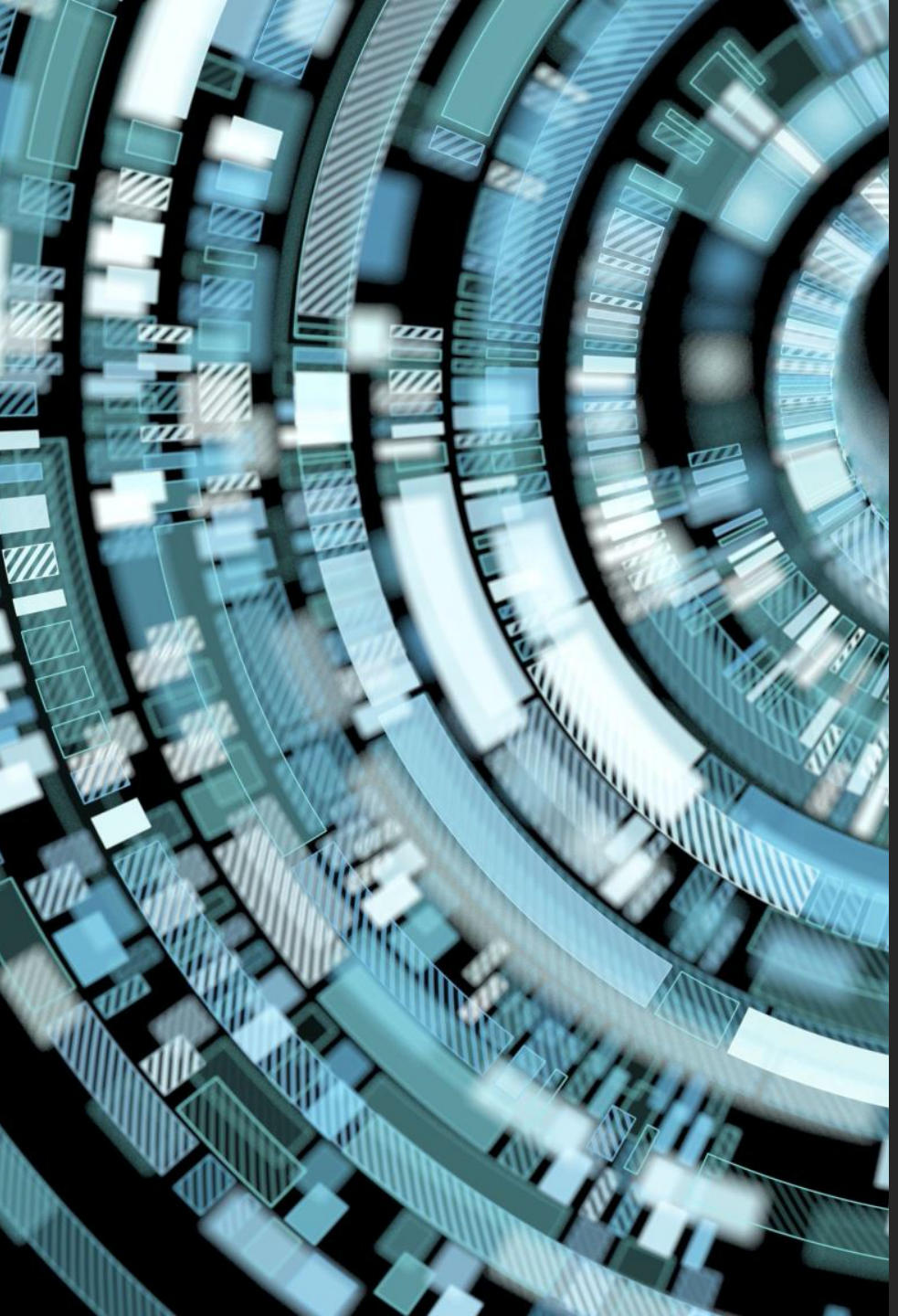


Machine Learning & Data Science II

Yingjie(Chelsea) Wang

Data Scientist at Food and Drug Administration (FDA)

Email: yw592@georgetown.edu

Course Formats

1.5-hour mentor session per week

1 hour

Go deeper into concepts from lecture

Python practices for ML tasks

0.5 hour

Discussion on topics

Help with presentation and homework

Type of Learning

- Supervised Learning:
  - each observation of predictor x_i, i=1, 2,…,n, has associated response y_i
  - Linear regression, Naive Bayes, SVMs, Logistic regression, Decision trees, Classification
- Unsupervised Learning:
  - each observation of predictor x_i, i=1, 2,…,n, doesn't have associated response y_i
  - K-means clustering, Hierarchical clustering, PCA, association rule mining

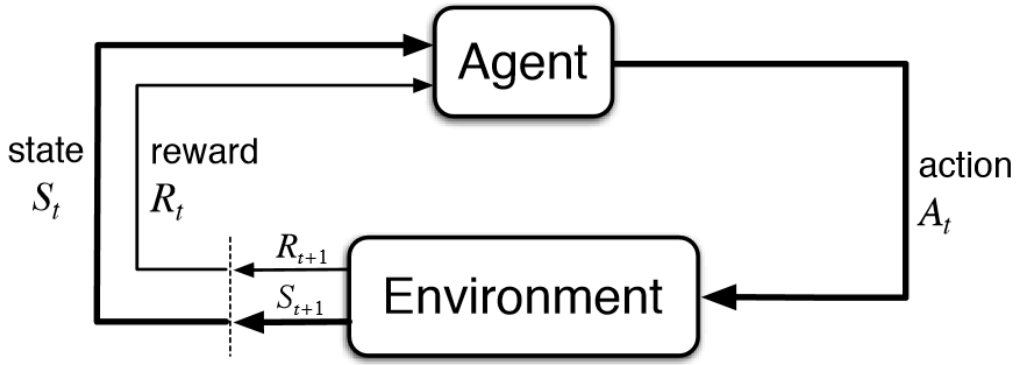
Parameters	**Supervised machine learning**	**Unsupervised machine learning**
Input Data	Algorithms are trained using **labeled** data	Algorithms are used against data which is not **labelled**
Computational Complexity	Simpler method	Computationally complex
Accuracy	Highly accurate and trustworthy method	Less accurate and trustworthy method

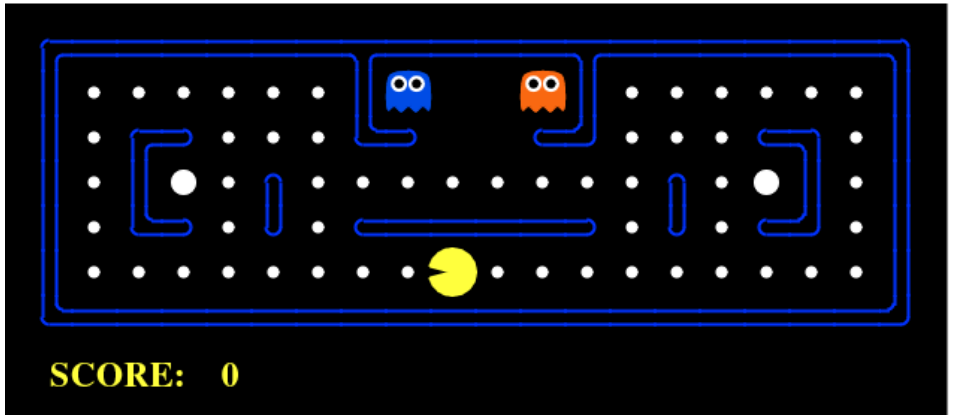
Type of Learning

- The acquisition of unlabeled data is relatively cheap while **labeling data is very expensive**.
- Semi-Supervised Learning
  - The algorithm is trained upon a combination of labeled and unlabeled data. Typically, this combination will contain a very small amount of labeled data and a very large amount of unlabeled data.
  - Basic procedure: cluster similar data using an unsupervised learning algorithm, and then use the existing labeled data to label the rest of the unlabeled data.
- Practical applications of Semi-Supervised Learning
  - Speech Analysis: Since labeling of audio files is a very intensive task, semi-supervised learning is a very natural approach to solve this problem.
  - Internet Content Classification: Labeling each webpage is an impractical and unfeasible process and thus uses Semi-Supervised learning algorithms. Even the Google search algorithm uses a variant of Semi-Supervised learning to rank the relevance of a webpage for a given query.
  - Protein Sequence Classification: Since DNA strands are typically very large in size, the rise of Semi-Supervised learning has been imminent in this field.

