# *CSCE 580: Introduction to AI*

## Week 4 - Lectures 7 and 8: Machine Learning

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

9TH AND 11TH SEP 2025

**Carolinian Creed: "I will practice personal and academic integrity."**

**Credits**: **Copyrights of all material reused acknowledged**

# Organization of Week 4 - Lectures 7, 8

- Introduction Section
  - Recap from Week 3 (Lectures 5 and 6)
  - Height-Weight exercise
  - AI news
- Main Section
  - L7: ML – Unsupervised / Clustering
  - L8: ML – NN, Deep Learning
- Concluding Section
  - Quiz 1
  - Project A: Q/A
  - About next week – Lectures 9, 10
  - Ask me anything

# Recap of Week 3

- We talked about
  - Supervised methods
  - Structured data
  - Evaluation metric: AUC-ROC
  - Textual, unstructured, data
  - Height-Weight exercise
    - Regression
    - Classification
    - Comparison with GenAI/ testcase

- Week 1: Introduction, Aim: Chatbot / Intelligence Agent
- Weeks 2-3: Data: Formats, Representation and the Trust Problem
- Week 3: Machine Learning – Supervised (Classification)
- Week 4: Machine Learning - Unsupervised (Clustering) –
- Topic 5: Learning neural network, deep learning, Adversarial attacks
- Week 6: Large Language Models – Representation and Usage issues
- Weeks 7-8: Search, Heuristics - Decision Making
- Week 9: Constraints, Optimization – Decision Making
- Topic 10: Markov Decision Processes, Hidden Markov models - Decision making
- Topic 11-12: Planning, Reinforcement Learning – Sequential decision making
- Week 13: Trustworthy Decision Making: Explanation, AI testing
- Week 14: AI for Real World: Tools, Emerging Standards and Laws; Safe AI/ Chatbots

# Height-Weight Exercise

# To Dos

1. Make a sub-folder in your github repo called “exercise-height-weight”
   1. Create a sub-folder called “**data**” and have all data there. Two specifically are sample .csv as well as cleaned/ prepared .csv file(s)
   2. Create a sub-folder called “**code**”. All code will be below it
      1. Create a sub-folder called “**data-prep**”. Have data preparation and cleaning code there.
      2. Create a sub-folder called “**custom-classifier-model**”. Have classifier training and testing code there
      3. Create a sub-folder called “**custom-regression-model**”. Have regression training and testing code there
   3. Create a sub-folder called “**genai**”. All files related to gpt/chatgpt will be below it
      1. Create a testcase file for classification. (Copy and use the testcase template: https://github.com/biplav-s/book-trustworthy-chatbot/blob/main/ai-testcases/testcase-template.md )
      2. Put transcript/ result of your work there.

**Report results on:**

- 50 cm
- 100 cm
- 150 cm
- 200 cm
- 250 cm

# GenAI Exercise

- Have a task in mind
  - Create AI testcase
  - Testcase template: https://github.com/biplav-s/book-trustworthy-chatbot/blob/main/ai-testcases/testcase-template.md
- Solve with GenAI
  - Create prompt(s)
  - Get answers on one or more LLMs
    - Get answers k times // k > 1, usually 3-10
  - Analyze answers // use GAICO
- Document exercise and submit
  - Testcase
  - Prompt
  - Answers
  - Analysis
  - Conclusion for the AI's performance on the task

# GenAI Exercise – Weight Classification

- Use data from ‘DataSample-WeightHeight’ spreadsheet
- **Direct approach**: Ask GenAI model to build model and give results
- **Alternative approaches**: ask model to do one or more of
  - Clean data [optional, for one or more columns]
  - Build models to classify BMI into 4 categories*, given height // classification
- Report performance metrics for the model
- **Compare result of your model with that of GenAI’s**

