AI/ML BOOTCAMP

William Phelps

Christopher Newport University/Jefferson Lab

CHRISTOPHER NEWPORT UNIVERSITY

Jefferson Lab
Thomas Jefferson National Accelerator Facility

Announcements

Course Topics*

1 Semester DS/AI Course in 4 lectures
45 hours in 8 hours

- **Lectures 1 & 2**
  - **Lecture 1**
    - Introduction to workflow and tools (git/anaconda/pycharm)
    - Introduction to Data Science
    - Python Review with jupyter notebooks
  - **Lecture 2**
    - Data Visualization
    - Principles of data visualization from Edward Tufte
    - Pandas introduction
- **Lectures 3 & 4**
  - **Lecture 3**
    - Machine Learning Introduction
    - Regression/Curvefit
    - Overfitting/Underfitting
    - kNN classifier
    - MNIST/Handwritten digit classifier
  - **Lecture 4**
    - Convolutional Neural Networks
    - CIFAR 10 and CIFAR 100 datasets
    - YoloV11 Demo with classification/pose detection/image segmentation
    - Background blue
    - VDOT traffic counter

*Tentative

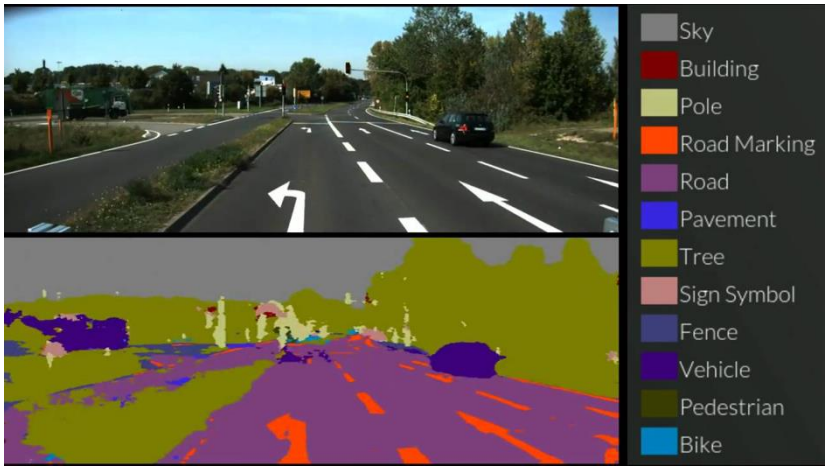
MACHINE LEARNING

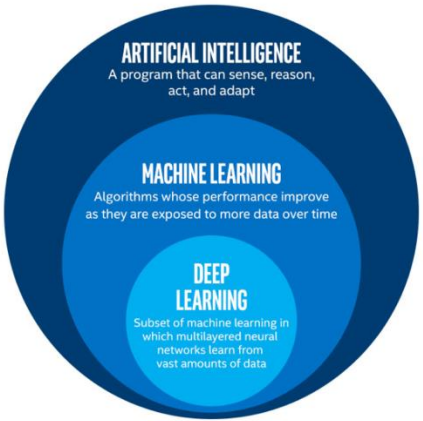
What is next?

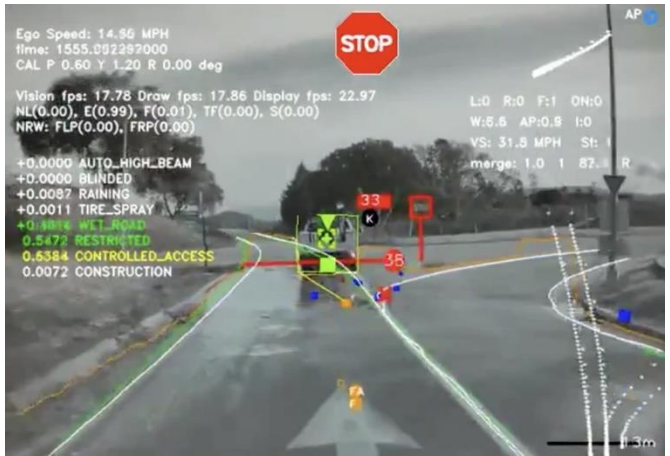
- Machine learning
  - Regression
    - Gradient descent fitting w/Scikit Learn
  - Classification
    - kNN Classification
  - Intro to Neural Networks
    - Keras MNIST dataset
    - Convolutional Neural Networks
    - CIFAR 10 and CIFAR100 datasets
  - More Deep Learning?

What is machine learning?

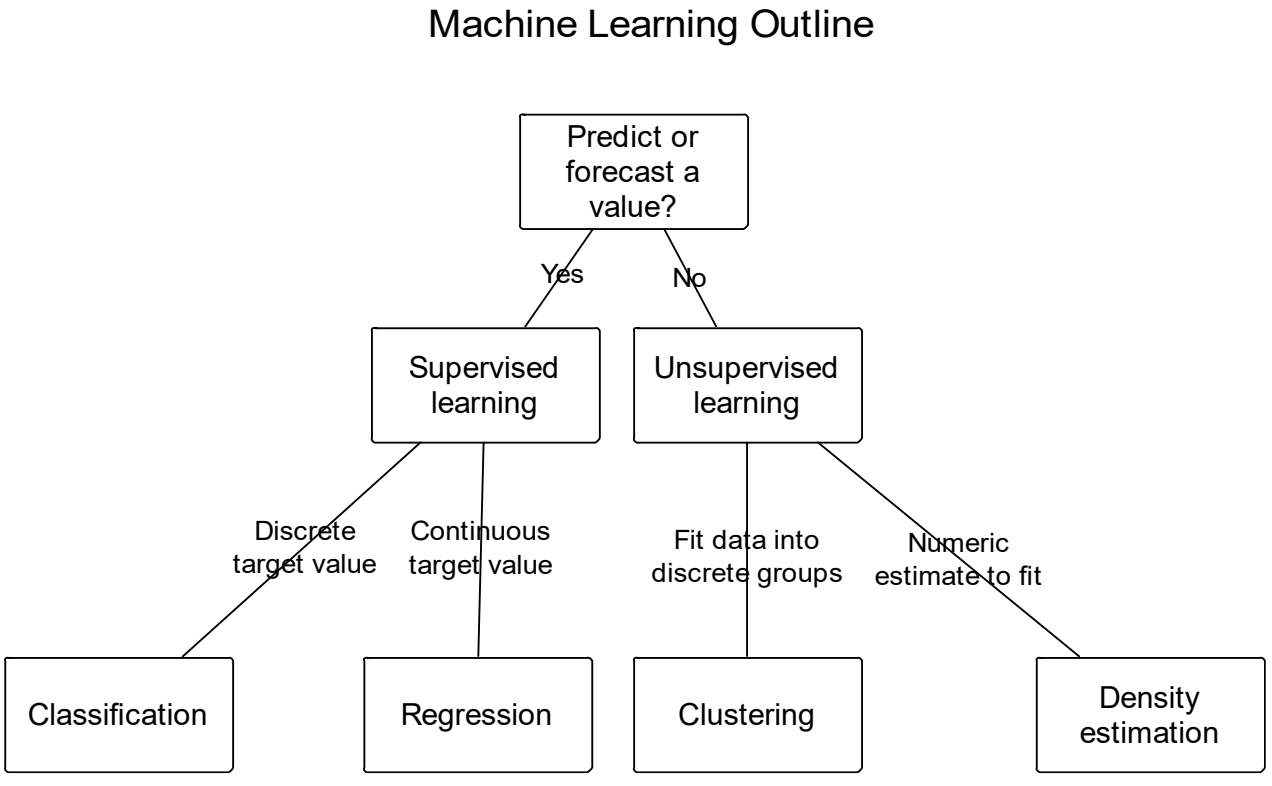
- Tom Mitchell: "A computer program is said to learn from experience *E* with respect to some class of tasks *T* and performance measure *P* if its performance at tasks in *T*, as measured by *P*, improves with experience *E*."
- Things learn when they change their behavior in a way that makes them perform better in the future.



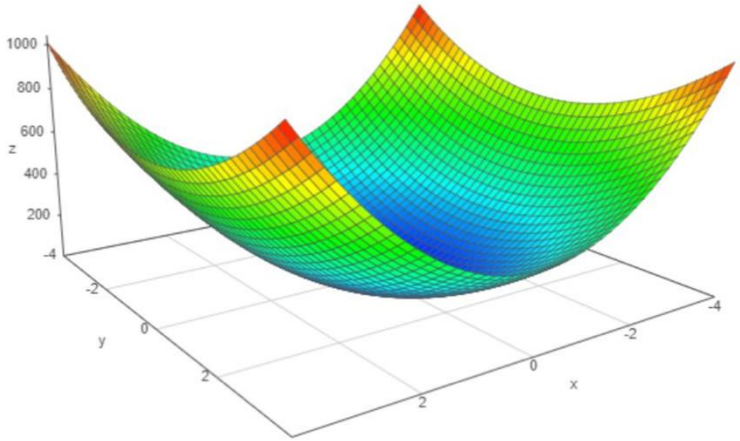


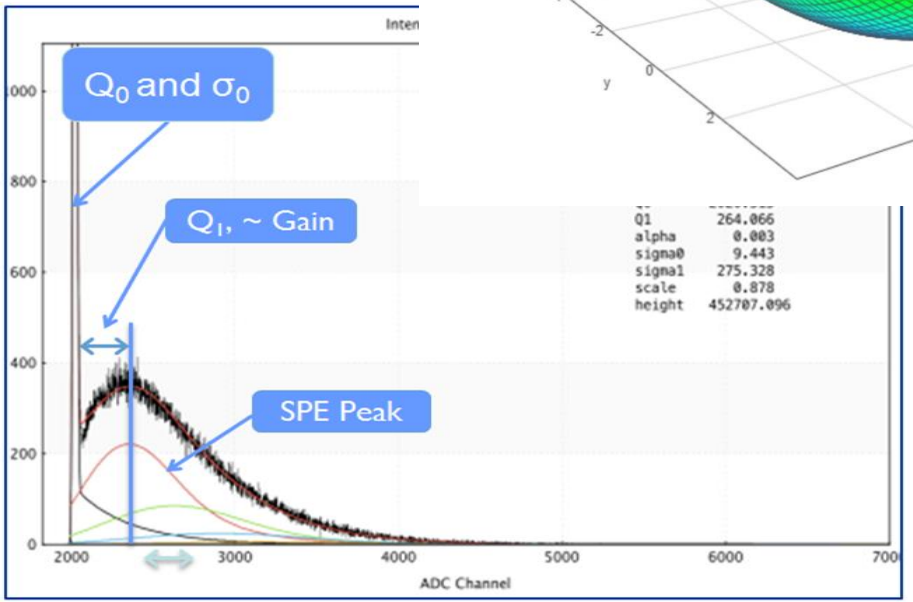


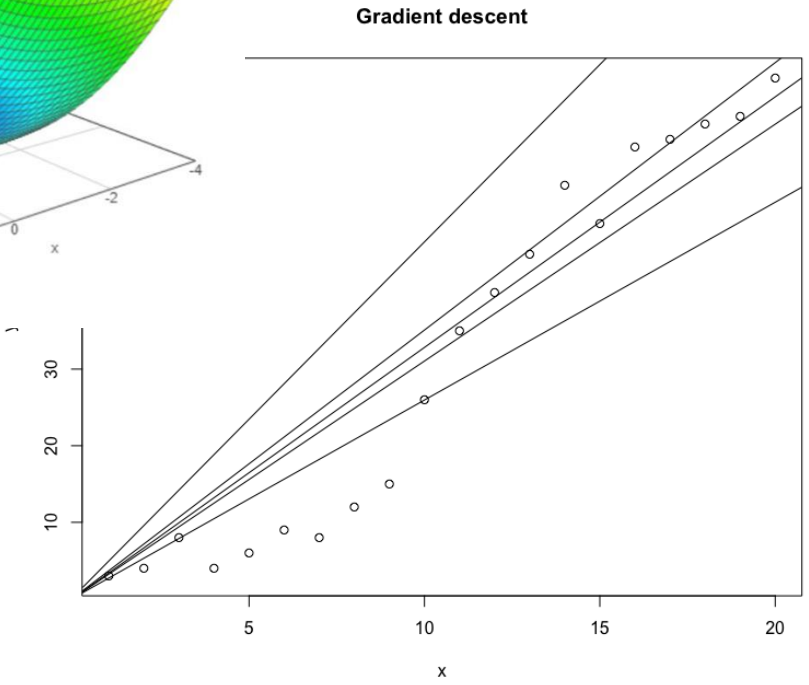
Outline of core ML problems



Machine Learning - Regression







Regression using gradient descent

- A process for modeling the relationship between variables of interest
- Gradient descent is a general approach in machine learning that could be used for solving linear regression