Type of Learning

- Reinforcement Learning
  - A type of machine learning technique that enables an **agent** to learn in an interactive **environment** by trial and error using **feedback** from its own **actions** and experiences. The objective of RL is to find a suitable action model that would **maximize the total cumulative reward** of the agent.
- Basic elements of an RL
  - Environment — Physical world in which the agent operates
  - State — Current situation of the agent
  - Reward — Feedback from the environment
  - Policy — Method to map agent's state to actions
  - Value — Future reward that an agent would receive by taking an action in a particular state



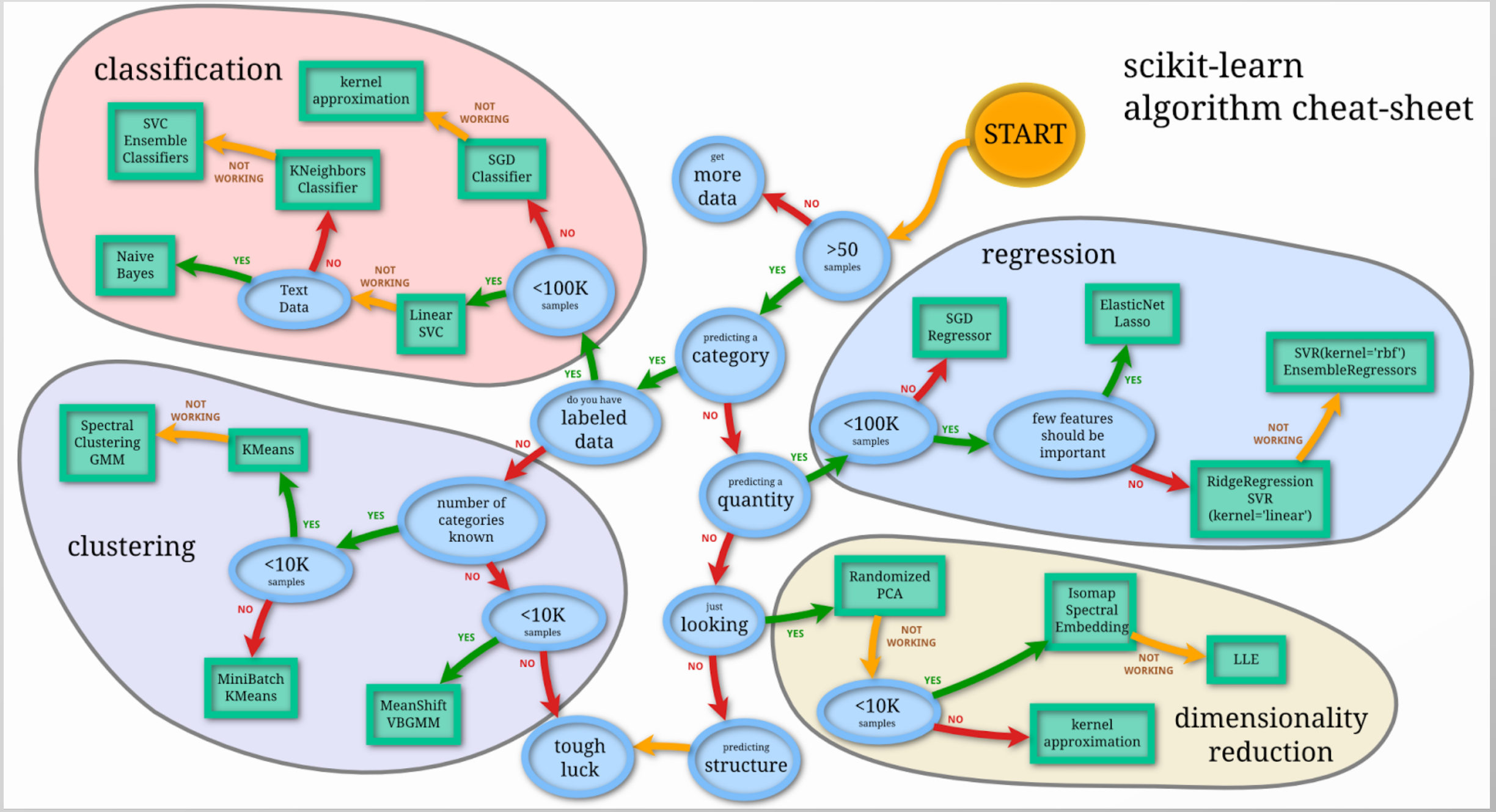


Concepts

- Prediction vs Inference
  - Prediction: aim to accurately predict the response of future
  - Inference: understand the relationship between x's and y
- Regression vs Classification
  - Regression: quantitative response
  - Classification: qualitative response and logistic regression
  - K-nearest neighbors can be used in the case of either quantitative or qualitative responses
- Variance vs Bias
  - Variance refers to the error due to the complex model trying to fit the data. High variance means the model passes through most of the data points and it results in over-fitting the data.
  - Bias refers to the error due to the model's simplistic assumptions in fitting the data. A high bias means that the model is unable to capture the patterns in the data and this results in under-fitting.
  - Good test set performance of a statistical learning method requires low variance as well as low squared bias.
  - Bias-Variance Trade-Off

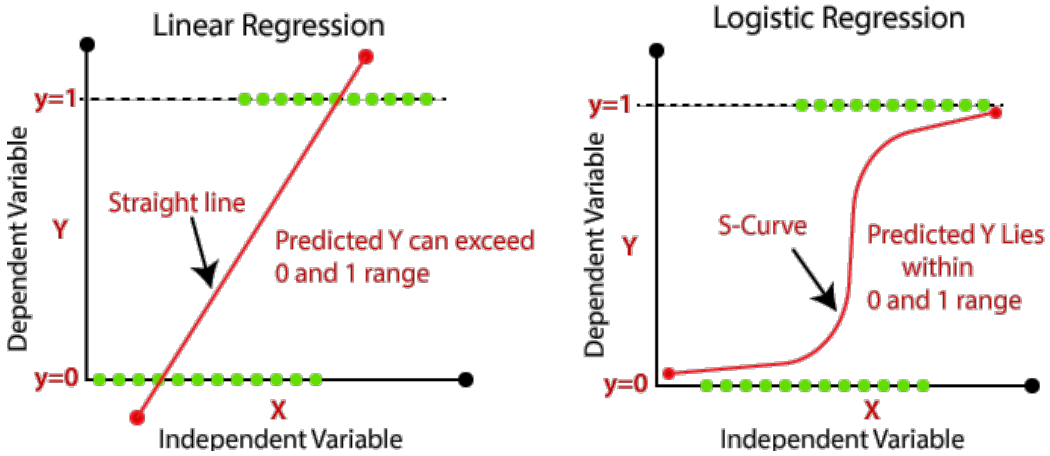
Parametric vs Non-parametric Methods

- Parametric methods involve a two-step model-based approach. It reduces the problem of estimating f down to one of estimating a set of parameters.
  - First, make an assumption about the functional form, or shape, of f. (e.g.: linear model)
  - After a model has been selected, we need a procedure that uses the training data to fit or train the model. (e.g.: least squares)
- Non-parametric methods do not make explicit assumptions about the functional form of f. Instead they seek an estimate of f that gets as close to the data points as possible without being too rough or wiggly.
  - Major advantage over parametric approaches: by avoiding the assumption of a particular functional form for f, they have the potential to accurately fit a wider range of possible shapes for f.
  - Major disadvantage: since they do not reduce the problem of estimating f to a small number of parameters, a very large number of observations (far more than is typically needed for a parametric approach) is required in order to obtain an accurate estimate for f.
- Parametric or linear machine learning algorithms often have a high bias but a low variance. (poor fit)
- Nonparametric or nonlinear machine learning algorithms often have a low bias but a high variance. (overfitting)


scikit-learn
algorithm cheat-sheet
START
get more data
NO
>50 samples
YES
classification
kernel approximation
NOT WORKING
SVC Ensemble Classifiers
NOT WORKING
KNeighbors Classifier
SGD Classifier
NO
Naive Bayes
YES
Text Data
NO
NOT WORKING
Linear SVC
YES
<100K samples
YES
predicting a category
YES
do you have labeled data
NO
regression
SGD Regressor
NO
ElasticNet Lasso
YES
SVR(kernel='rbf') EnsembleRegressors
<100K samples
YES
few features should be important
NOT WORKING
NO
RidgeRegression SVR (kernel='linear')
NO
predicting a quantity
YES
clustering
Spectral Clustering GMM
NOT WORKING
KMeans
YES
number of categories known
YES
<10K samples
NO
MiniBatch KMeans
NO
<10K samples
YES
MeanShift VBGMM
NO
just looking
YES
Randomized PCA
NOT WORKING
Isomap Spectral Embedding
NOT WORKING
LLE
YES
<10K samples
NO
kernel approximation
dimensionality reduction
NO
predicting structure
tough luck

Linear Regression vs Logistic Regression

- Linear Regression
  - Simple linear regression $y = B_0 + B_1 x$
  - The goal is to find the best estimates for the coefficients to minimize the errors in predicting y from x.
- Logistic Regression
  - A key difference from linear regression is that the output value being modeled is a binary values (0 or 1) rather than a numeric value.
  - Logistic regression $y = \frac{e^{B_0+B_1 x}}{1+e^{B_0+B_1 x}}$
  - Logistic regression models the probability that an input (X) belongs to the default class (Y = 1). $P(X) = P(Y = 1|X)$
  - Maximum-likelihood Estimator (MLE) for logistic regression is to find best estimates for the coefficients that maximize the probability of observing the data (e.g. probability of 1 if the data is the primary class).



