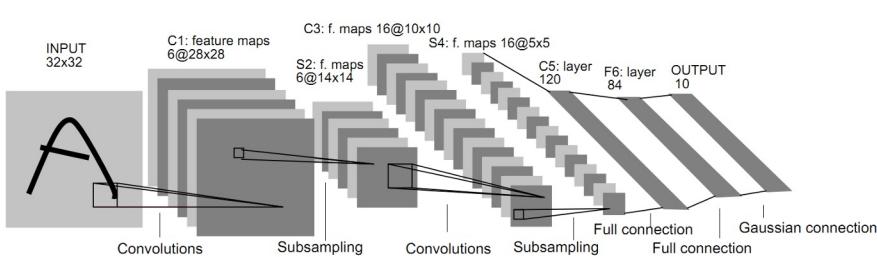
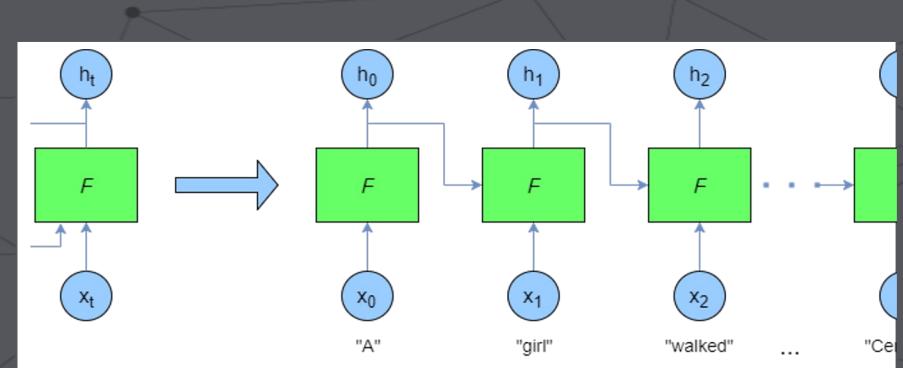
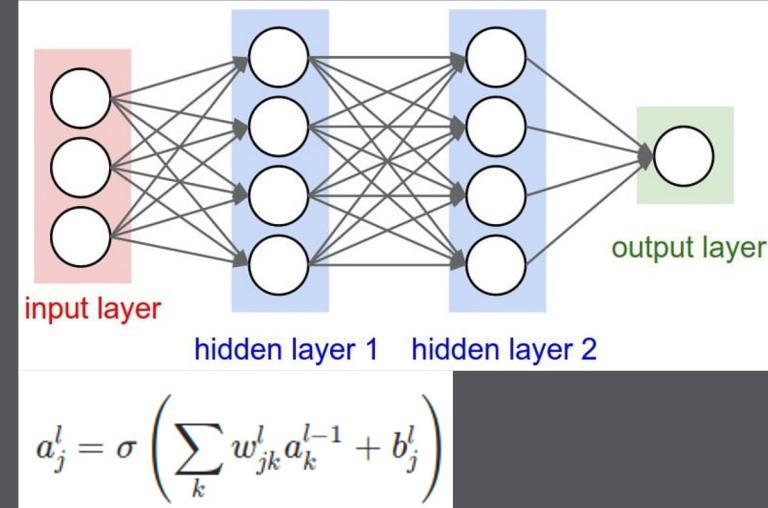


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.



DNN Advantages

- Often can lead to better results than human crafted algorithms
 - Provided the data is sufficient
- Can ingest huge varieties of raw data with limited or no preprocessing
 - Text
 - Images
 - Sounds
 - etc.