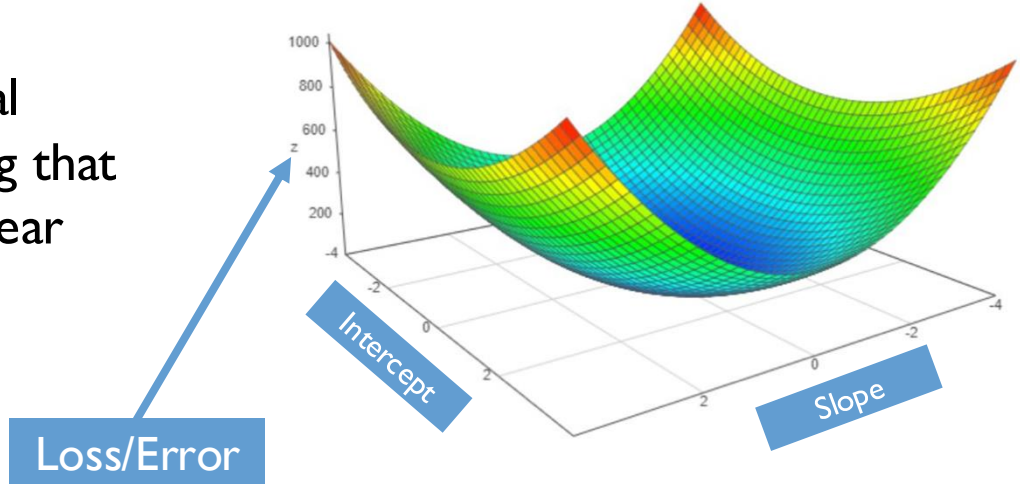
$$\chi_c^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

where:

$c =$ Degrees of freedom

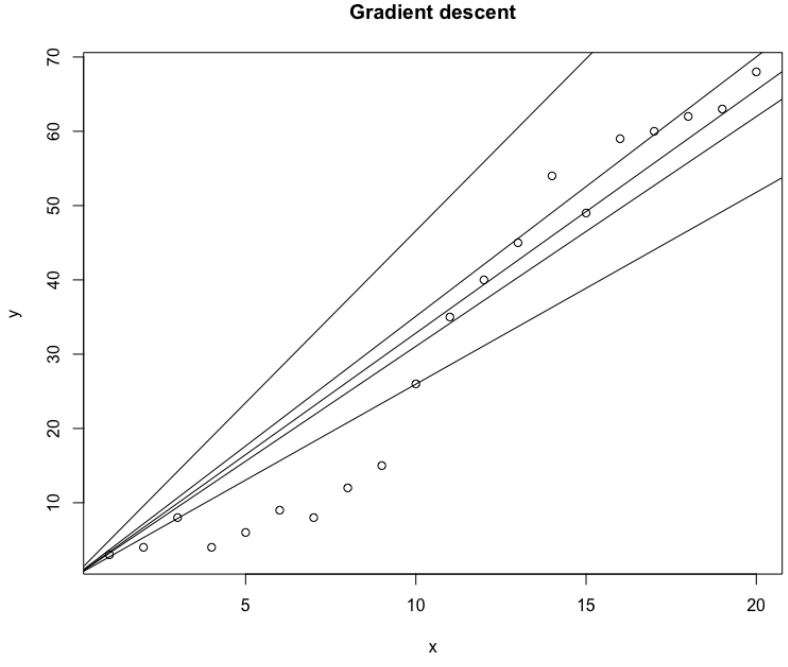
$O =$ Observed value(s)

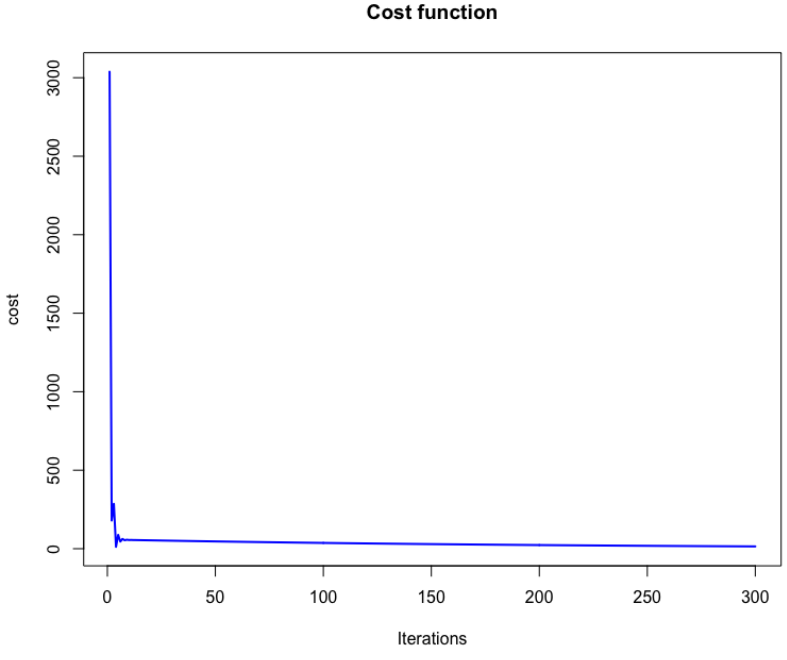
$E =$ Expected value(s)



Error surface for various lines created using linear regression (x represents slope, y represents intercept and z is the error value).

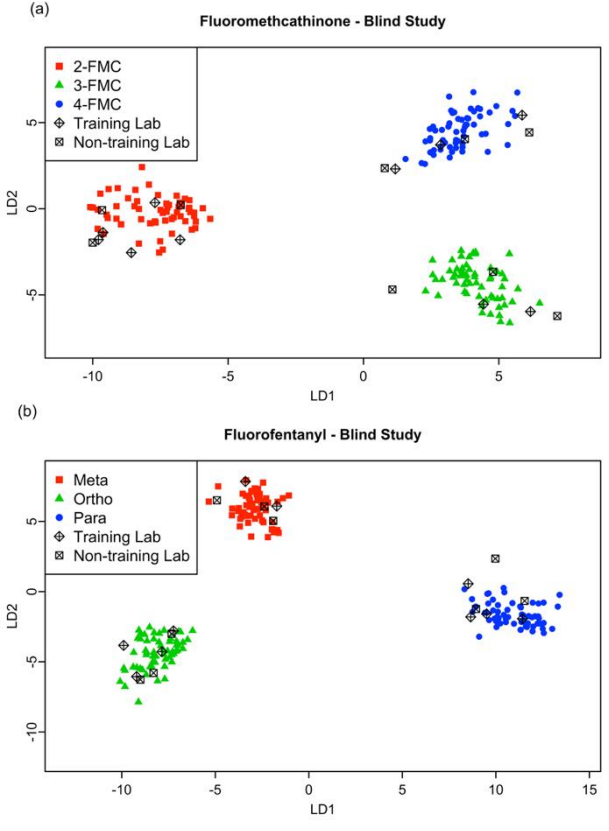
Linear regression using gradient descent





Machine Learning – Classification/Clustering

- Classification is a machine learning technique that is used to categorize data samples/events/items
- Clustering is a way to classify samples/events/items in an unsupervised way!



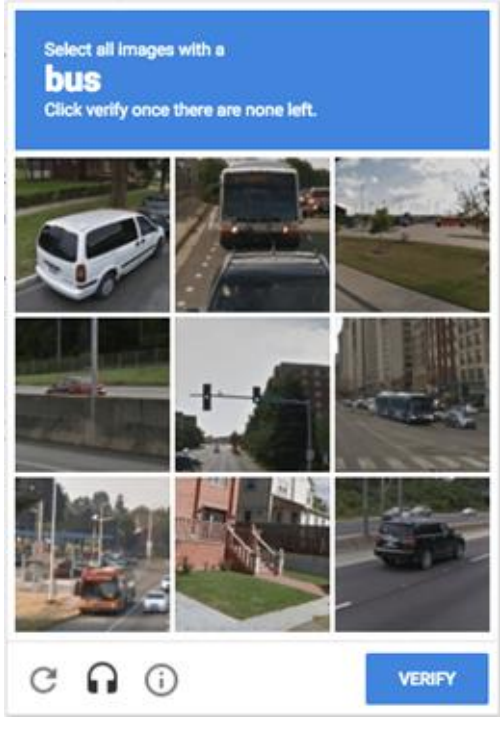
Neural Networks

- MNIST handwritten digit database
- (Modified National Institute of Standards and Technology dataset)
- 60,000 handwritten digits from the census bureau for training
- 10,000 test images
- Each image is 28x28 pixels and has a corresponding label
- Install tensorflow
  - pip install tensorflow

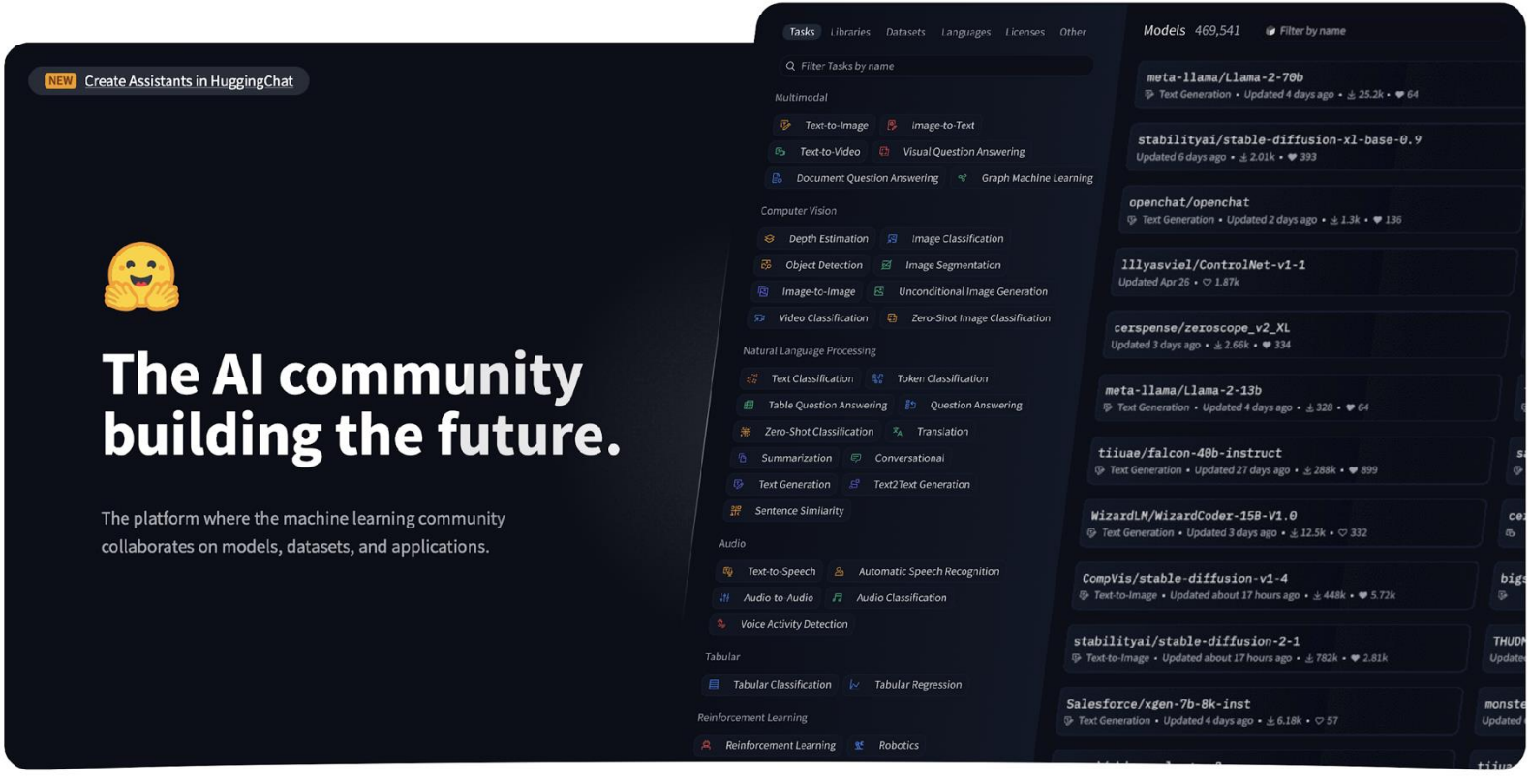
Autonomous Vehicles - Object Detection

You've been creating training datasets!

https://xkcd.com/1897/



Proliferation of Pretrained Models



YOLOv8

- YOLOv8 (You Only Look Once, version 8) is the latest iteration in the YOLO series of real-time object detection algorithms. It is designed to achieve high accuracy and speed in detecting multiple objects in images or video streams.

```python
from ultralytics import YOLO

# Load a model
model = YOLO("yolov8n.yaml")  # build a new model from scratch
model = YOLO("yolov8n.pt")  # load a pretrained model (recommended for training)

# Use the model
model.train(data="coco128.yaml", epochs=3)  # train the model
metrics = model.val()  # evaluate model performance on the validation set
results = model("https://ultralytics.com/images/bus.jpg")  # predict on an image
path = model.export(format="onnx")  # export the model to ONNX format
```

https://github.com/ultralytics/ultralytics

Computational advances

An Nvidia RTX 3090 is roughly equivalent to the world's fastest supercomputer in 2002

=

Machine Learning Summary

- **Machine learning**: field that explores the use of algorithms that can learn from the data and use that knowledge to make predictions on the data it has not seen before.
- **Supervised learning**: branch of machine learning that includes problems where a model could be built using the data and true labels or values.
- **Unsupervised learning**: branch of machine learning that includes problems where we do not have true labels for the data to train with. Instead, the goal is to somehow organize the data in some meaningful clusters or densities.