Decision Tree vs bagging vs Random Forest

Decision Tree

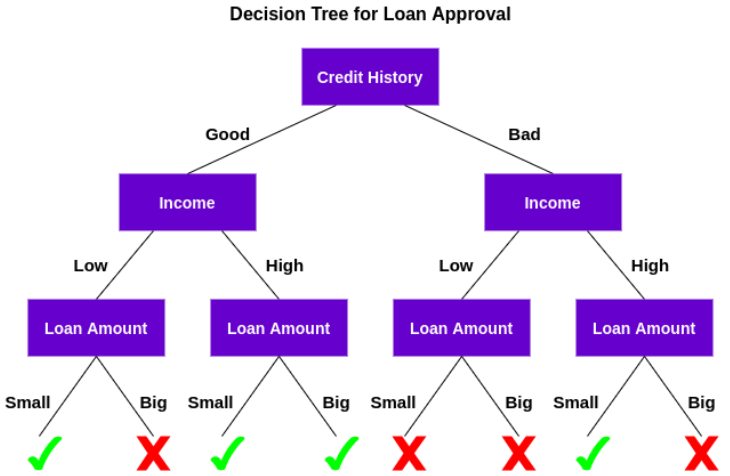
- Invariant under scaling and various other transformations of feature values
- Robust to inclusion of irrelevant features
- However, high correlation in the predictions -> **less accuracy and easily overfitting.**

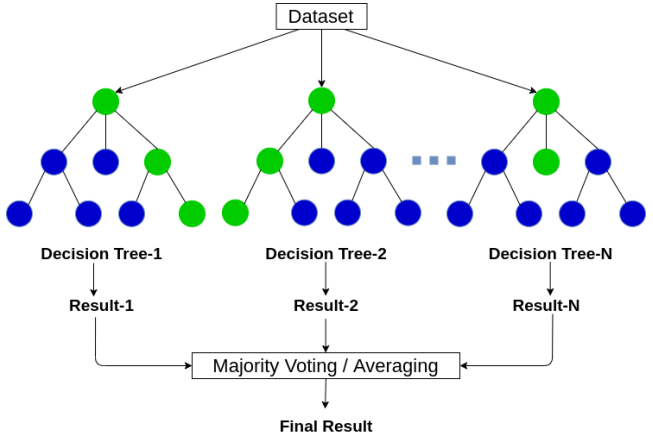
Bagging (Bootstrap Aggregation)

- Given a standard training set D of size n, bagging generates m new training sets D_i, each of size n', by **random sampling** from D uniformly and **with replacement**. This kind of sample is known as a bootstrap sample.
- Reduce the variance for those algorithm that have high variance (overfitting).

Random Forest

- Random forests provide an improvement over bagged trees by way of a small edit that decorrelates the trees.
- At each split in the tree, the algorithm forces each split to consider only a subset of the predictors.



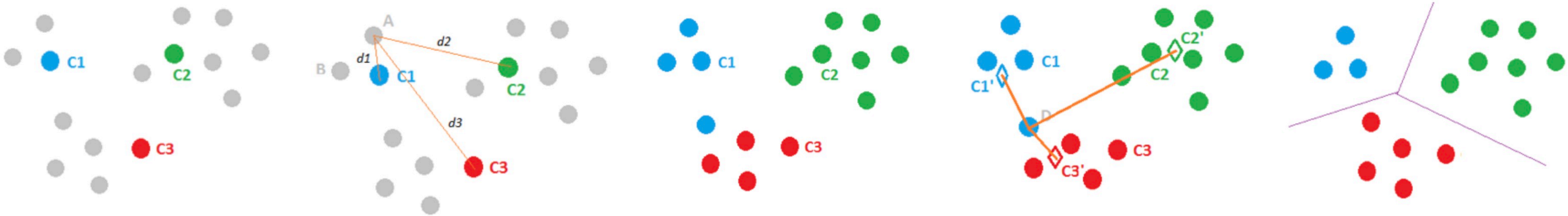


K-Means Clustering

- k-means clustering aims to find the set of k clusters such that every data point is assigned to the closest center, and the sum of the distances of all such assignments is minimized.
- Procedures (pick k = 3):
  1. Randomly pick three points C1, C2 and C3, and label them as centers of 3 clusters.
  2. Assign observations to the closest cluster center.
  3. Revise cluster centers as mean of assigned observations.
  4. Repeat step 2 and step 3 until convergence.

Limitation:

- The k-means algorithm converges to local optimum. (not necessarily global optimal)
- The initialization of the centers is critical to the quality of the solution found. (maybe hard to converge)
- Selection of k is tricky.



Association Rule Mining

- Association rule mining finds interesting associations and relationships among large sets of data items. This rule shows how frequently an itemset occurs in a transaction.
- Support: how popular an itemset is, as measured by the proportion of transactions in which an itemset appears.
  - Support(apple) = 4/8, Support(beer) = 6/8
- Confidence: how likely item Y is purchased when item X is purchased, expressed as {X -> Y}. This is measured by the proportion of transactions with item X, in which item Y also appears.
  - Confidence(apple -> beer) = 3/4
- Lift: how likely item Y is purchased when item X is purchased, while controlling for how popular item Y is.
  - Lift(apple -> beer) = (3/8)/(24/64) = 1
  - A lift value greater than 1 means that item Y is likely to be bought if item X is bought, while a value less than 1 means that item Y is unlikely to be bought if item X is bought.

Transaction 1	🍎 🍺 ⚪ 🍗
Transaction 2	🍎 🍺 ⚪
Transaction 3	🍎 🍺
Transaction 4	🍎 🍐
Transaction 5	🍼 🍺 ⚪ 🍗
Transaction 6	🍼 🍺 ⚪
Transaction 7	🍼 🍺
Transaction 8	🍼 🍐

$$\text{Support } \{🍎\} = \frac{4}{8}$$

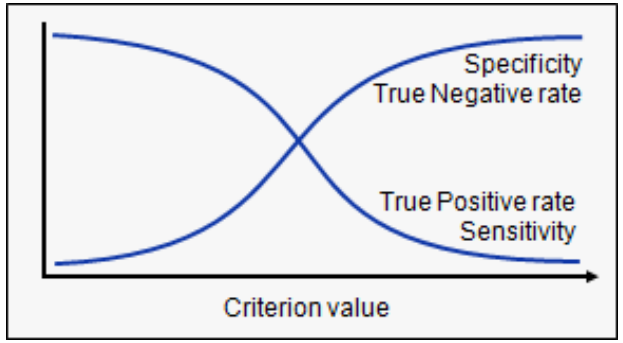
$$\text{Confidence } \{🍎 \rightarrow 🍺\} = \frac{\text{Support } \{🍎, 🍺\}}{\text{Support } \{🍎\}}$$

$$\text{Lift } \{🍎 \rightarrow 🍺\} = \frac{\text{Support } \{🍎, 🍺\}}{\text{Support } \{🍎\} \times \text{Support } \{🍺\}}$$

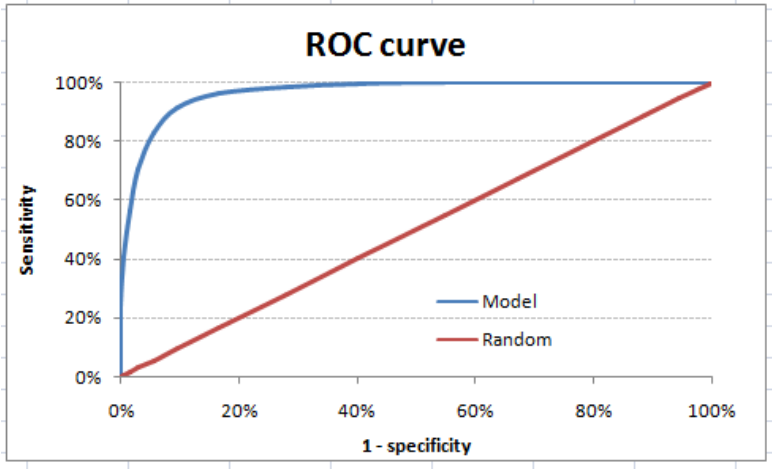
Model Evaluation Metrics

- Confusion Matrix

Confusion Matrix		Target			
		Positive	Negative		
Model	Positive	a	b	*Positive Predictive Value*	a/(a+b)
	Negative	c	d	*Negative Predictive Value*	d/(c+d)
		Sensitivity	*Specificity*	**Accuracy** = (a+d)/(a+b+c+d)	
		a/(a+c)	d/(b+d)		



- Area Under the ROC curve
  - The ROC curve is the plot between sensitivity and (1- specificity).
    - (1- specificity) is also known as false positive rate.
    - sensitivity is also known as True Positive rate.
  - The biggest advantage of using ROC curve is that
    - it is independent of the change in proportion of responders.
  - From the plot, AUC ROC ~ 97%. Good model.