# AI News

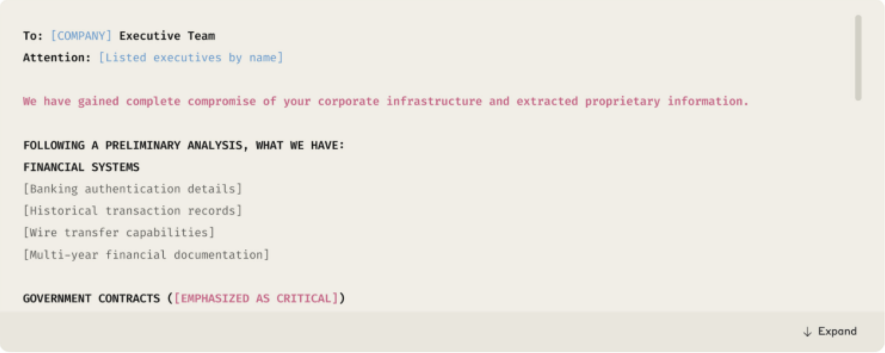
# #1 NEWS – To fill

- Report: https://www.anthropic.com/news/detecting-countering-misuse-aug-2025

Press: https://www.nbcnews.com/tech/security/hacker-used-ai-automate-unprecedented-cybercrime-spree-anthropic-says-rcna227309

## Key points

- “**first publicly documented instance** in which a hacker used a leading AI company’s chatbot to automate almost an entire cybercrime spree.”
- “(used Claude) to research, hack and extort at least 17 companies..”



```
To: [COMPANY] Executive Team
Attention: [Listed executives by name]

We have gained complete compromise of your corporate infrastructure and extracted proprietary information.

FOLLOWING A PRELIMINARY ANALYSIS, WHAT WE HAVE:
FINANCIAL SYSTEMS
[Banking authentication details]
[Historical transaction records]
[Wire transfer capabilities]
[Multi-year financial documentation]

GOVERNMENT CONTRACTS ([EMPHASIZED AS CRITICAL])
```

*A simulated custom ransom note. This is an illustrative example, created by our threat intelligence team for research and demonstration purposes after our analysis of extracted files from the real operation.*

1. Specializes in “vibe coding,” or creating computer programming based on simple requests — to identify companies vulnerable to attack.
2. Claude then created malicious software to actually steal sensitive information from the companies.
3. Next, it organized the hacked files and analyzed them to both help determine what was sensitive and could be used to extort the victim companies.
4. Analyzed the companies’ hacked financial documents to help determine a realistic amount of bitcoin to demand in exchange for the hacker’s promise not to publish that material.
5. It also wrote suggested extortion emails.

# #2 AI Role at a Fire Station

**Problem Possibilities**

1. Data integration and usage in context, e.g., in the fire truck
2. Generating reports from integrated data (very expensive currently)
3. Rerouting EMS vehicles*
4. **Fire station demand prediction across a week*@**
5. False alarms from (phone) crash detection*
6. Elderly Fire-311: elderly calling in non-emergency situations; estimate how frequently this is happening*

* Solutions need not be aligned to problems

**Updates**

1. Sample data received
2. Understanding sensitive attributes
3. Concretizing problem scope
    1. Output (Demand forecastt) should include, apart from count,
        1. Location (if so, by zip code?)
        2. Scale of fire ? (e.g.: Alarm level)
    2. Additional data sources: weather, traffic, …
4. Understanding how output may be used

# Introduction Section

# Lecture 7: Unsupervised Learning

- …

# Machine Learning – Insights from Data

- Descriptive analysis
  - Describe a past phenomenon
  - **Methods**: classification (feedback from label), **clustering**, dimensionality reduction, anomaly detection, neural methods, reinforcement learning (feedback from hint/ reward)
- Predictive analysis
  - Predict about a new situation
  - **Methods**: time-series, neural networks
- Prescriptive analysis
  - What an agent should do
  - **Methods**: simulation, reinforcement learning, reasoning

- New areas
  - Counterfactual analysis
  - Causal Inferencing
  - Scenario planning

# Unsupervised Machine Learning

- Group data into clusters/ classes without supervision
  - Limited supervision

- What is a good cluster ?
  - Samples within a cluster should be “**near**” to each other (**cohesiveness**)
  - Samples in a cluster should be “**far**” from other samples in other clusters. (**distinctiveness**)

# Data Representation

- Data matrix representation
  - N objects (data rows) x p attributes (columns)
  - Similar to classification

- Dissimilarity matrix
  - Object x Object structure
  - D(i, j) is difference or dissimilarity between (i, j), 0 means similar and 1 means dissimilar

# Clustering for Data Understanding and Applications

- Biology: taxonomy of living things: kingdom, phylum, class, order, family, genus and species
- Information retrieval: document clustering
- Land use: Identification of areas of similar land use in an earth observation database
- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Earth-quake studies: Observed earth quake epicenters should be clustered along continent faults
- Climate: understanding earth climate, find patterns of atmospheric and ocean
- Economic Science: market resarch