REGRESSION

Nonlinear Least Squares Regression

- **Estimate model parameters** by minimizing the sum of squared residuals between data and model predictions.
- Applicable when the relationship between variables is nonlinear.
  - **Function can be linear or nonlinear,** though the former is easily solvable using other methods
- Typically solved using methods like the Levenberg–Marquardt algorithm.
  - **Variant of gradient descent**
- Essential in fields from business (e.g., modeling diminishing returns) to biology (e.g., enzyme kinetics) and chemistry (e.g., reaction rate analysis).
- Implemented in Python via **scipy.optimize.curve_fit**.
- Provides flexible, accurate modeling of complex, real-world phenomena.
  - Provides a covariance matrix for uncertainty estimation!

Nuclear Physics

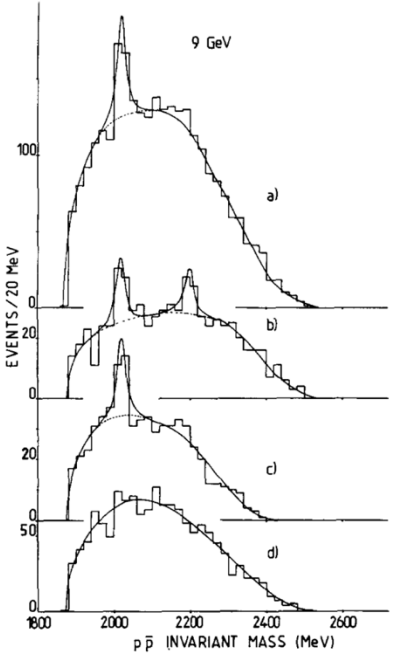
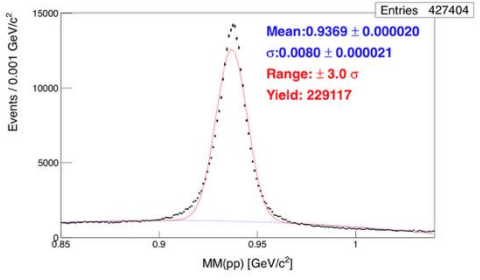
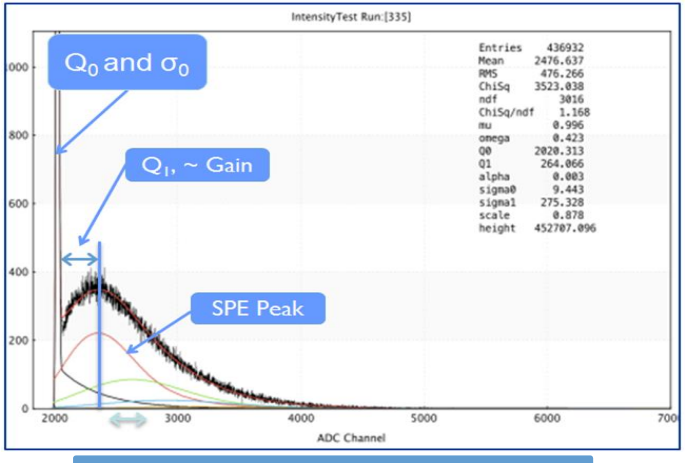


Figure 1.2: The invariant mass distributions for $p\bar{p}$ in the reaction $\pi^- p \rightarrow p_f \pi^- [p\bar{p}]$ using a 9 and 12 GeV π^- beam. a) Shows all events, b) shows events with an mass selection, selecting the $\Delta(1232)$ region in the $p_{fast}\pi^-$ invariant mass, c) shows events with an mass selection, selecting the $N(1520)$ region in the $p_{fast}\pi^-$ invariant mass, and d) showing the mass distribution without the $\Delta(1232)$ and $N(1520)$ mass selections in the $p_{fast}\pi^-$ invariant mass. [6]

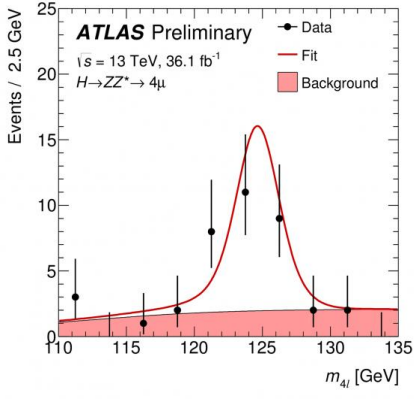
New Particle Characterization

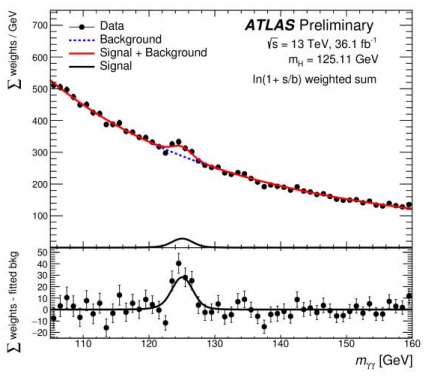


Yield Estimation w/Background



Detector Calibration





Higgs Boson

Nonlinear regression using gradient descent

- A process for modeling the relationship between variables of interest
- Gradient descent is a general approach in machine learning that could be used for solving linear regression

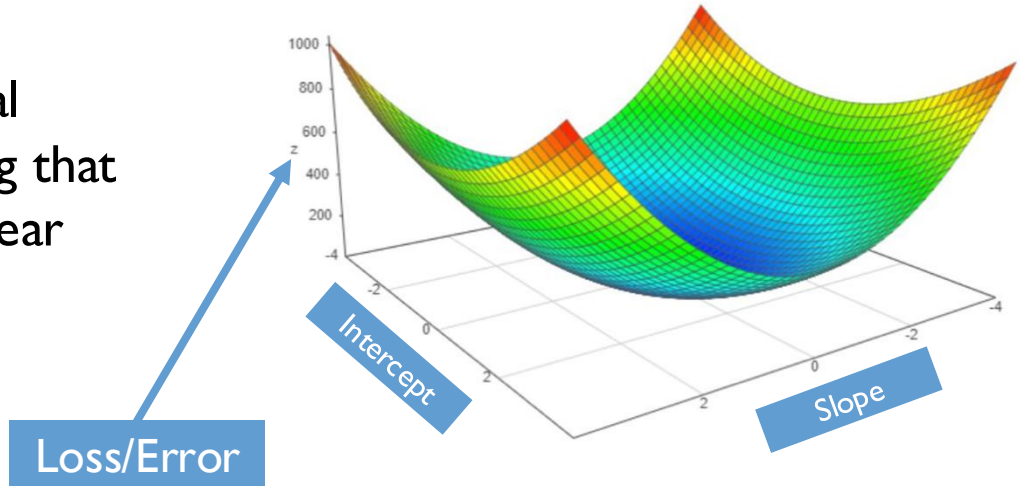
$$\chi_c^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

where:

$c =$ Degrees of freedom

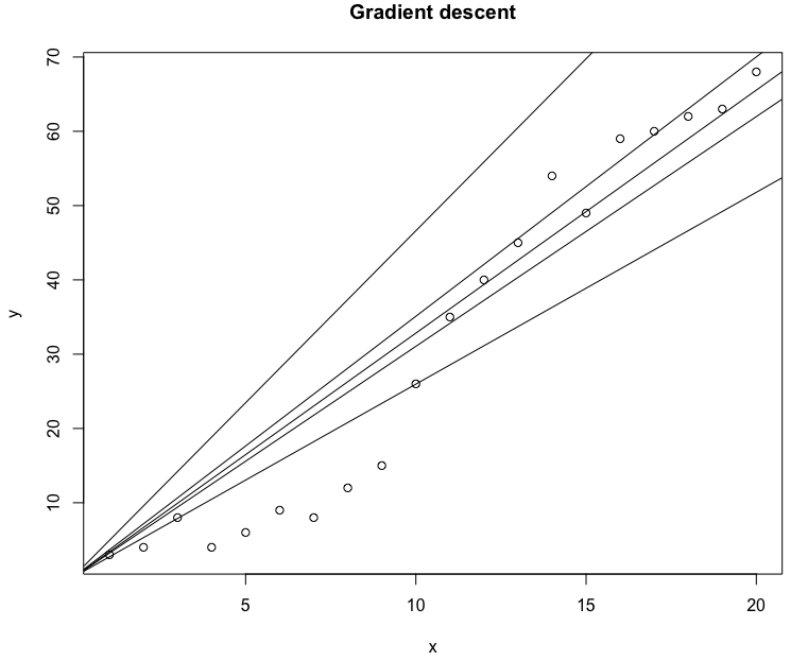
$O =$ Observed value(s)

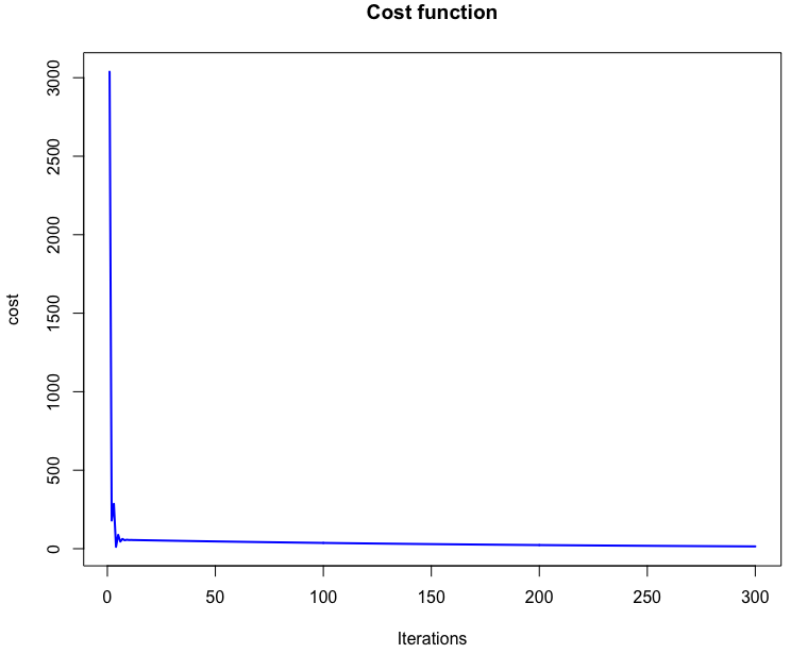
$E =$ Expected value(s)



Error surface for various lines created using linear regression (x represents slope, y represents intercept and z is the error value).

Nonlinear regression using gradient descent





scipy.optimize.

curve_fit

curve_fit(*f*, *xdata*, *ydata*, *p0*=None, *sigma*=None, *absolute_sigma*=False, *check_finite*=None, *bounds*=(-inf, inf), *method*=None, *jac*=None, *, *full_output*=False, *nan_policy*=None, **kwargs) [source]

Use non-linear least squares to fit a function, f, to data.

Assumes `ydata = f(xdata, *params) + eps`.

Parameters:

f : *callable*

The model function, f(x, ...). It must take the independent variable as the first argument and the parameters to fit as separate remaining arguments.

xdata : *array_like*

The independent variable where the data is measured. Should usually be an M-length sequence or an (k,M)-shaped array for functions with k predictors, and each element should be float convertible if it is an array like object.

ydata : *array_like*

The dependent data, a length M array - nominally `f(xdata, ...)`.

p0 : *array_like, optional*

Initial guess for the parameters (length N). If None, then the initial values will all be 1 (if the number of parameters for the function can be determined using introspection, otherwise a ValueError is raised).