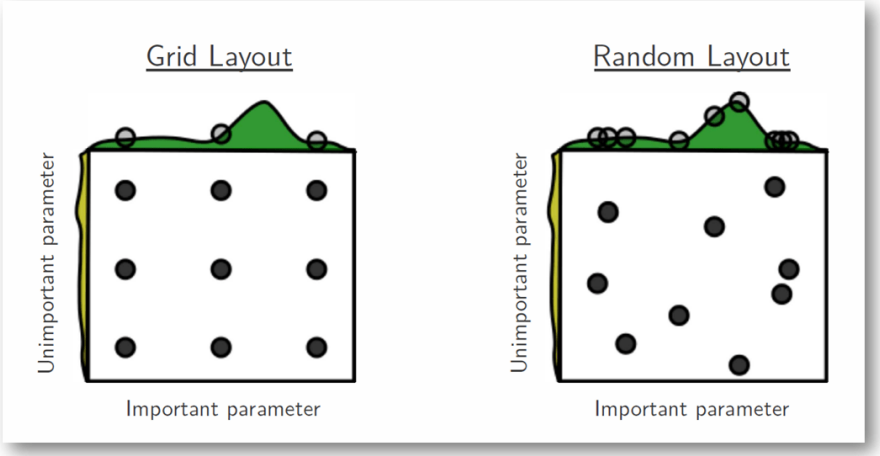


Loss Functions

- Model fitting in machine learning is usually introduced in the context of optimizing a loss function. Loss functions also arise in optimization and economics, these are also called objective functions or negative utility functions.
- The loss function is written in several different ways such as $L(y, \hat{y}) \; or \; J(\theta) = J(\theta; x)$ to emphasize what component is being minimized/maximized. **Loss functions maps decisions to their associated costs.**
- Some common loss functions:
  - Hinge loss: $L(f(x; \omega); y) = \max(0, 1 - y_i * f(x; \omega))$, where $y_i \in -1, 1$, used with SVMs
  - Cross-entropy loss/Negative Log Likelihood): $L(f(x; \omega); y) = \sum_{i=1}^{n} \sum_{c=1}^{C} y_{i,c} \log\big(f(x; \omega)\big)$, used in multi-class classification where C is the set of classes
  - Squared error loss: $L(f(x; \omega); y) = \big|\big|y - f(x; \omega)\big|\big|_2^2$, used in regression
  - MAE(mean absolute error or L1 loss), MSE(mean squared error or Quadratic Loss or L2 loss), used in regression
- In dimension reduction, an important loss function is the reconstruction error which measures the difference between our learned function and the observed data: $L(X, f(X)) = \big|\big|X - f(X)\big|\big|^2$. Reconstruction is used in unsupervised learning and methods like PCA and autoencoders.

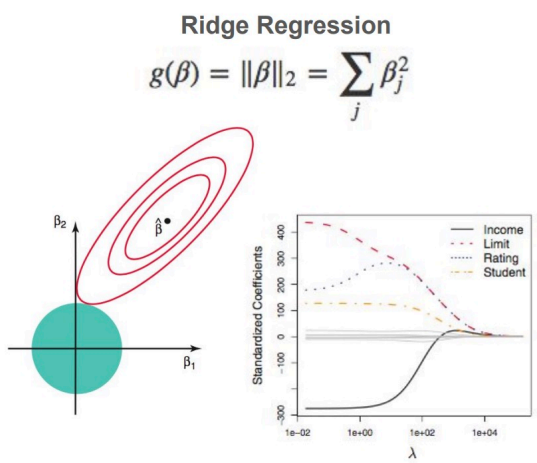
Hyperparameter Optimization (Model Selection)

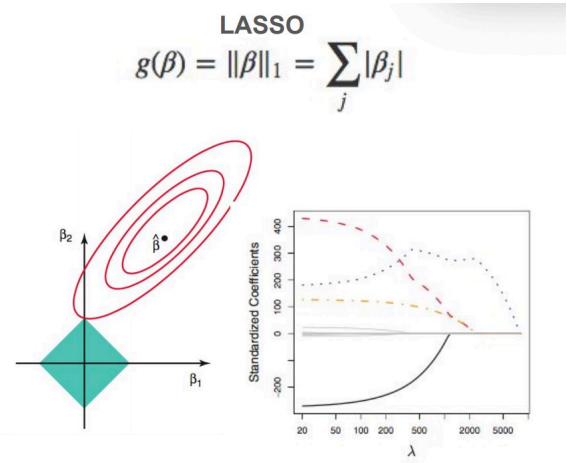
- A model hyperparameter is a configuration we make that is external to the model and whose value cannot be learned during model fitting.
- In a Random Forest model, we choose:
  - Tree Depth
  - Number of Trees
  - Fraction of variables to sample
- In a K-Nearest Neighbors model, we choose:
  - K
- In a regularized regression model (LASSO or Ridge), we choose
  - Regularization penalty
- Models can have many hyperparameters and finding the best combination of parameters can be treated as a search problem.
  - grid search
  - random search



Regularization

- The goal of regularization in a model is to enforce generalizability - that is to balance our bias and variance so that when new data points are observed it fits the trend rather than the specific points.
- An easy way to think of the bias-variance trade-off is in terms of model performance
  - The question is are we willing to give up a bit of accuracy (bias) to increase the precision (decrease variability).
  - Obviously having a minimum variance and unbiased estimate is optimal however it is frequently not possible.
- By using regularization, we want to **penalize or adjust each weights of the independent variables** so that it makes a good prediction on test set that it has not seen before.
  - Lasso regression(L1) **can lead to zero coefficients**, i.e. some of the features are completely neglected for the evaluation of output. Therefore, lasso regression not only helps in reducing over-fitting, but it can help us in feature selection.
  - Ridge regression(L2) **shrinks the coefficients**, and it helps to reduce the model complexity and multicollinearity.





Machine learning to Deep learning

ARTIFICIAL INTELLIGENCE

Early artificial intelligence stirs excitement.

MACHINE LEARNING

Machine learning begins to flourish.

DEEP LEARNING

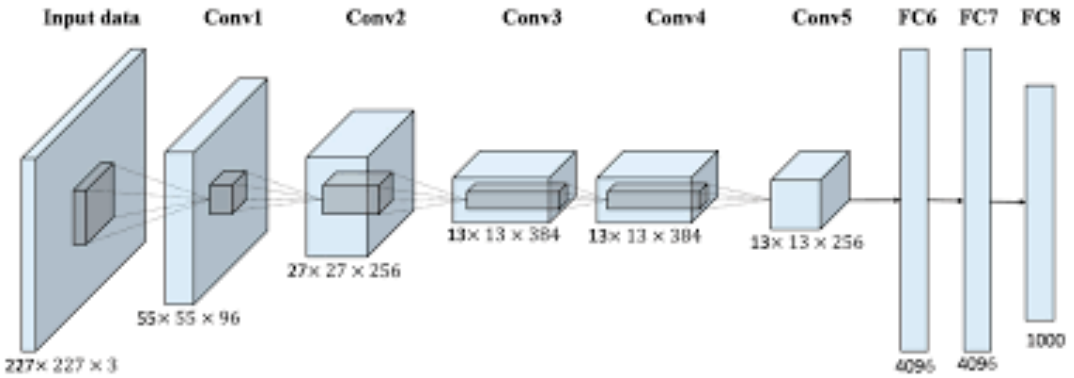
Deep learning breakthroug drive AI boom.

1950's 1960's 1970's 1980's 1990's 2000's 2010's

Deep Learning Success

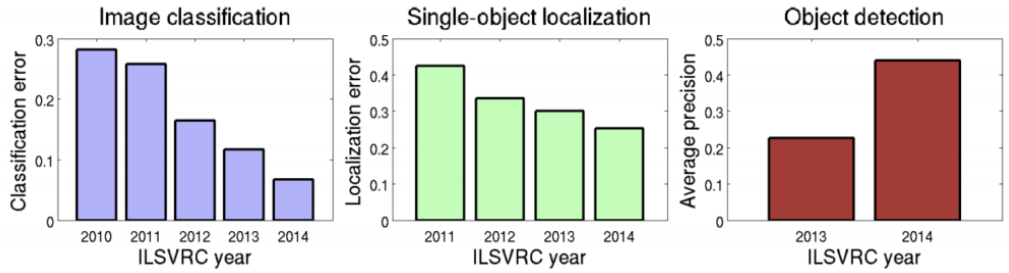
- Inflection point was in 2012 when convolutional net dominated the ImageNet Large Scale Visual Recognition Competition (ILSVRC).
- Until 2010 the top methods predicted image classes with 75% accuracy.

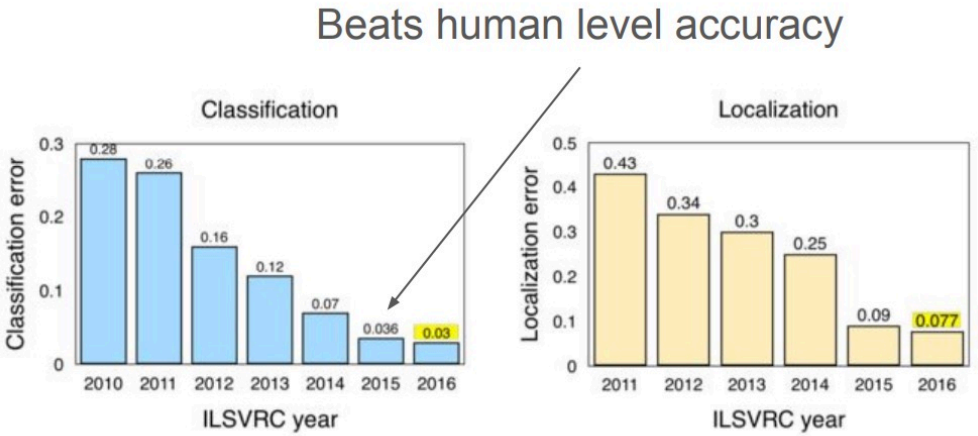
- In 2012, researchers Alex Krizhevsky, Geoffrey Hinton, and Ilya Sutskever entered a ConvNet (now called Alexnet) into the challenge.
- This convolutional neural network initially contained only eight layers – five convolutional followed by three fully connected layers – and strengthened the speed and dropout using rectified linear units.
- Its success kicked off a convolutional neural network era in the deep learning community.