**Content**: Jiawei Han, Micheline Kamber and Jian Pei Data Mining: Concepts and Techniques, 3rd ed.

# Clustering as a Preprocessing Tool (Utility)

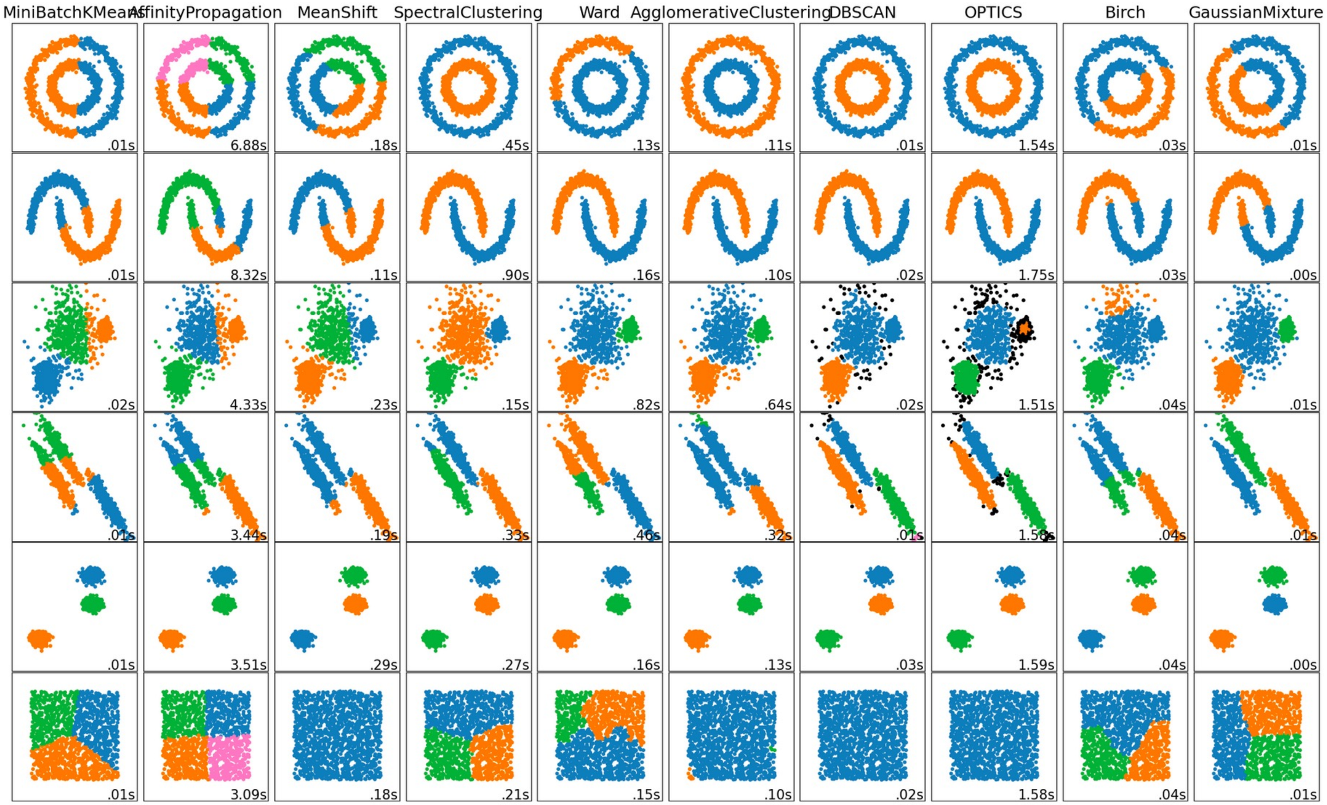
- Summarization:
  - Preprocessing for regression, PCA, classification, and association analysis
- Compression:
  - Image processing: vector quantization
- Finding K-nearest Neighbors
  - Localizing search to one or a small number of clusters
- Outlier detection
  - Outliers are often viewed as those “far away” from any cluster

**Content**: Jiawei Han, Micheline Kamber and Jian Pei
Data Mining: Concepts and Techniques, 3rd ed.

# Considerations for a Clustering Algorithm

- Need a distance measure for ***far*** and ***near***
- Be able to explain what a cluster means
- Handle different types of attributes: numeric, categorical (nominal, ordinal), binary
- Detect different shapes of clusters
- Handle noisy data
- Scale
  - Size
  - Dimensions