How to use curve_fit

```
def gauss(x, amp, mean, sigma): # x is an array, others are scalar parameters
    return amp*np.exp(-(x-mean)**2 / (2*sigma)**2)
[9]
```

Define function as y=f(x, params)

Output parameters

function

Data: x, y, y_sigma

Initial Guess (Opt)

```
params , cov_matrix = curve_fit(gauss, bin_centers, y_values, sigma=y_error, p0=[1000000.0, 48.01,1.0])
[11]
```

Call curve_fit with function, x, y, y_error, and initial guess (optional)

In Class Exercise

OVER/UNDERFITTING

Overfitting and Underfitting

towardsdatascience.com

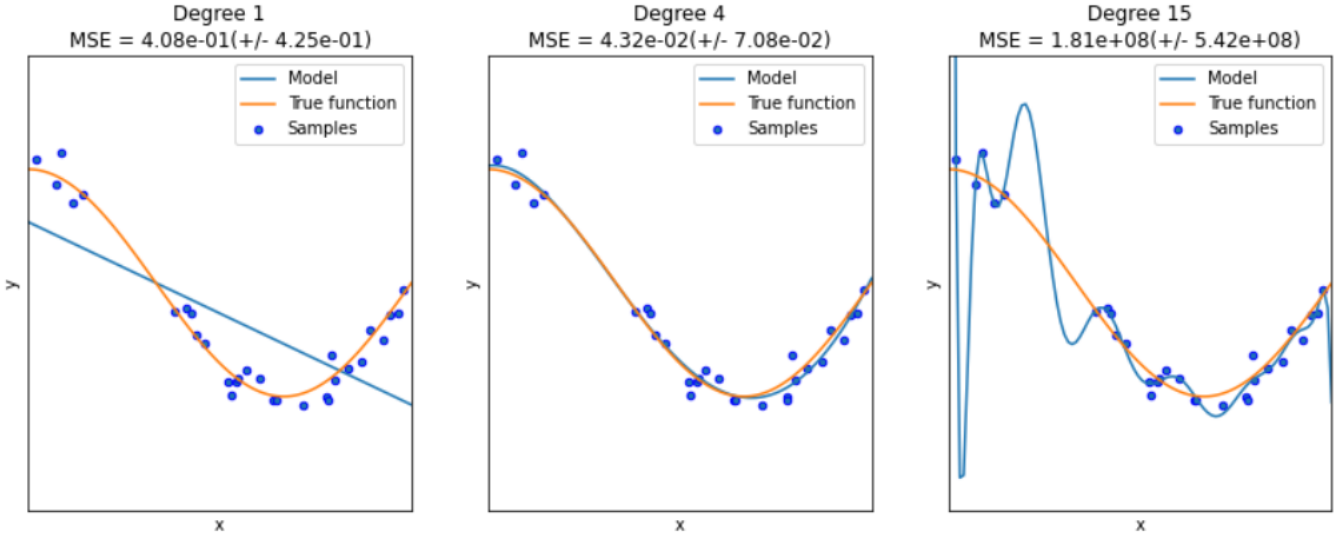
- **Underfitting** happens when the model has a very high bias and is unable to capture the complex patterns in the data. This leads to higher uncertainty since the model is not complex enough to fit the underlying data
- **Overfitting** is the opposite in the sense that the model is too complex (or higher model) and captures even the noise in the data. Therefore, in this case one would observe a very low test error value.

Overfitting and Underfitting

- Last week we were able to fit datasets that we generated using the same "model" or function that they were generated with
- When given a dataset in the real world you might not know what function you should use to fit the data
- You can also see overfitting and underfitting in other ML techniques other than regression in a 2D space (what we have been doing)

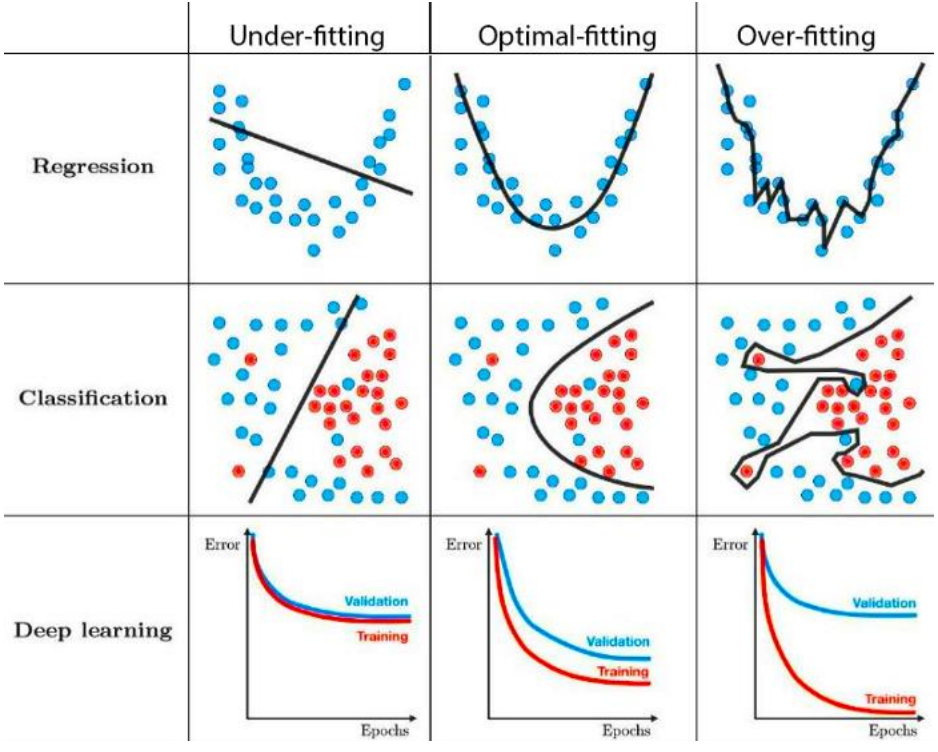
Overfitting and Underfitting

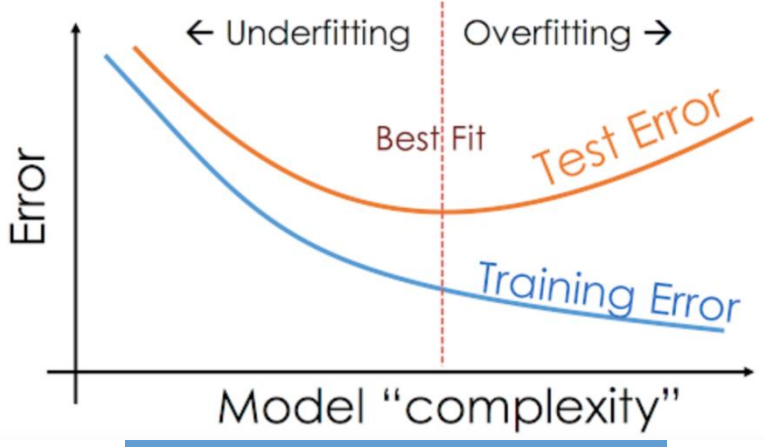
towardsdatascience.com



Overfitting and Underfitting

towardsdatascience.com

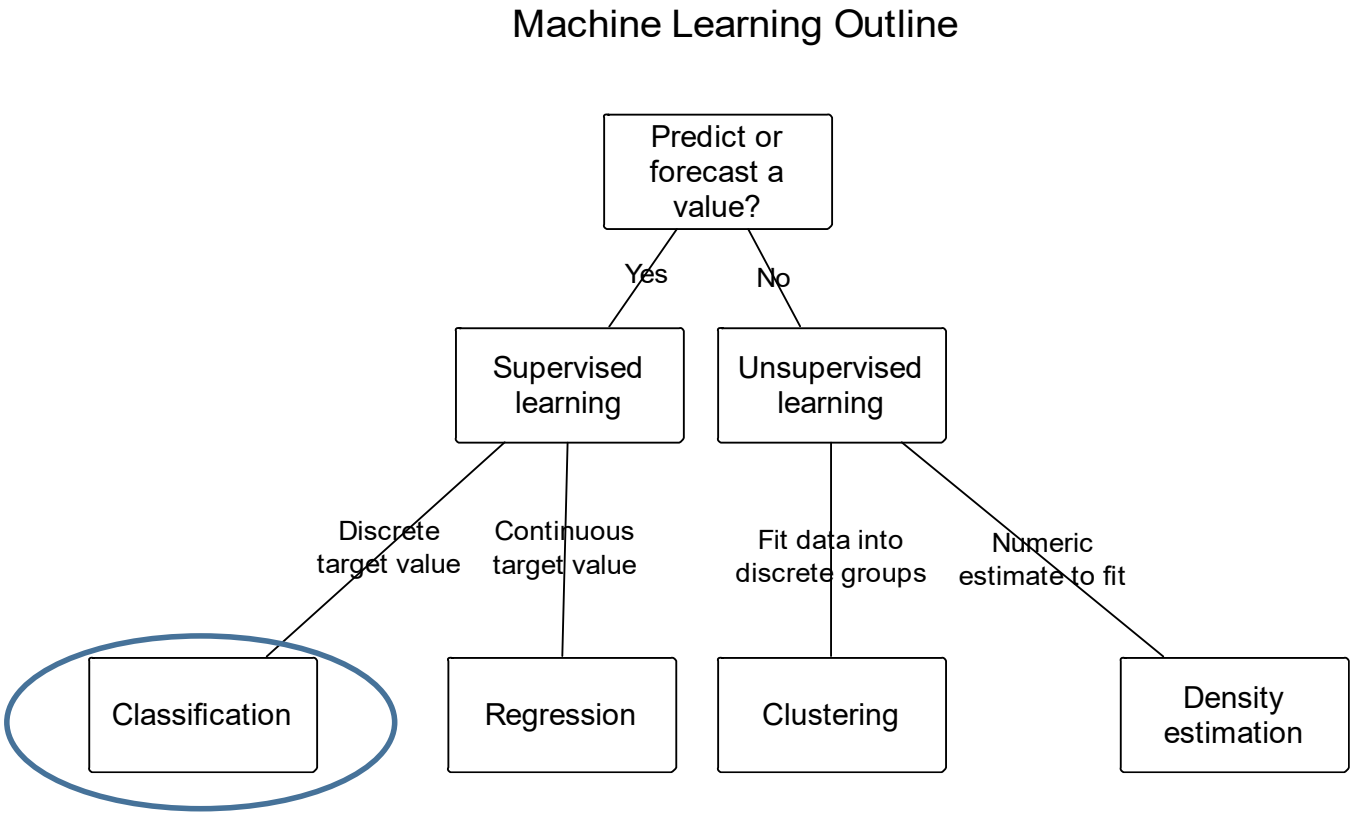




For Neural Networks

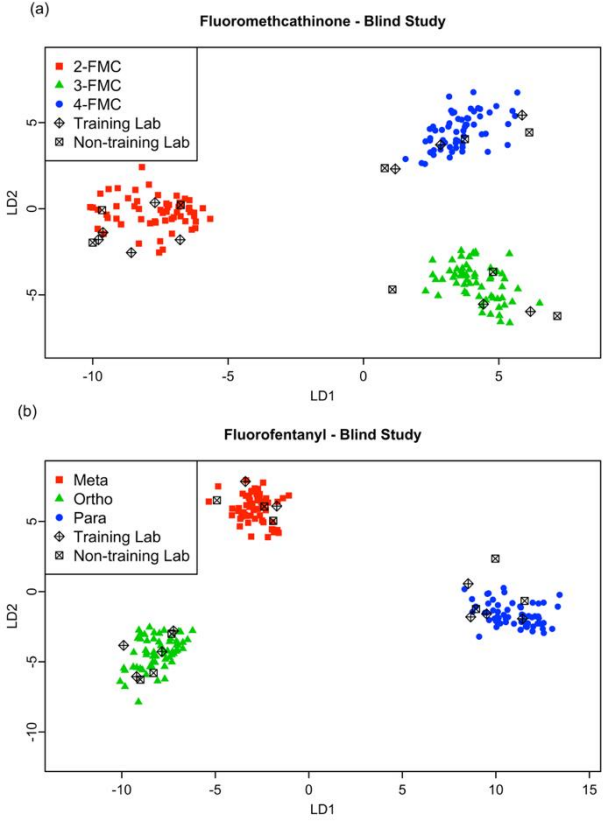
CLASSIFICATION

Outline of core ML problems



Machine Learning – Classification/Clustering

- Classification is a machine learning technique that is used to categorize data samples/events/items
- Clustering is a way to classify samples/events/items in an unsupervised way!