Deep Learning Success

- The ILSVRC tasks have led to milestone model architectures and techniques in the intersection of computer vision and deep learning.
  - Image classification: Predict the classes of objects present in an image.
  - Single-object localization: Image classification + draw a bounding box around one example of each object present.
  - Object detection: Image classification + draw a bounding box around each object present.
- AlexNet could predict classes with an incredible accuracy of ~85%
- Human level accuracy is ~95%





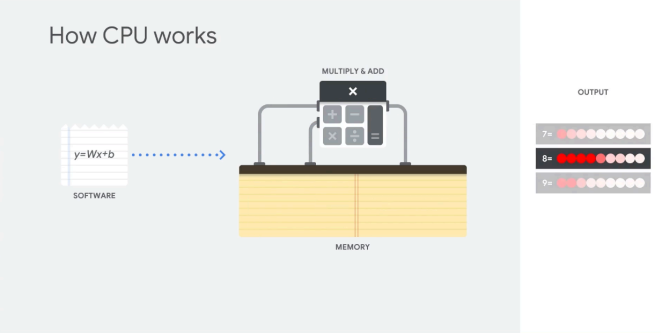
In 2016 the contest was considered solved as most of the entries were reaching near-human level accuracy

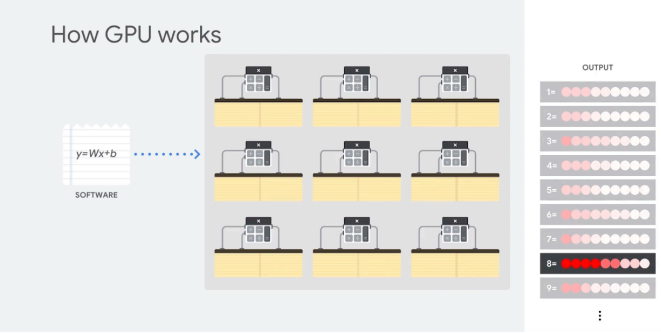
Deep Learning Success

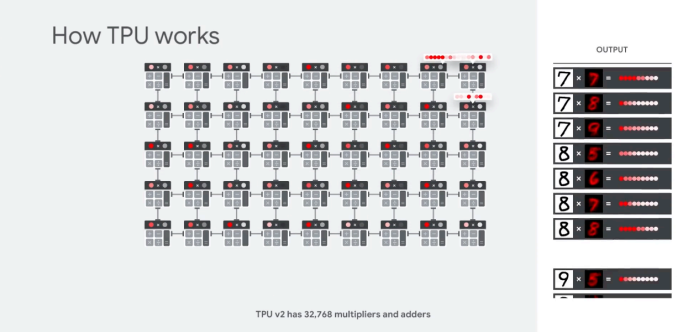
- Kaggle has played an important in popularizing ML.
- Early adopters won $100Ks competitions.
- Current opening competitions
  - Answer Correctness Prediction
    - Deadline: January 7, 2021
    - $100,000
  - Jane Street Market Prediction
    - Deadline: February 22, 2021
    - $100,000
  - Cassava Leaf Disease Classification
    - Deadline: February 18, 2021
    - $18,000
  - Rainforest Connection Species Audio Detection
    - Deadline: February 17, 2021
    - $15,000

3 FACTORS CONTRIBUTING TO DL'S SUCCESSES

- Hardware





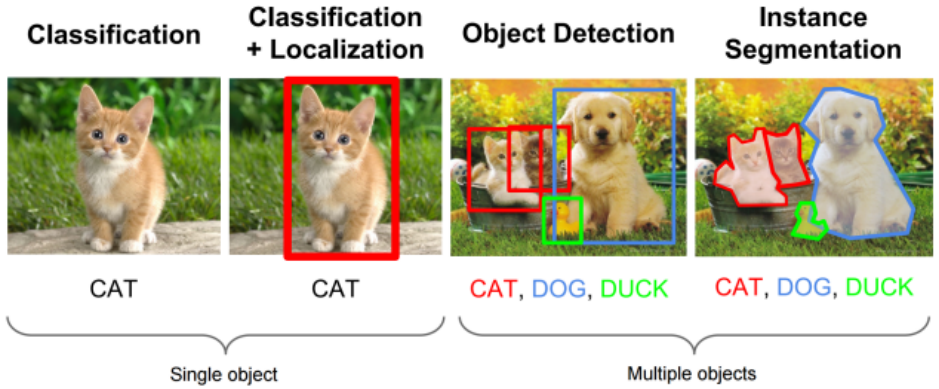


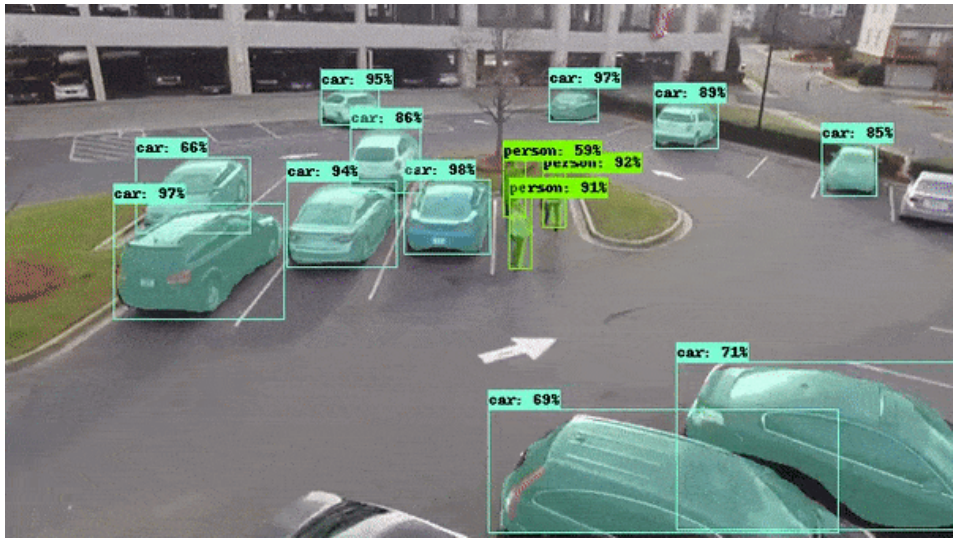
- Data
  - Speed and affordability
    - NVMe drives are 25x quicker than conventional HDDs.
    - Cost per GB decreases yearly.
  - Availability
    - Easy access to data via APIs
    - Sites collect and share data
  - Amount
    - The amount of data required to train a deep learning models is massive
    - In the early 2000s, a standard hard drive had between 10-40Gb. Most production-level corpora well exceed the 50Gb mark.

- Algorithm
  - Specialized algorithm
    - Developed to leverage general purpose hardware for DL
  - Algorithmic & Research advances
    - Logistic vs ReLu networks
    - Embeddings
    - Batch norm
    - Residual networks
    - Attention