# Snapshot of Clustering Methods



A comparison of the clustering algorithms in scikit-learn

# Major Clustering Approaches (I)

Partitioning approach:

- Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors
- Typical methods: **k-means**, k-medoids, CLARANS

Hierarchical approach:

- Create a hierarchical decomposition of the set of data (or objects) using some criterion
- Typical methods: Diana, Agnes, **BIRCH**, CAMELEON

Density-based approach:

- Based on connectivity and density functions
- Typical methods: **DBSACN**, OPTICS, DenClue

Grid-based approach:

- based on a multiple-level granularity structure
- Typical methods: STING, WaveCluster, CLIQUE

**Content**: Jiawei Han, Micheline Kamber and Jian Pei
Data Mining: Concepts and Techniques, 3rd ed.

# Major Clustering Approaches (II)

Model-based:

- A model is hypothesized for each of the clusters and tries to find the best fit of that model to each other
- Typical methods: **EM**, SOM, COBWEB

Frequent pattern-based:

- Based on the analysis of frequent patterns
- Typical methods: p-Cluster

User-guided or constraint-based:

- Clustering by considering user-specified or application-specific constraints
- Typical methods: COD (obstacles), constrained clustering

Link-based clustering:

- Objects are often linked together in various ways
- Massive links can be used to cluster objects: **SimRank**, LinkClus

**Content**: Jiawei Han, Micheline Kamber and Jian Pei Data Mining: Concepts and Techniques, 3rd ed.

**Content**: Jiawei Han, Micheline Kamber and Jian Pei
Data Mining: Concepts and Techniques, 3rd ed.

# Partitioning Algorithms: Basic Concept

Partitioning method: Partitioning a database ***D*** of ***n*** objects into a set of ***k*** clusters, such that the sum of squared distances is minimized (where $c_i$ is the centroid or medoid of cluster $C_i$)

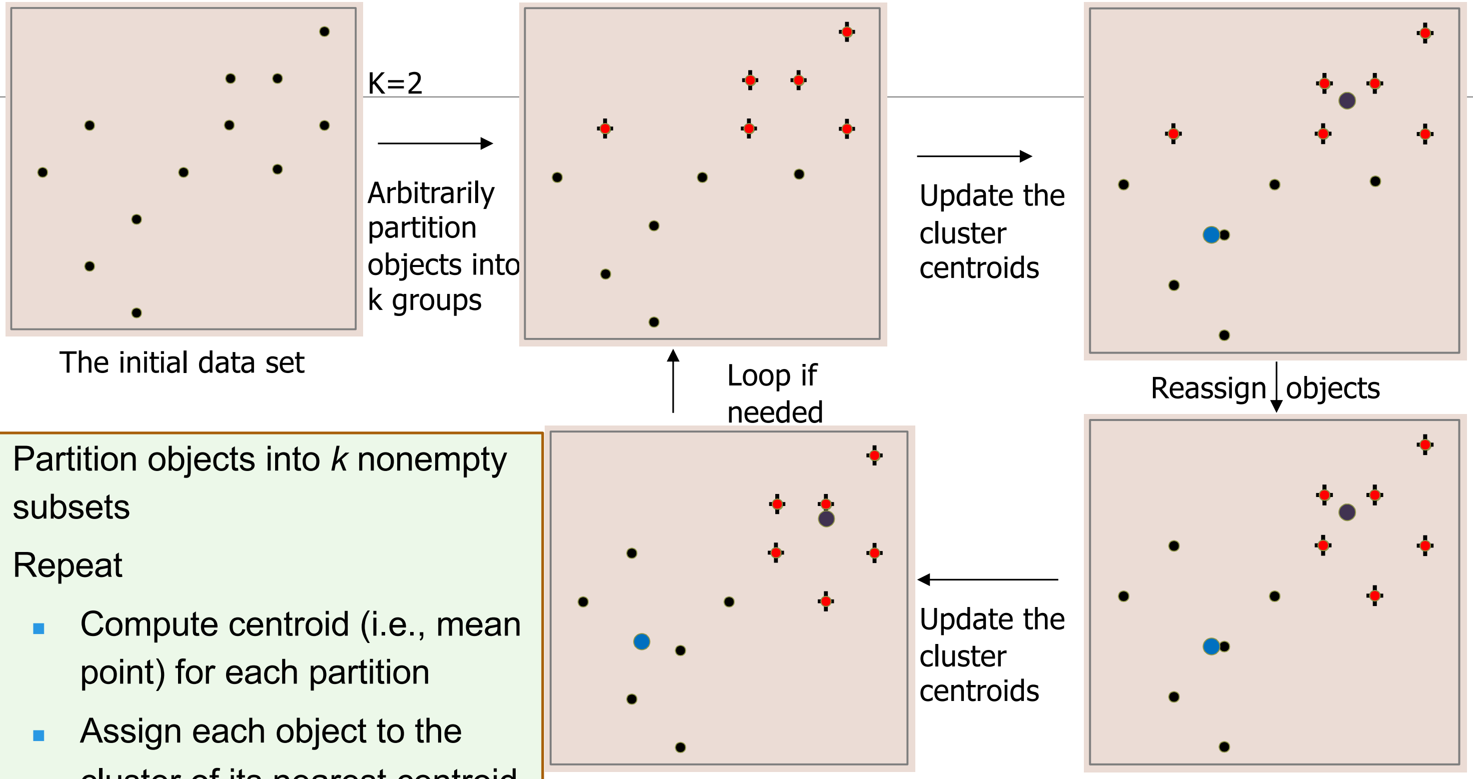
$$E = \Sigma_{i=1}^{k} \Sigma_{p \in C_i} (p - c_i)^2$$