Classification

- Classification is can be supervised or unsupervised
- In supervised classification the training data is labeled and the number of classes is fixed (cat vs dog)
- When there are only two classes involved it is called binomial classification
- There are many methods but k Nearest Neighbor (kNN) is the one of the most popular and a great way to introduce the material

kNN Algorithm

- First you need a dataset with labels
- When given a new data point:
  - Calculate the distance to all points (Usually Euclidean distance)
  - Get the k nearest neighbors
  - Take the majority vote to determine classification

scikit-learn

Machine Learning in Python

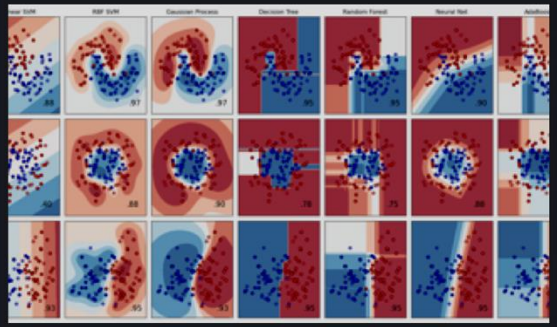
Getting Started | Release Highlights for 1.6

- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.
Algorithms: Gradient boosting, nearest neighbors, random forest, logistic regression, and more...

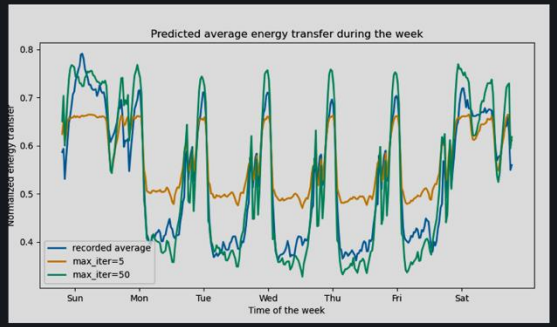


Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, stock prices.
Algorithms: Gradient boosting, nearest neighbors, random forest, ridge, and more...

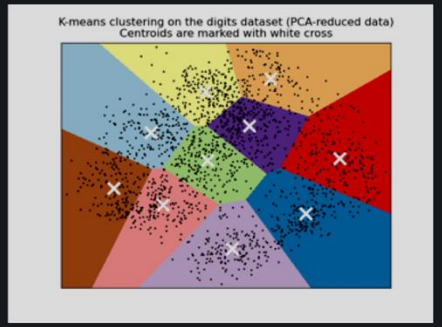


Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, grouping experiment outcomes.
Algorithms: k-Means, HDBSCAN, hierarchical clustering, and more...



Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Model selection

Comparing, validating and choosing parameters and models.

Preprocessing

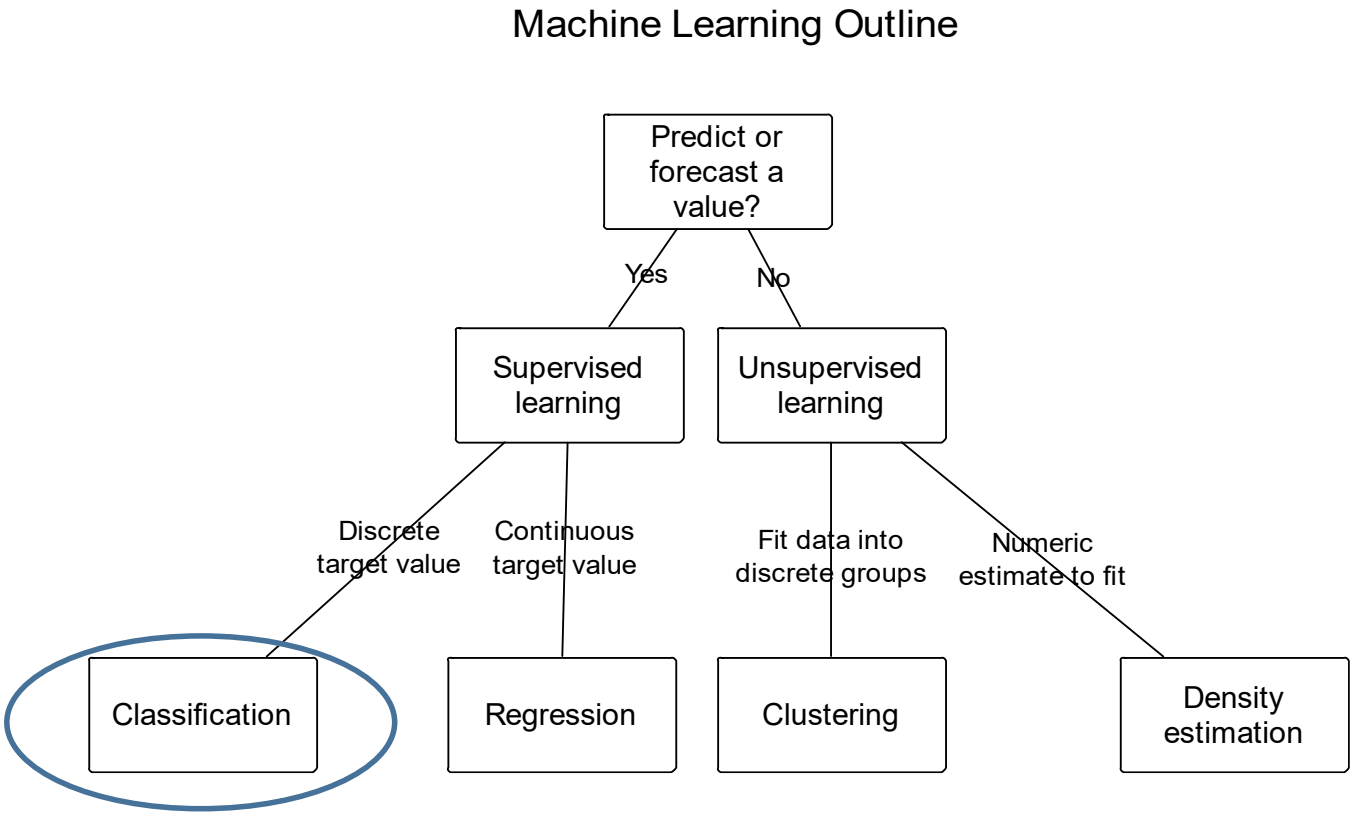
Feature extraction and normalization.

Applications: Transforming input data such as text

In-class Exercise

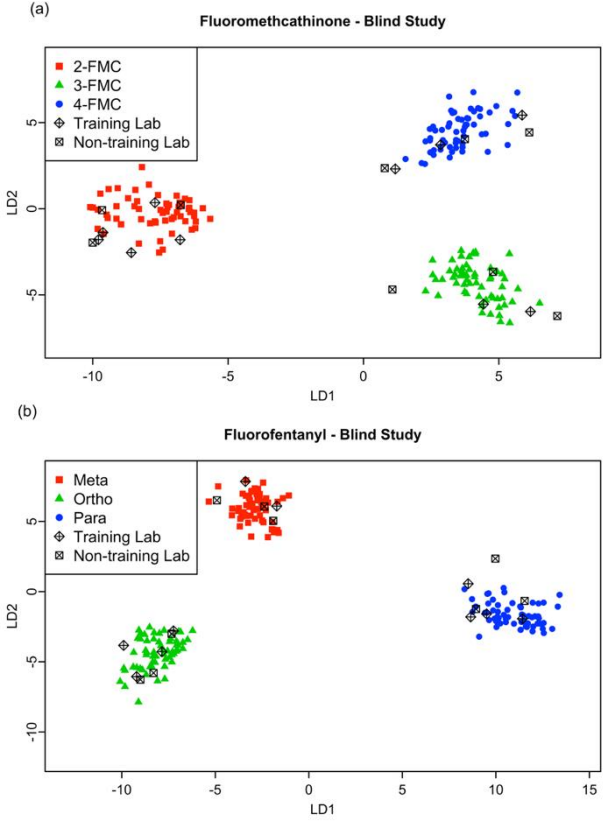
CLASSIFICATION

Outline of core ML problems



Machine Learning – Classification/Clustering

- Classification is a machine learning technique that is used to categorize data samples/events/items
- Clustering is a way to classify samples/events/items in an unsupervised way!



Classification

- Classification is can be supervised or unsupervised
- In supervised classification the training data is labeled and the number of classes is fixed (cat vs dog)
- When there are only two classes involved it is called binomial classification
- There are many methods but k Nearest Neighbor (kNN) is the one of the most popular and a great way to introduce the material

kNN Algorithm

- First you need a dataset with labels
- When given a new data point:
  - Calculate the distance to all points (Usually Euclidean distance)
  - Get the k nearest neighbors
  - Take the majority vote to determine classification

kNN Algorithm

- What is the class at k=3?
- What is the class at k=5?

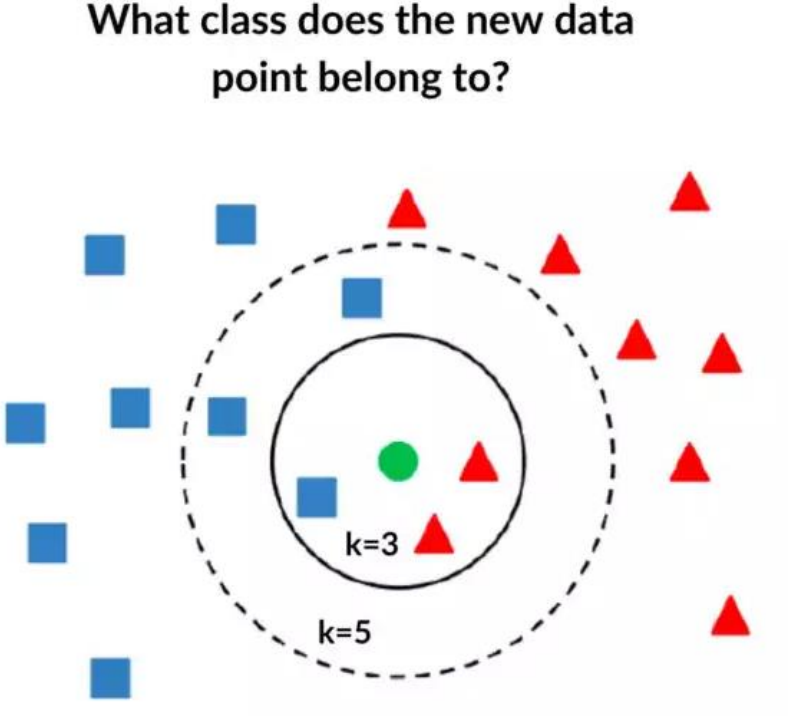


Figure source: spotintelligence.com

scikit-learn

Machine Learning in Python

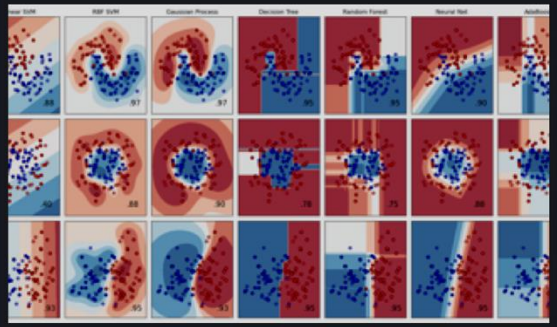
Getting Started | Release Highlights for 1.6

- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.
Algorithms: Gradient boosting, nearest neighbors, random forest, logistic regression, and more...

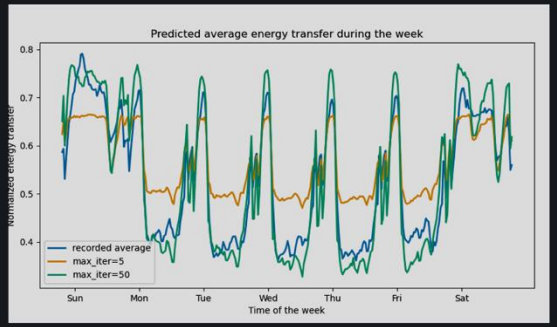


Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, stock prices.
Algorithms: Gradient boosting, nearest neighbors, random forest, ridge, and more...

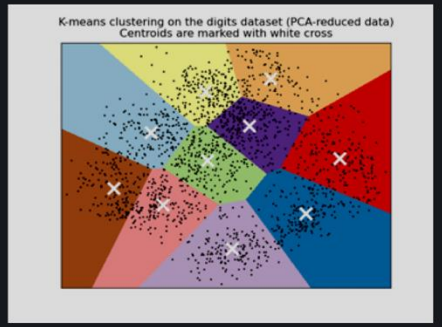


Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, grouping experiment outcomes.
Algorithms: k-Means, HDBSCAN, hierarchical clustering, and more...



Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Model selection

Comparing, validating and choosing parameters and models.

Preprocessing

Feature extraction and normalization.