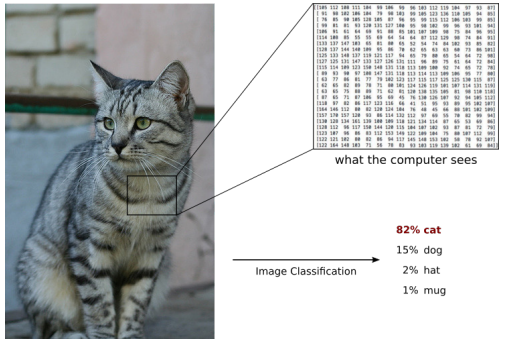
Deep Learning Application

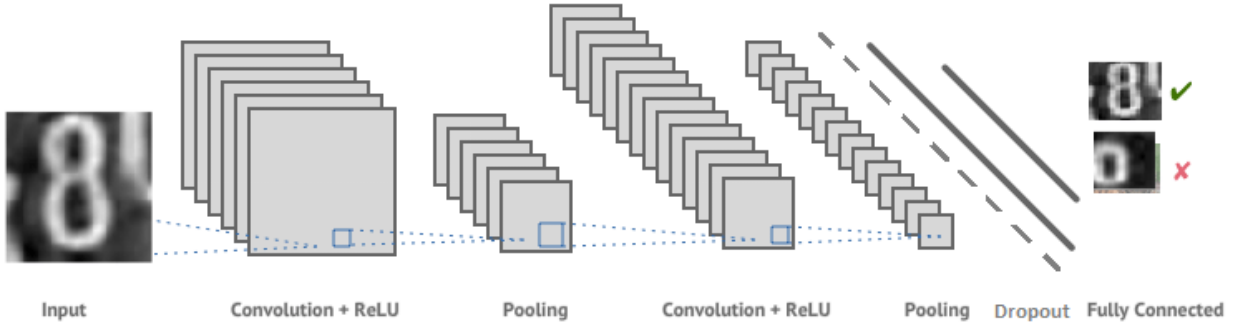
- Object Detection

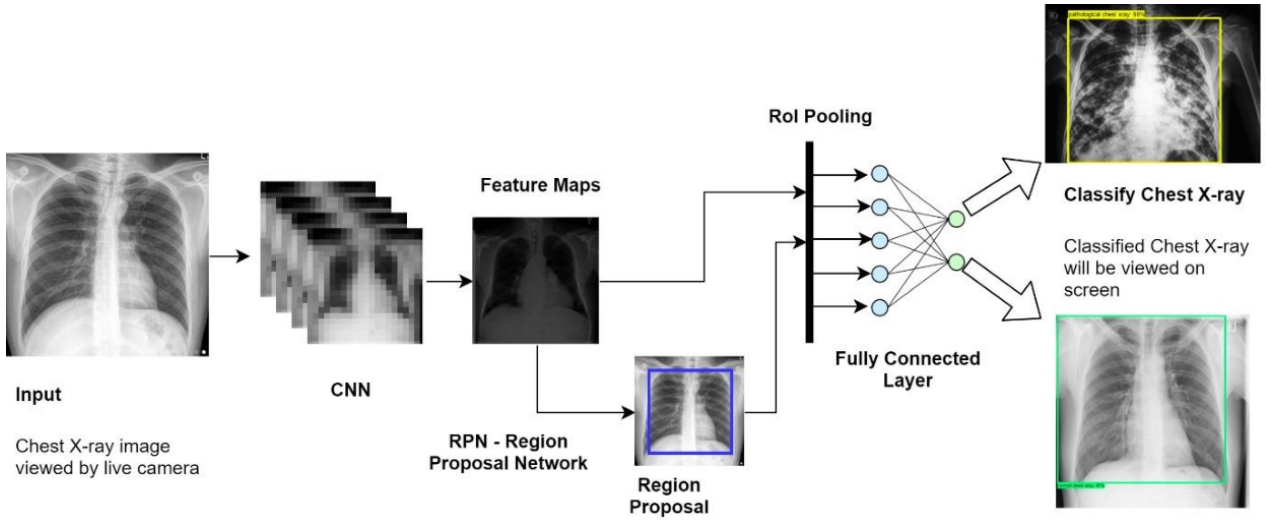




- Image Classification

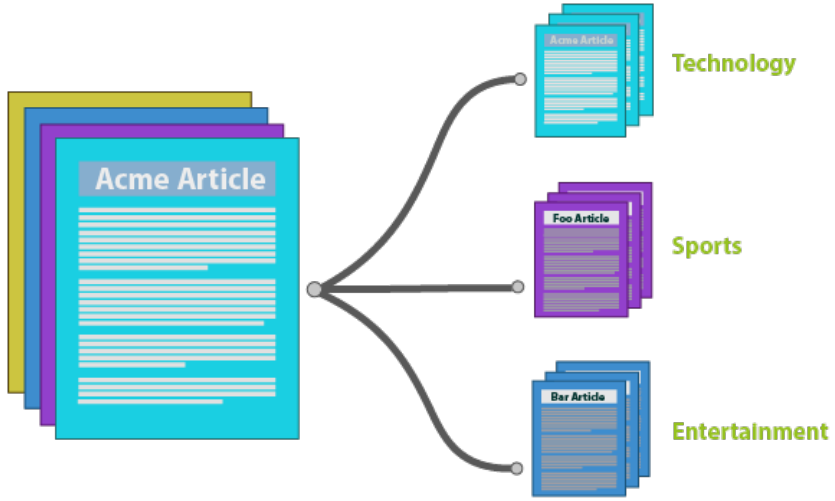


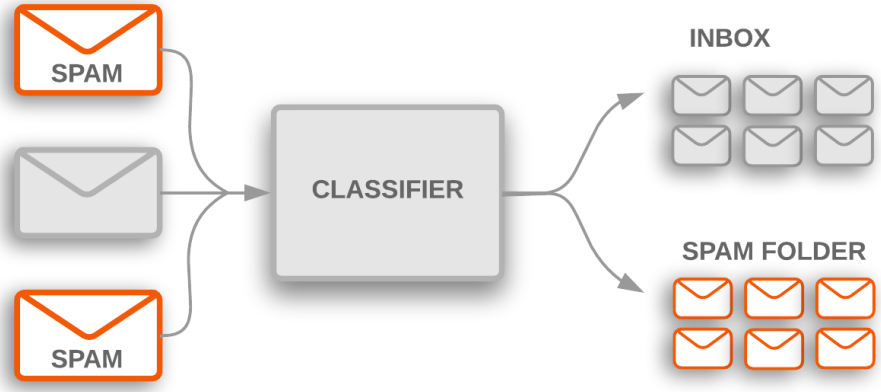




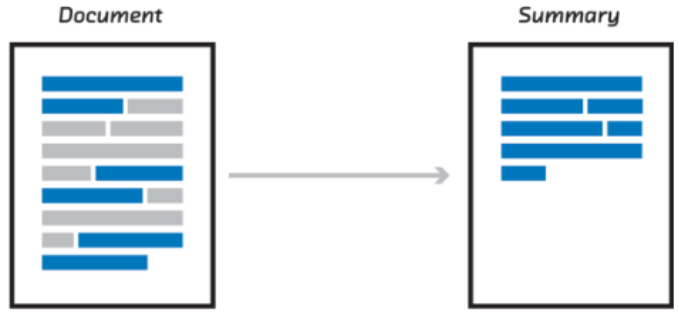
Deep Learning Application

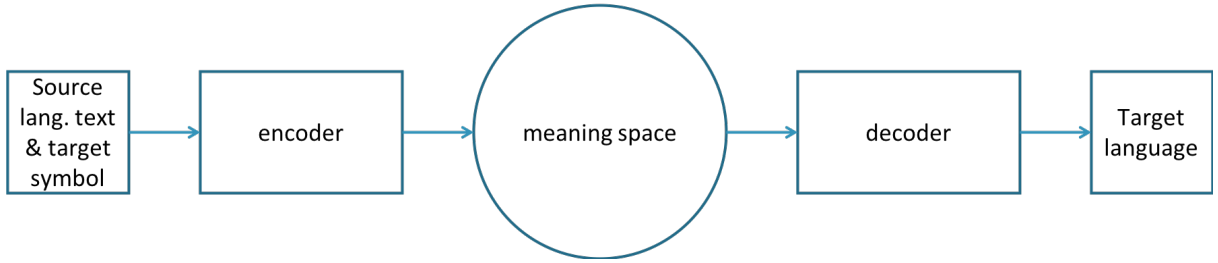
- Text Classification





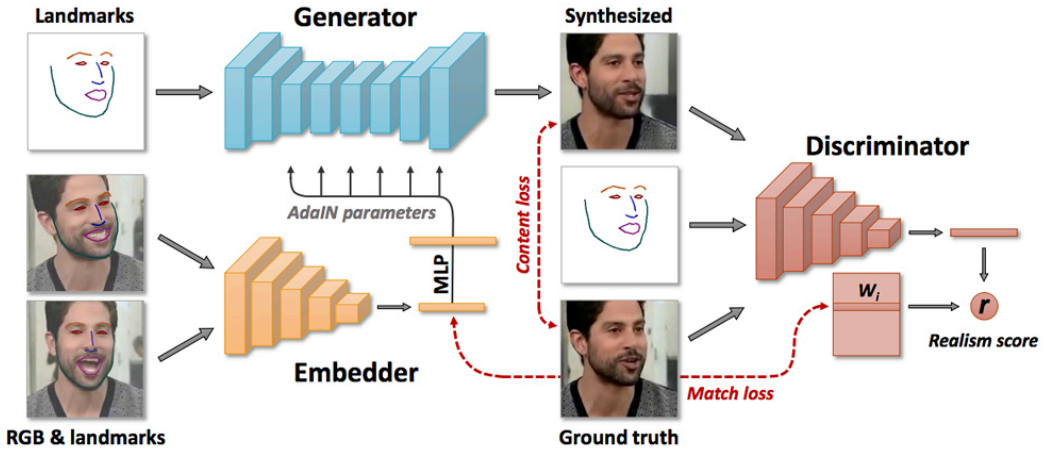
- Text Generation

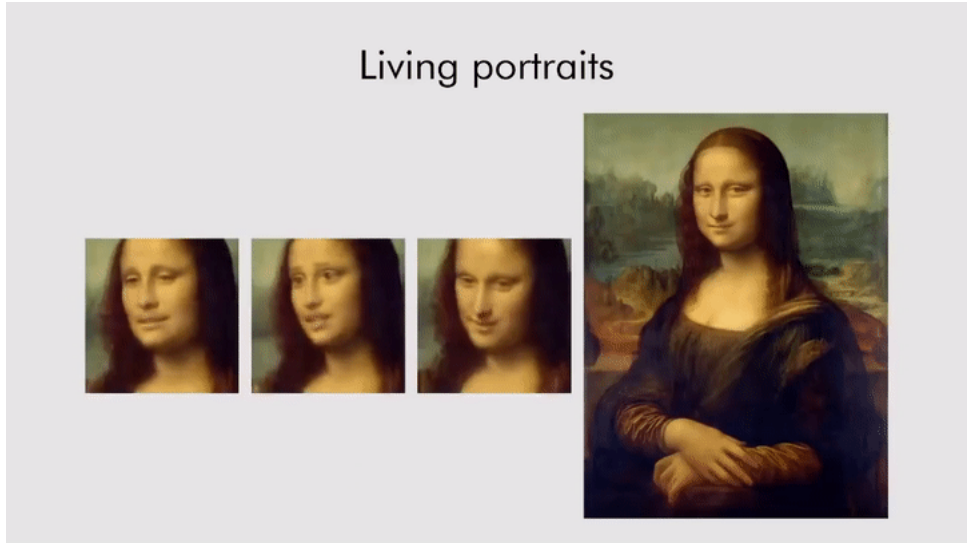




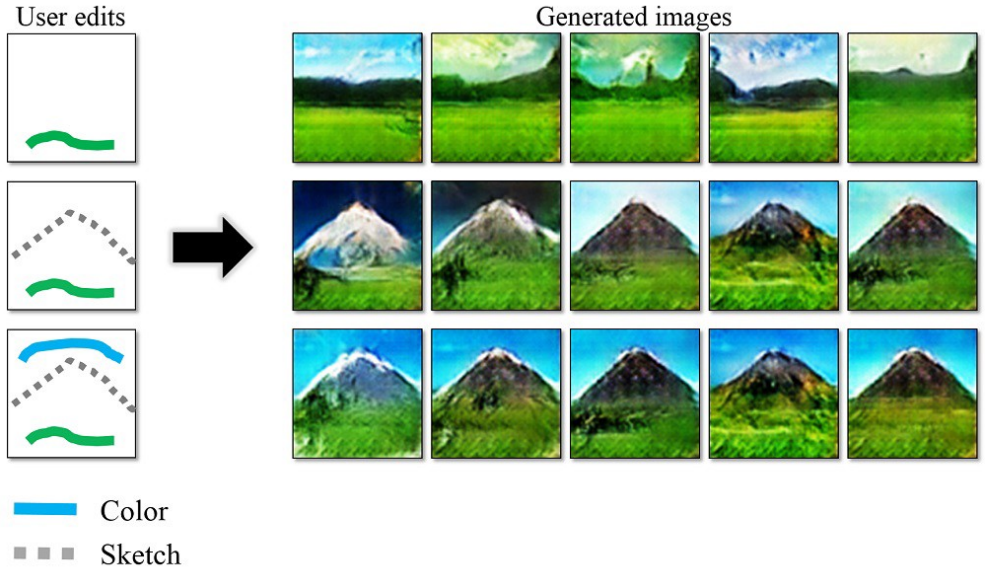
Deep Learning Application - GAN

- Deep Fake



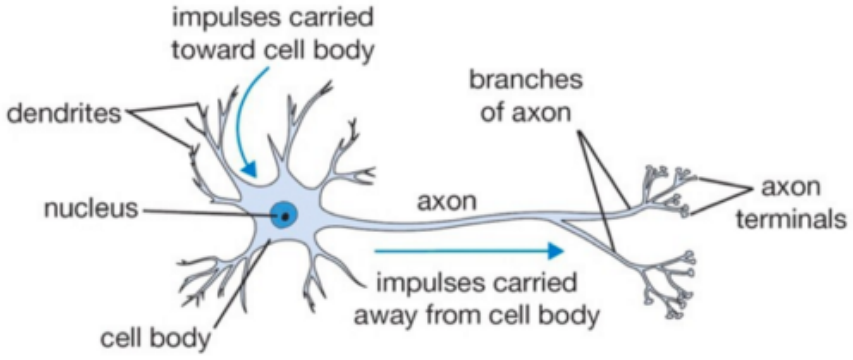


- GAN Art

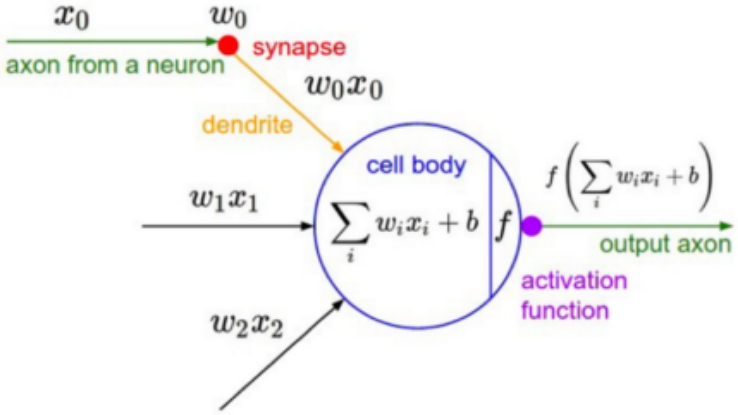


Deep Learning Basic

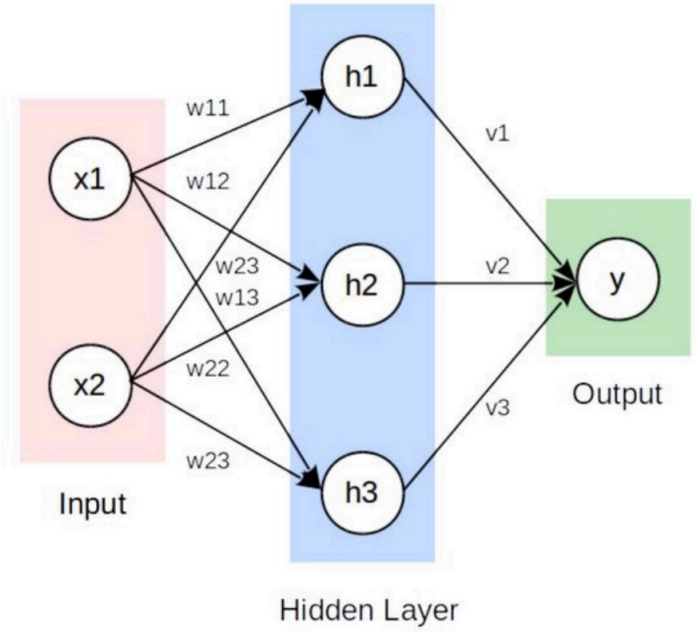
- Biological Inspiration
  - A cell receives lots of inputs into its dendrites