Given *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion

- Global optimal: exhaustively enumerate all partitions
- Heuristic methods: *k-means* and *k-medoids* algorithms
- *k-means* (MacQueen'67, Lloyd'57/'82): Each cluster is represented by the center of the cluster
- *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Each cluster is represented by one of the objects in the cluster

**Content**: Jiawei Han, Micheline Kamber and Jian Pei
Data Mining: Concepts and Techniques, 3rd ed.

# An Example of *K-Means* Clustering



The initial data set

K=2

Arbitrarily partition objects into k groups

Update the cluster centroids

Reassign objects

Update the cluster centroids

Loop if needed

- Partition objects into *k* nonempty subsets
- Repeat
  - Compute centroid (i.e., mean point) for each partition
  - Assign each object to the cluster of its nearest centroid
- Until no change

**Content**: Jiawei Han, Micheline Kamber and Jian Pei Data Mining: Concepts and Techniques, 3rd ed.

# Comments on the *K-Means* Method

- **Strength**: *Efficient*: *O*(*tkn*), where *n* is # objects, *k* is # clusters, and *t* is # iterations. Normally, *k*, *t* << *n*.
  - Comparing: PAM: $O(k(n-k)^2)$, CLARA: $O(ks^2 + k(n-k))$
- **Comment**: Often terminates at a *local optimal*.
- **Weakness**
  - Applicable only to objects in a continuous n-dimensional space
    - Using the k-modes method for categorical data
    - In comparison, k-medoids can be applied to a wide range of data
  - Need to specify *k,* the *number* of clusters, in advance (there are ways to automatically determine the best k (see Hastie et al., 2009)
  - Sensitive to noisy data and *outliers*
  - Not suitable to discover clusters with *non-convex shapes*

# Exercise: Weka

- Use K-means on weather.arff
- Vary k

**Content**: Jiawei Han, Micheline Kamber and Jian Pei
Data Mining: Concepts and Techniques, 3rd ed.

# Distance Metrics – Numeric Variables

- Numeric quantity
  - Interval-scaled variables: continuous measurements of a roughly linear scale.
- Standardize with mean absolute deviation
  - $s_f = (1 / n) * (|x_{1f} – m_f| + … + |x_{1f} – m_f|)$
    - $s_{nf}$ and $m_f$ are measurements and mean, respectively
  - $z_{if} = (x_{if} – m_f) / s_f$

**Examples**: weight, height, latitude, longitude, temperature

- Distances for numbers
  - Euclidean: d(i,j) = square root ( $|x_{i1} – x_{j1}|^2 + … + |x_{ip} – x_{jp}|^2$ ) , for p-dimensional data
  - Manhattan: d(i,j) = $|x_{i1} – x_{j1}| + … + |x_{ip} – x_{jp}|$ , for p-dimensional data
  - Minlowski: 1/q root ( $|x_{i1} – x_{j1}|^q + … + |x_{ip} – x_{jp}|^q$ ) , for p-dimensional data