Applications: Transforming input data such as text

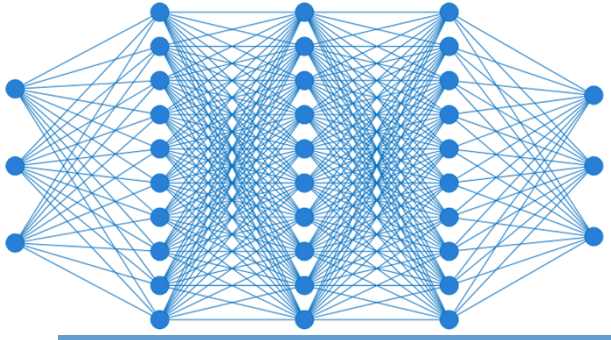
In-class Exercise

NEURAL NETWORKS

Neural Networks

- We have discussed machine learning techniques such as regression using nonlinear least squares methods and classification using k Nearest Neighbors
- What if we could perform classification or regression like the human brain does?
- We look at images and can easily tell whether it is a cat or a dog!
- With neural networks in the next few lectures we will mainly focus on supervised learning.
  - Labeled data (ex: image has label: cat or dog)

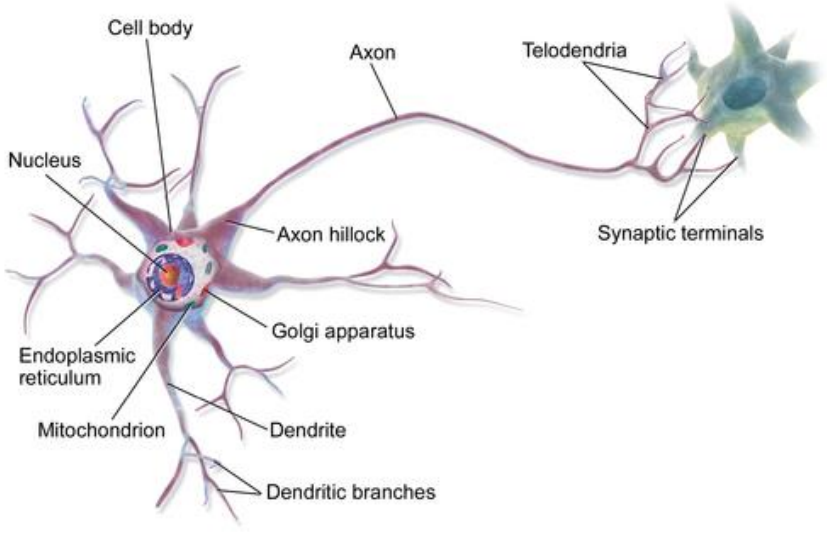
Cat or Doge?



Example Neural Network

Early History

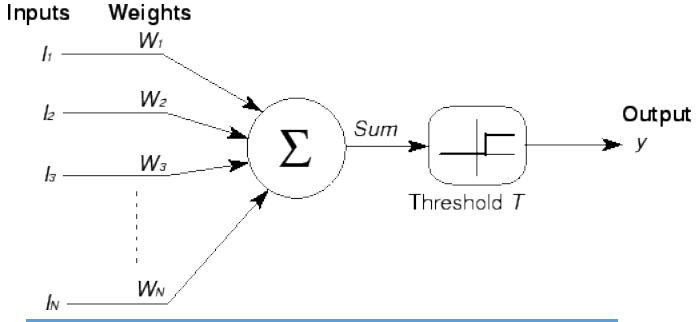
- Artificial Neural Networks were proposed in 1943
  - Warren McCulloch a neurophysiologist
  - Walter Pitts, mathematician
- Donald Hebb furthered ANNs
  - 'Hebbian Learning' — a model of learning based on neural plasticity, proposed by Donald Hebb in his book "The Organization of Behaviour" often summarized by the phrase: "Cells that fire together, wire together."



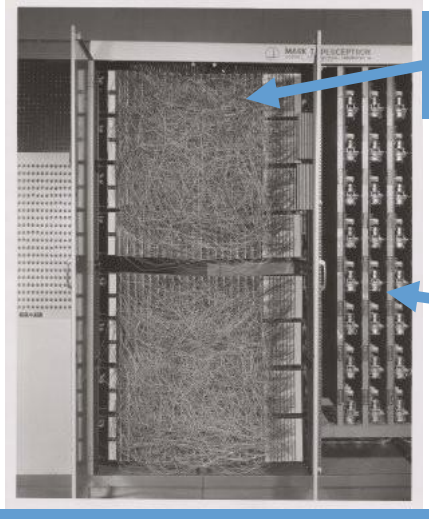
- The human brain has approximately 86 billion neurons
  - ~100 trillion connections
  - ~300k mi of connections
  - Axons can be very short or up to a meter in length

Early History

- Frank Rosenblatt created the first machine called the Mark I Perceptron in 1958
- It was based on the McCulloch-Pitts neuron
  - Warren McCulloch
  - Walter Pitts
- The Mark I Perceptron was connected to a 20x20 pixel "camera" made of cadmium sulfide photocells



McCulloch-Pitts Neuron



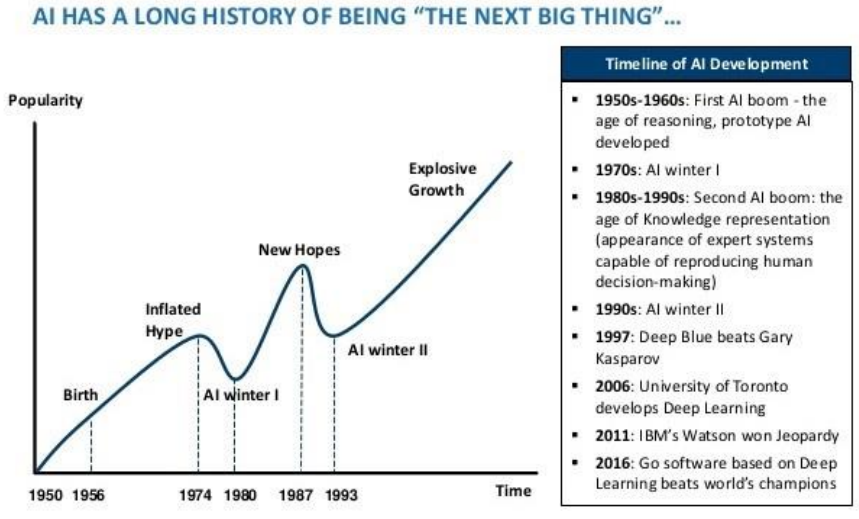
Patch Panel for input variations

Potentiometers

Mark I Perception - 1958

Early History and Hype

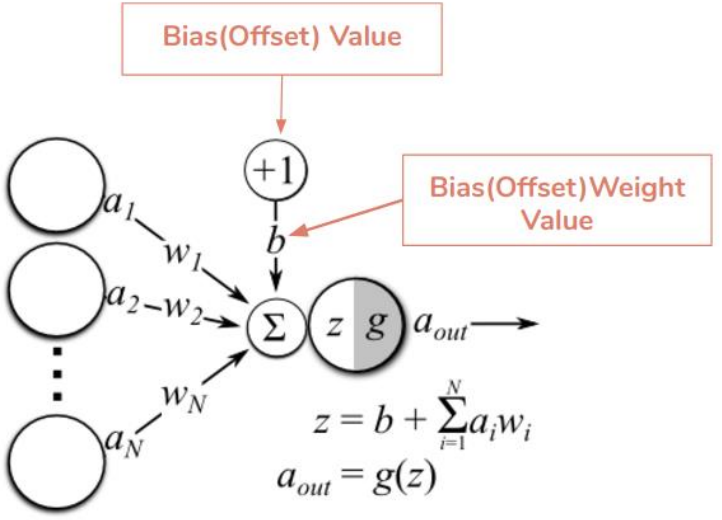
- Research continued into the 60's and was receiving much attention and funding
  - Machines were only able to separate linearly separable data
- In 1969, "Perceptrons" was published by Marvin Minsky, founder of MIT AI Lab and Seymour Papert the director of the AI Lab
  - Caused much doubt in MLPs and contributed to AI Winter



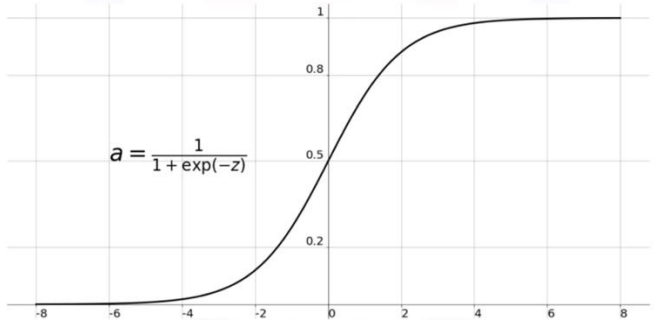
https://www.actuaries.digital/2018/09/05/history-of-ai-winters/

Neuron and Activation Functions

- Neural Networks are composed of individual Neurons
- Neurons have inputs and those inputs have trainable parameters called weights
- The weights are applied to the respective inputs and summed together
- After summing the inputs/weights and bias the output is determined by the activation function



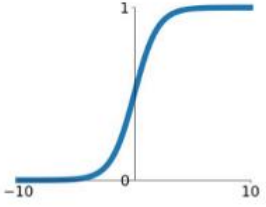
Sigmoid Function



Activation Functions

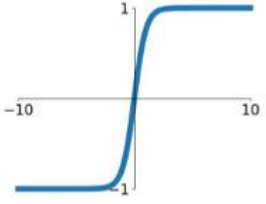
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



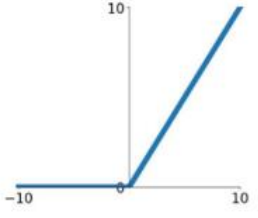
tanh

$$\tanh(x)$$



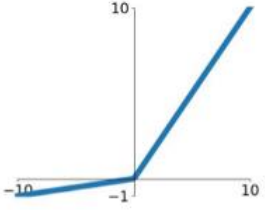
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

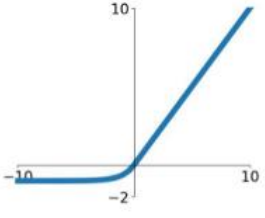


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

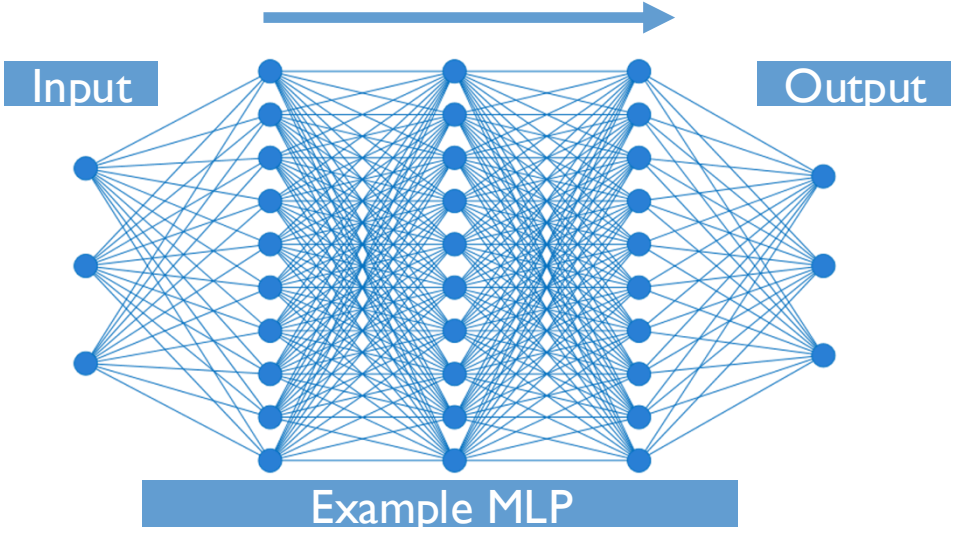
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