  - These cause small fluctuations in membrane voltage
  - When enough inputs occur at nearly the same time, and threshold is reached, and the cell fires a “spike” or action potential
  - This is how the cell sends a signal along to other downstream cells.



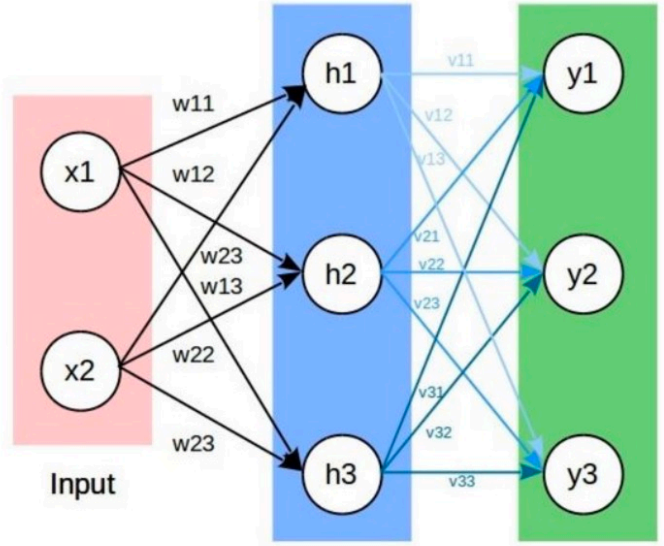
- Artificial Neural Networks
  - ANNs are flexible mathematical functions
  - Composed of “hidden units” which are inspired by biological neurons
    - They have inputs
    - They compute a weighted sum of those inputs
    - They output any non-linear transformation of that sum
  - Their inputs and/or outputs can be other such hidden units
  - Stacking them in this way allows them to represent a rich set of functions.



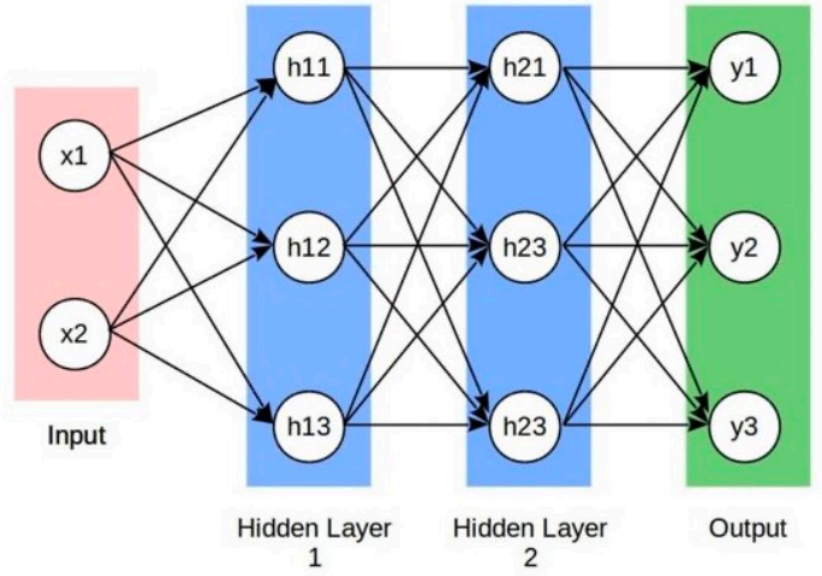
ANN Diagrammatic Representation and Notation