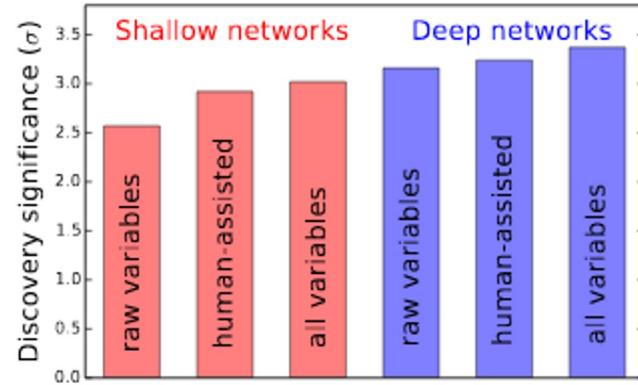
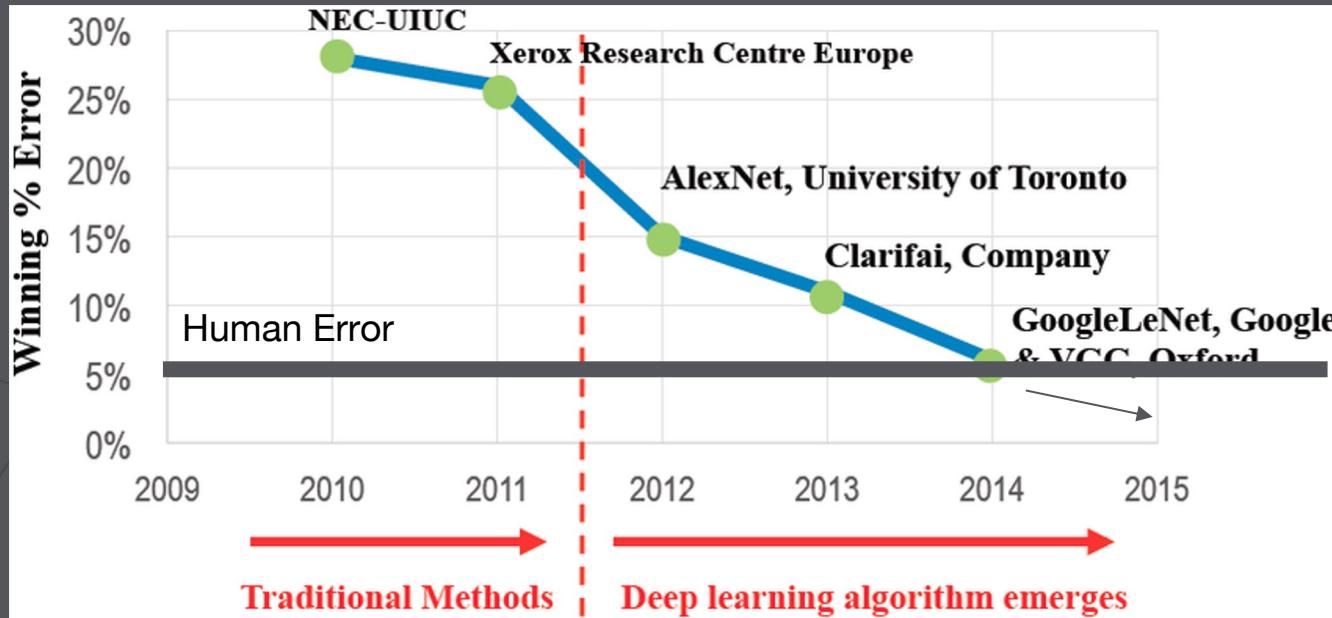


FIG. 5. Comparison of discovery significance for the traditional learning method (left) and the deep learning method (right) using the low-level variables, the high-level variables and the complete set of variables.

Images

Rule based algorithms are difficult to design images, deep learning wins
Image-net Challenge: Identify the subject of an image out of 1000 classes



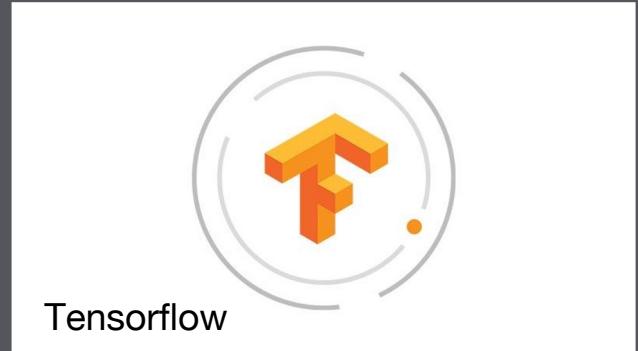
When should you use Deep Learning

- When you have a lot of data
 - DNN improve with data for longer than other traditional ML models
- When you have no traditional models
 - traditional models aren't preformant
- When your data is unstructured
 - Images, text, sound, etc.
- When your optimization goal can be computed quickly or approximately
 - Likelihoods can be approximated in mini-batches
 - Differential goals can be learned with back-prop
 - Differential goals can be
 - Non-Differential use Reinforcement learning



Software Tools

- Tons of implementations of basic ML tools
 - Matlab
 - R
 - STAT
 - SAS
- Most ML research and development is happening in two packages
 - Tensorflow
 - PyTorch(R has interfaces to both the above)



Tensorflow



O

Hardware Tools

- GPUS
 - Nvidia GPUs are the primary tools for machine learning
 - Primarily due to CUDA
- CPUS
 - Fine for smaller models, can't compete with GPUS for larger models
- Tensor Cores
 - Next generation of core highly optimized for tensor operations used in Deep Learning
 - Google's TPUs
 - Nvidia's RTX/Volta lines
 - FP16 only
- Others
 - FPGAs
 - Graph Cores and other custom chips
- As a user CPUS are a good place to start, and GPUs will be the main workhorse



HPC - Talapas



Standard Compute



GPU Nodes



Big Memory Nodes



Network Attached Storage (1.5PB)



Login Nodes



Conclusions

- ML Starts with Datasets
- We'll spend a lot of this week on the mechanics and pitfalls of creating AI Ready data
- ML algorithms work as universal function approximators
- This makes them tremendously flexible