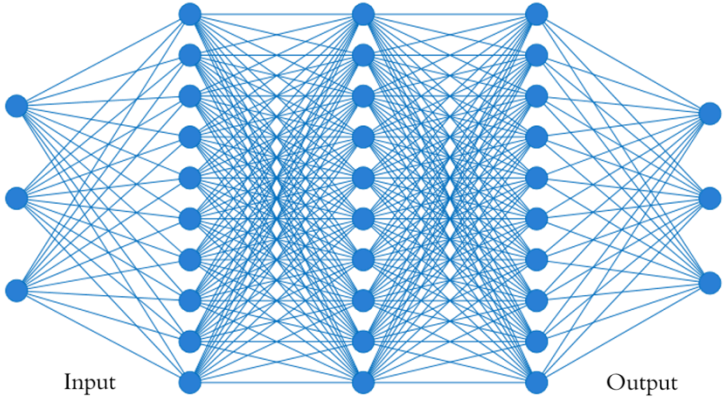


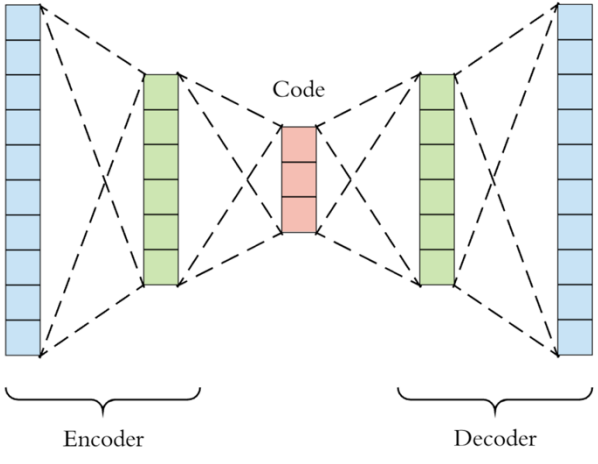
Multilayered perceptron MLP

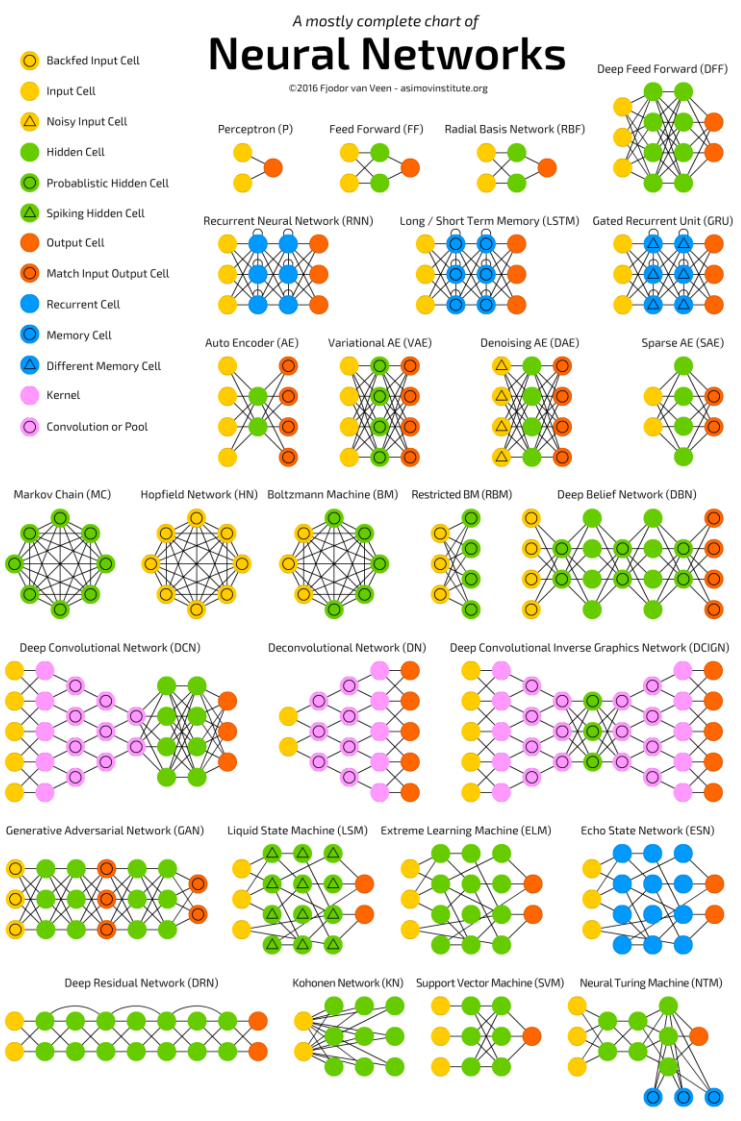
- The arrangement of neurons within a neural network is called a network architecture
- In the 1980's creating layers of perceptrons became a popular network architecture
  - It is still widely used today
- This architecture is powerful but posed significant challenges when it came to setting weights and biases
  - Future: Backpropagation/Gradient Descent



Neural Network Architectures

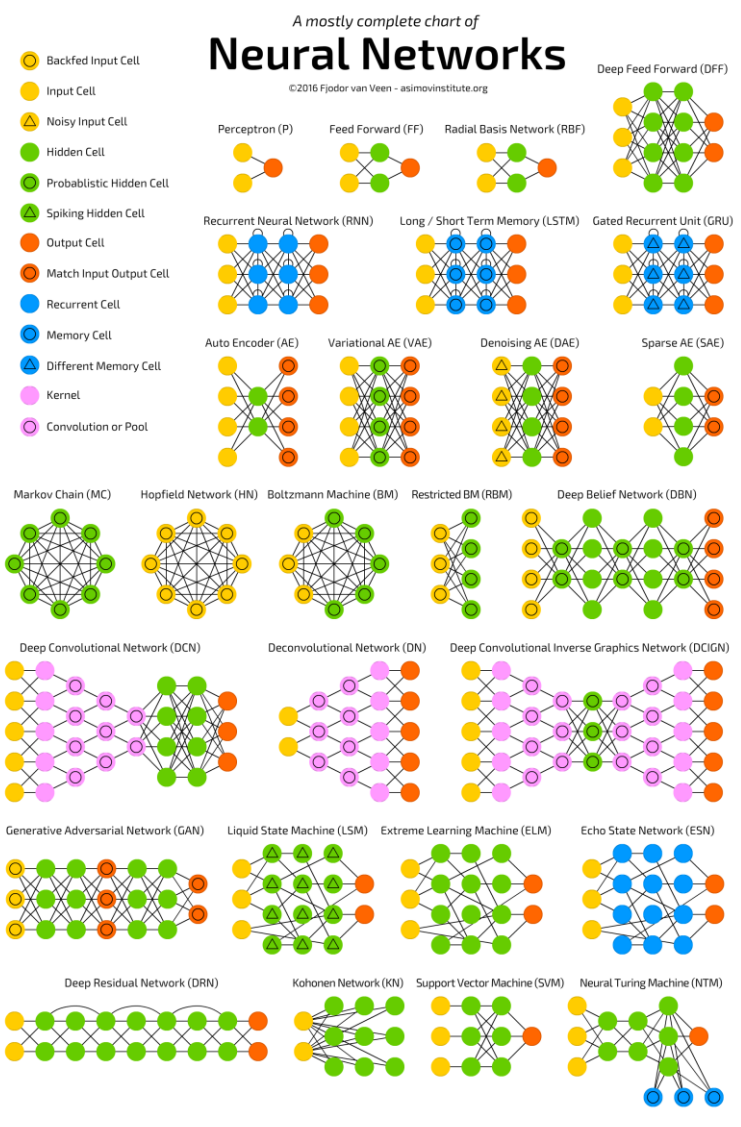






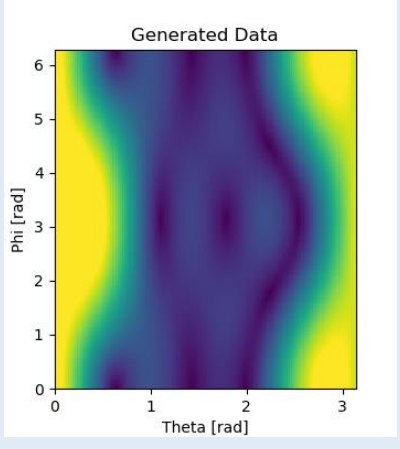
Neural Network Architectures

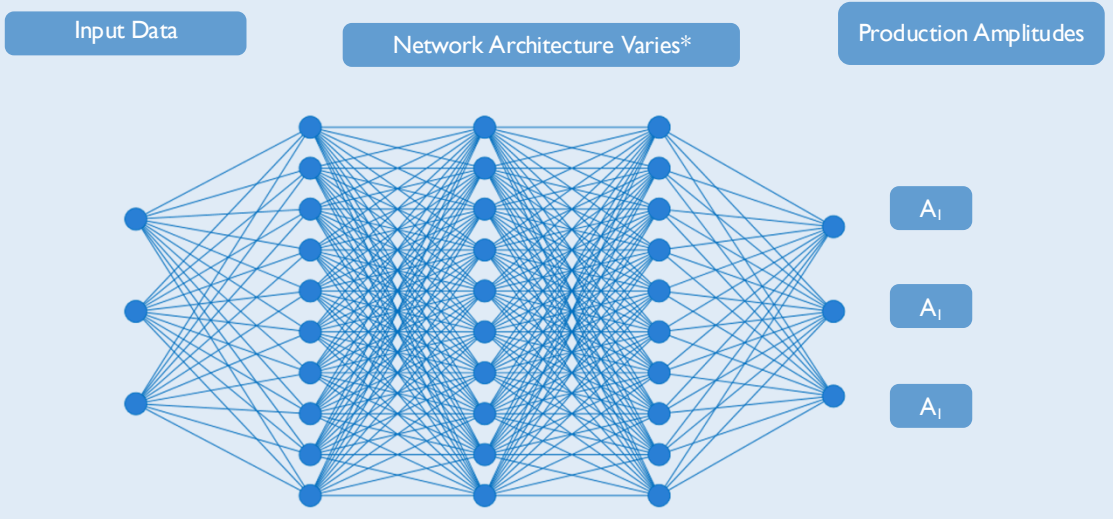
- MLPs – General all around neural network, what everyone thinks of when you mention NNs
- Recurrent Neural Networks/LSTMs – Used for time series data
- **Convolutional NNs – Good for images or other data where the relative position of features is important**
- Autoencoders – Unsupervised learning, good for denoising
- Generative models – GANs, VAEs, etc. Good for generating new data based on prior knowledge
- …



MLP Example – PWA

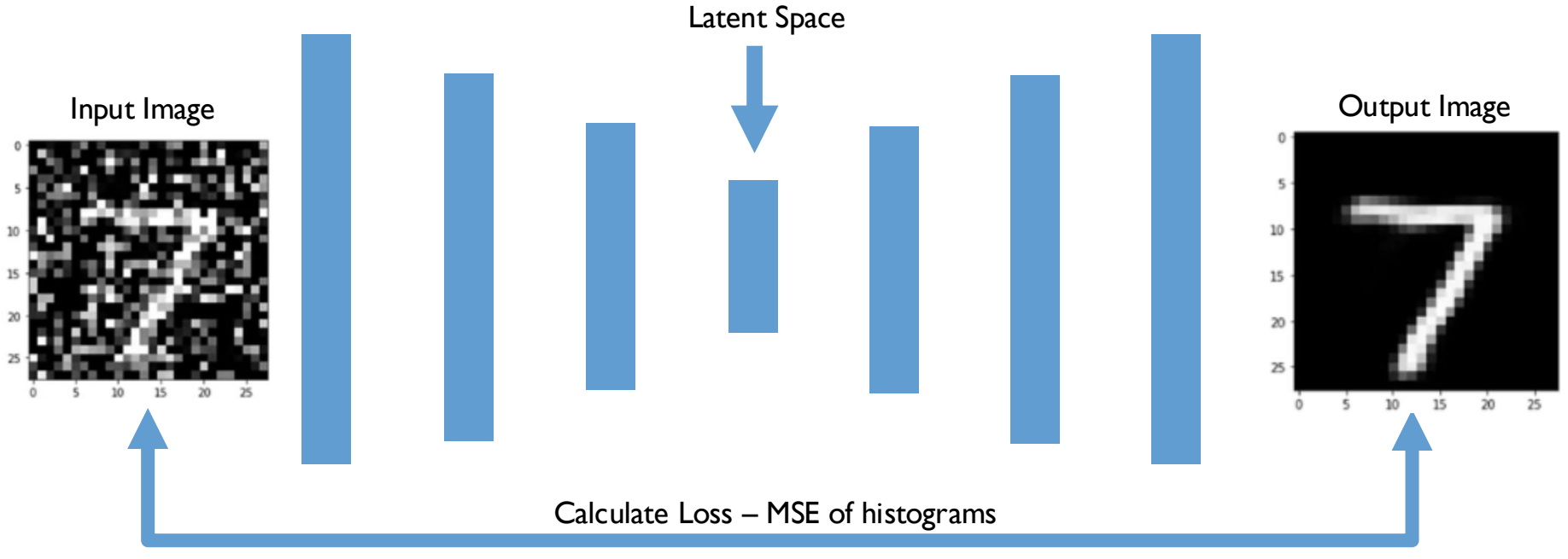
- Read in 100x100 pixel hits from CSV files, provide production amplitudes.
- Model Architecture: MLP w/3 layers of 2000 neurons, relu activation function





Autoencoder Example – Hackathon

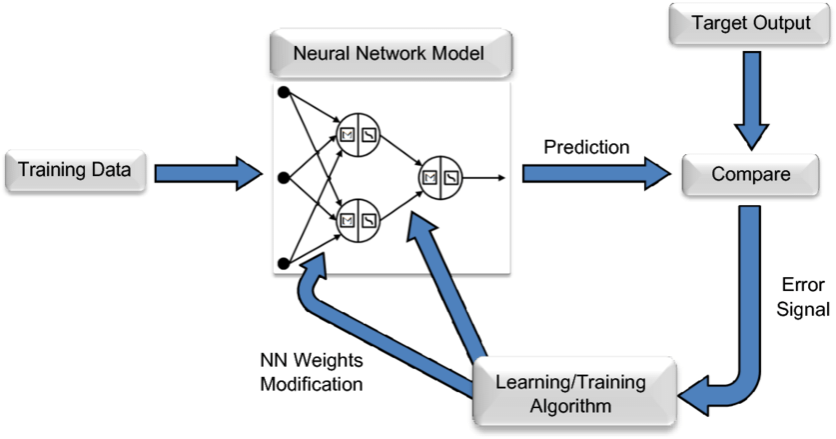
Unsupervised learning!