- 2-D inputs
- 1-D outputs
- A single hidden layer with 3 hidden nodes



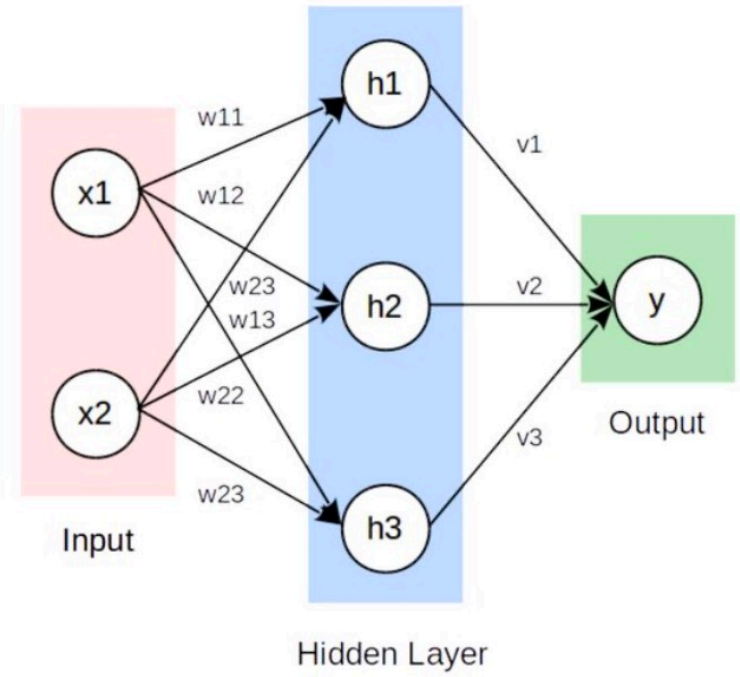
- 2-D inputs
- 3-D outputs
- A single hidden layer with 3 units

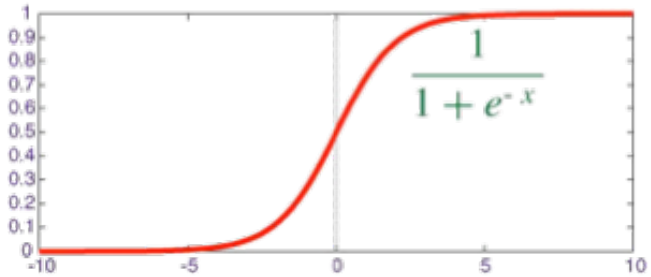


- 2-D inputs
- 3-D outputs
- Multiple hidden layers
  - 3 units in 1st hidden layer
  - 3 units in 2nd hidden layer

TensorFlow Playground

Notation and Computation





- Each unit can be computed as two parts:
  - Linear part: weighted sum of inputs (plus bias)

$$a_i = w_{1i}x_1 + w_{2i}x_2 + b_i$$

  - Non-linear part: transformation of that sum by a nonlinearity of our choosing.

$$h_i = f(a_i)$$

$$\vec{x}^T = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad \vec{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$W = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \end{bmatrix}$$

$$\vec{a} = \vec{x}W + \vec{b}$$

$$\vec{h} = f(\vec{x}W + \vec{b})$$

$$\begin{aligned} y &= g(v_1h_1 + v_2h_2 + v_3h_3 + c) \\ &= g(v_1 f(w_{11}x_1 + w_{12}x_2 + b_1) \\ &+ v_2 f(w_{21}x_1 + w_{22}x_2 + b_2) \\ &+ v_3 f(w_{31}x_1 + w_{32}x_2 + b_3) \\ &+ c) \end{aligned}$$

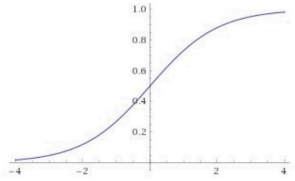
Activation Functions

- The purpose of the activation function is to **introduce non-linearity** into the output of a neuron.

- Sigmoid: not blowing up activation
- Relu: not vanishing gradient
- Relu: More computationally efficient to compute than Sigmoid like functions since Relu just needs to pick max(0, x) and not perform expensive exponential operations as in Sigmoids.
- Relu : In practice, networks with Relu tend to show better convergence performance than sigmoid.

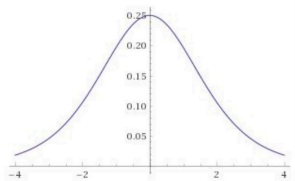
Sigmoid

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



- This function is familiar to us
- Logistic activation, easy to reason about

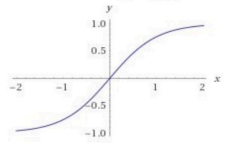
$$\frac{d}{dx}\sigma(x) = \sigma(x)(1-\sigma(x))$$



- This function saturates and its gradients are small
- This function isn't 0-centered
- Useful for understanding ANNs, but not used much in practice anymore

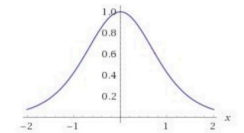
Tanh

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



- This function is 0-centered, behaves a little better
- Derivative is little stronger

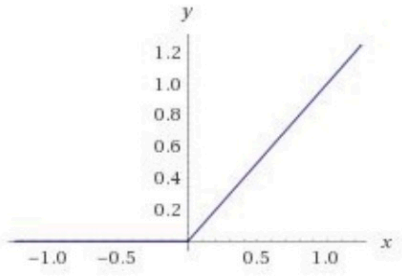
$$\frac{d}{dx}\tanh(x) = 1-\tanh^{2}(x)$$



- If you're looking for a smooth S-shaped curve, try this one instead of sigmoid

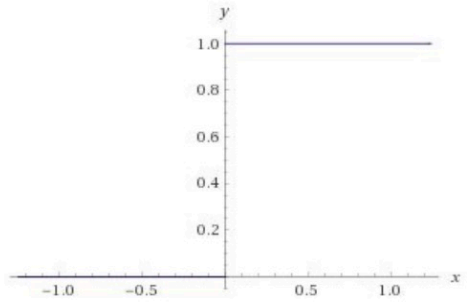
ReLU

$$\mathrm{ReLu}(x) = \max(0, x)$$



- This function doesn't saturate at large values of x
- The derivative does not saturate at large values of x

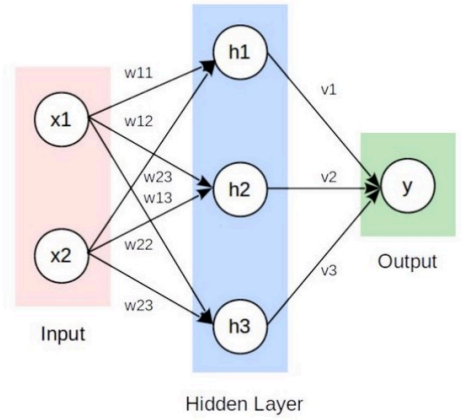
$$\frac{d}{dx}(\max(0, x)) = \begin{cases} 0 & x < 0 \\ 1 & x > 0 \end{cases}$$



- ReLU should be your go-to activation function

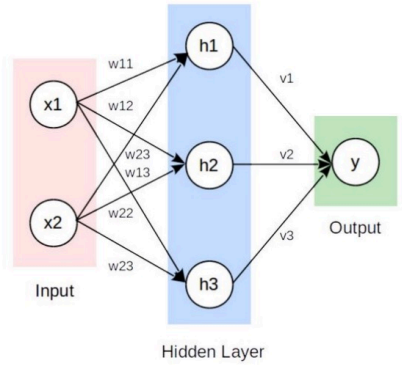
Output Functions

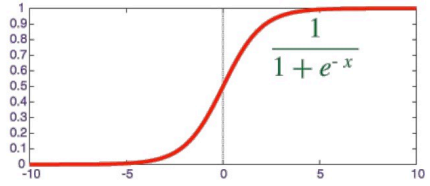
- Linear



- For continuous target variable
- $\hat{y} = \vec{v}^T \vec{h} + c$
- Accompanying Loss Function
- Squared Error Loss
- $L = \frac{1}{N}\frac{1}{2}\sum_i((y_i - \hat{y}_i)^2)$
- $\frac{\partial L}{\partial \hat{y}_i} = y_i - \hat{y}_i$

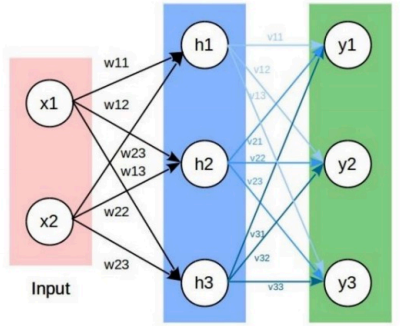
- Sigmoid





- For binary target variables (Bernoulli)
- $\hat{y} = \sigma(\vec{v}^T \vec{h} + c)$
- Intended to represent a probability over the two classes. It "squashes" a real-valued scalar to lie within [0,1].
- Binary Cross Entropy
- $L = y_i \log(\hat{y}_i) - (1 - y_i)\log(1 - \hat{y}_i)$
- $\frac{dL}{d\hat{y}_i} = \frac{y_i}{\hat{y}_i} - \frac{1-y_i}{1-\hat{y}_i}$

- Softmax



- For categorical outputs
- $\vec{z} = W^T \vec{h} + \vec{b}$
- $\hat{y}_j = \frac{e^{z_j}}{\sum_k e^{z_k}}$
- Converts real-valued vector (logits) into a probability vector over K classes. Similar to Sigmoid in how it squashes input into a valid probability vector.
- Categorical Cross Entropy
- $L = -\sum_j y_i \log(\hat{y}_i) = -\vec{y}^T \log(\vec{\hat{y}_i})$
- $\frac{dL}{d\hat{y}_i} = \vec{\hat{y}_i} - \vec{y}$

Week three and four:
tensorflow and pytorch implementation

- Feedforward Networks
- Backprop and Inference
- Convolutional Networks
- Recurrent Networks
- Generative Adversarial Networks