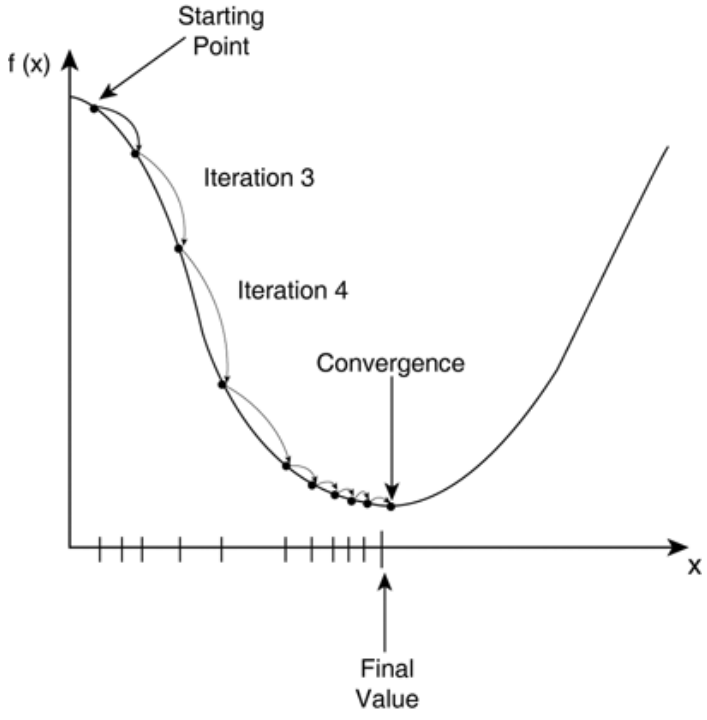
2000→1000→100→3 …

Backpropagation and Gradient Descent

- Up until 1986 backpropagation was not used in neural networks
- Without backpropagation you cannot calculate the gradient for each weight
- Loss function defined to compare prediction w/target output



Training a Neural Network



Gradient Descent

Tools of the Trade

- Long gone are the days where you would write your own NN Library!!!
- Today there are two main packages competing for dominance: PyTorch (Facebook/Meta) and Tensorflow/Keras (Google)
- Python 3.9-3.12 – Anaconda (except at JLab)
  - Keras/TensorFlow - NN Libraries
  - Pandas/Numpy - Data Handling
  - Matplotlib - Visualization
- Train on your own machine or on google colab

Sample Training Script

```
test = pd.read_csv("TRAIN/TRAIN.csv")
labels = pd.read_csv("TRAIN/TRAIN_labels.csv")
activation = 'relu'

model = Sequential()
model.add(Dense(units=1000, activation=activation, input_shape=(3600, )))
model.add(Dense(units=1000, activation=activation))
model.add(Dense(units=1000, activation=activation))
model.add(Dense(units=2))
model.compile(optimizer=adam(lr=.001), loss='mean_squared_error', metrics=['accuracy'])

model.fit(test, labels[labels.columns[1:]], epochs=300, batch_size=256, validation_split=0.2)
```

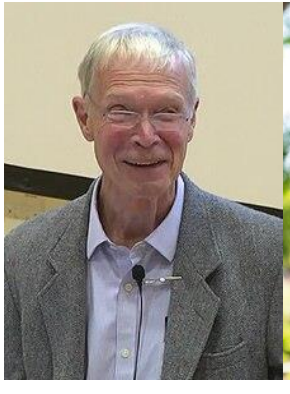
Deep Learning

- Fathers of Deep Learning: Yann LeCun, Geoffrey Hinton, and Yoshua Bengio
- With increasing computation resources and backpropagation deep learning became a possibility in the 90's
- First trained on GPUs in 2009/2010 allowing for even larger models!

2018 ACM A.M. Turing Award

2024 Nobel Prize in Physics

- Announced **10/8/24**
- "for foundational discoveries and inventions that enable machine learning with artificial neural networks"
- John Hopfield and Geoffrey Hinton
- 11m Kroner -> $1m USD



PRESS RE

8 October 2024

The Nobel Prize in Physics 2024

The Royal Swedish Academy of Sciences has decided to award the Nobel Prize in Physics 2024 to

John J. Hopfield
Princeton University, NJ, USA

Geoffrey E. Hinton
University of Toronto, Canada

"for foundational discoveries and inventions that enable machine learning with artificial neural networks"

They trained artificial neural networks using physics

This year's two Nobel Laureates in Physics have used tools from physics to develop methods that are the foundation of today's powerful machine learning. John Hopfield created an associative memory that can store and reconstruct images and other types of patterns in data. Geoffrey Hinton invented a method that can autonomously find properties in data, and so perform tasks such as identifying specific elements in pictures.

When we talk about artificial intelligence, we often mean machine learning using artificial neural networks. This technology was originally inspired by the structure of the brain. In an artificial neural network, the brain's neurons are represented by nodes that have different values. These nodes influence each other through connections that can be likened to synapses and which can be made stronger or weaker. The network is *trained*, for example by developing stronger connections between nodes with simultaneously high values. This year's laureates have conducted important work with artificial neural networks from the 1980s onward.

John Hopfield invented a network that uses a method for saving and recreating patterns. We can imagine the nodes as pixels. The *Hopfield network* utilises physics that describes a material's characteristics due to its atomic spin – a property that makes each atom a tiny magnet. The network as a whole is described in a manner equivalent to the energy in the spin system found in physics, and is trained by finding values for the connections between the nodes so that the saved images have low energy. When the Hopfield network is fed a distorted or incomplete image, it methodically works through the nodes and updates their values so the network's energy falls. The network thus works stepwise to find the saved image that is most like the imperfect one it was fed with.

Geoffrey Hinton used the Hopfield network as the foundation for a new network that uses a different method: the *Boltzmann machine*. This can learn to recognise characteristic elements in a given type of data. Hinton used tools from statistical physics, the science of systems built from many similar components. The machine is trained by feeding it examples that are very likely to arise when the machine is run. The Boltzmann machine can be used to classify images or create new examples of the type of pattern on which it was trained. Hinton has built upon this work, helping initiate the current explosive development of machine learning.

"The laureates' work has already been of the greatest benefit. In physics we use artificial neural networks in a vast range of areas, such as developing new materials with specific properties," says Ellen Moons, Chair of the Nobel Committee for Physics.

John J. Hopfield, born 1933 in Chicago, IL, USA. PhD 1958 from Cornell University, Ithaca, NY, USA. Professor at Princeton University, NJ, USA.

Geoffrey E. Hinton, born 1947 in London, UK. PhD 1978 from The University of Edinburgh, UK. Professor at University of Toronto, Canada.

Prize amount: 11 million Swedish kronor, to be shared equally between the laureates.
Further information: www.kva.se and www.nobelprize.org
Press contact: Eva Nevelius, Press Secretary, +46 70 878 67 63, eva.nevelius@kva.se
Experts: Olle Eriksson, +46 18 471 34 25, olle.eriksson@physics.uu.se and Anders Irbäck, +46 46 222 34 93, anders.irback@cec.lu.se, members of the Nobel Committee for Physics.

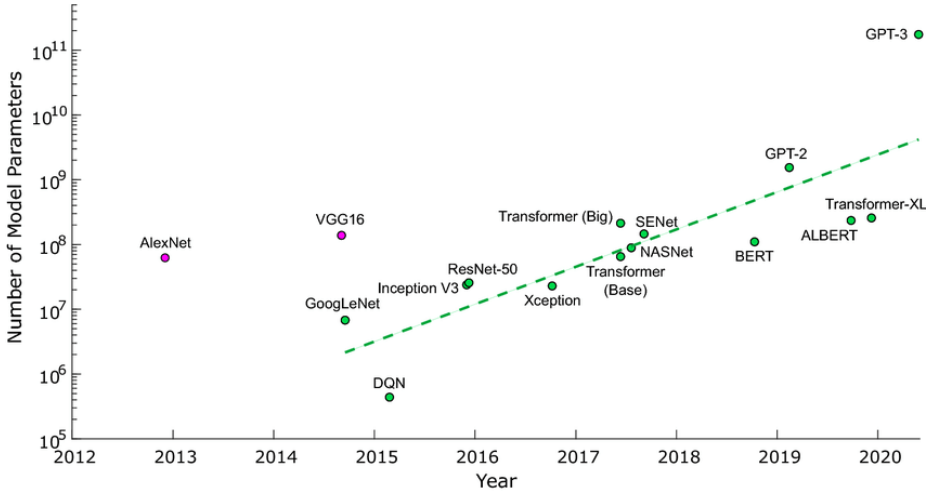
The Royal Swedish Academy of Sciences, founded in 1739, is an independent organisation whose overall objective is to promote the sciences and strengthen their influence in society. The Academy takes special responsibility for the natural sciences and mathematics, but endeavours to promote the exchange of ideas between various disciplines.

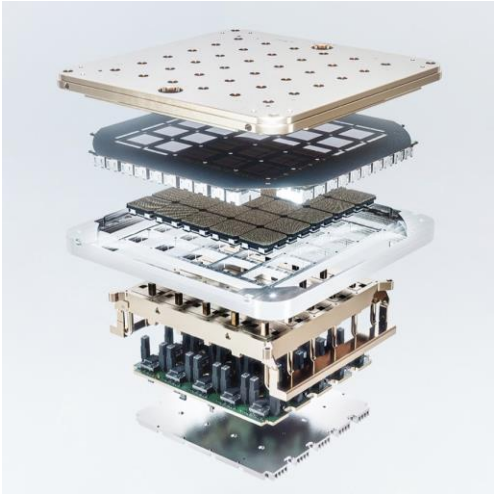
BOX 50005, SE-104 05 STOCKHOLM, SWEDEN
TEL +46 8 673 95 00 • WWW.KVA.SE
BESÖK/VISIT: LILLA FRESCATIVÄGEN 4A, SE-114 18 STOCKHOLM, SWEDEN

KUNGL. VETENSKAPS-AKADEMIEN
THE ROYAL SWEDISH ACADEMY OF SCIENCES

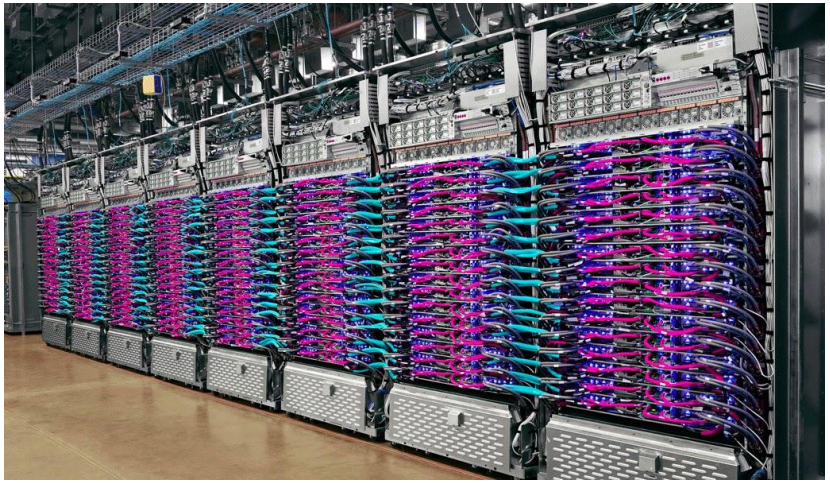
Computing Resources

- First trained on GPU in 2009
- Models are getting larger and larger and require specialized hardware to train





Tesla D1 Dojo Chip



Google TPU

Computational advances

An Nvidia RTX 3090 is roughly equivalent to the world's fastest supercomputer in 2002

=

Nvidia RTX 3090
35.58 Tflops of FP32
500 Watts
Released in 2020 ~$2k

NEC, Earth-simulator
5120 1GHz Cores
3.2MW, 35 TFlops

Your homework

- MNIST handwritten digit database
- (Modified National Institute of Standards and Technology dataset)
- 60,000 handwritten digits from the census bureau for training
- 10,000 test images
- Each image is 28x28 pixels and has a corresponding label

History of MNIST

MNIST From 1993 with Yann LeCunn

Autonomous Vehicles - Object Detection

You've been creating training datasets!

https://xkcd.com/1897/