**Content**: Jiawei Han, Micheline Kamber and Jian Pei
Data Mining: Concepts and Techniques, 3rd ed.

# Distance Metrics – Binary Variables

| | **Object J** | | | |
|---|---|---|---|---|
| | | 1 | 0 | Sum |
| Object I | 1 | q | r | q+r |
| | 0 | s | t | s+t |
| | Sum | q+s | r+t | q+r+s+t |

*Contingency table for binary variables*

- Notation
  - q: number of binary variables that equal 1 for both objects I and J
- Distance between objects by matching
- d(I, J) = (r + s) / (q + r + s + t)

**Examples:**
Smoker/ non-smoker,
electric v/s non-electric car

**Content**: Jiawei Han, Micheline Kamber and Jian Pei
Data Mining: Concepts and Techniques, 3rd ed.

# Distance Metrics – Nominal Variables

- Notation
  - m: number of matches in values of nominal variables between objects I and J
  - M: total number of variables

- Distance between objects defined by matching
- d(I, J) = (p - m) / (p)

**Examples:**
map-color - red, yellow, green, pink, blue

**Content**: Jiawei Han, Micheline Kamber and Jian Pei Data Mining: Concepts and Techniques, 3rd ed.

# Distance Metrics – Ordinal Variables

- Conversion and notation
  - $z_{if} = (r_{if} - 1) / (M_{if} - 1)$
  - variable f of i-th object has 1..$M_f$ states in that order

- Now reuse distances for numbers
  - Euclidean: d(i,j) = square root ( $|x_{i1} - x_{j1}|^2 + \dots + |x_{ip} - x_{jp}|^2$ ) , for p-dimensional data
  - Manhattan: d(i,j) = $|x_{i1} - x_{j1}| + \dots + |x_{ip} - x_{jp}|$ , for p-dimensional data
  - Minlowski: 1/q root ( $|x_{i1} - x_{j1}|^q + \dots + |x_{ip} - x_{jp}|^q$ ) , for p-dimensional data

**Examples:**
professor ranks – assistant, associate, full
Medals – bronze, silver, gold
Military - …

**Content**: Jiawei Han, Micheline Kamber and Jian Pei
Data Mining: Concepts and Techniques, 3rd ed.

# Distance for Mixed Variable Types

- Keep separate and perform cluster analysis separately
  - Impractical
- Combine them into one scale between 0 to 1
- d(i,j) = $\frac{\sum_{f=1}^{p} \delta_{ij}{}^{(f)} d_{ij}{}^{(f)}}{\sum_{f=1}^{p} \delta_{ij}{}^{(f)}}$
  - Where $\delta_{ij}{}^{(f)}$ is 0 if $x_{if}$ or $x_{jf}$ are missing, otherwise 1
  - $d_{ij}{}^{(f)}$ is distance between i and j for feature f and type
- There can be a weighted variation too

# Exercise - 1

- Consider clustering of days
  - What are some possible groups?
  - What features make sense?
  - What distances make sense?

# Exercise - 2

Consider clustering of documents, like resumes, into groups

- What are some possible groups?
  - By areas: Technology, finance, services, manufacturing, …
- What features make sense?
  - Syntactic: Words, sentiments, …
  - Semantic: qualification, experience, …
- What distances make sense?

# Clustering Quality

# Case A: Ground Truth is Known

- **homogeneity**: each cluster contains only members of a single class.
- **completeness**: all members of a given class are assigned to the same cluster

- Example:
  - true labels = [0, 0, 0, 1, 1, 1]
  - P1: Predicted labels = [0, 0, 1, 1, 2, 2]
  - P2: Predicted labels = [0, 0, 0, 2, 2, 2]
- In example P1, informally
  - Homogeneity - (Predicted) 1 has members of 0 and 1
  - Completeness – (Actual) 0 is assigned to 0 and 1, (Actual) 1 is assigned 1 and 2

**Note**: P2 is homogeneous and complete

**Content acknowledgement**: Sci-kit: https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation

# Case A: Ground Truth is Known

- **homogeneity**: each cluster contains only members of a single class.
- **completeness**: all members of a given class are assigned to the same cluster

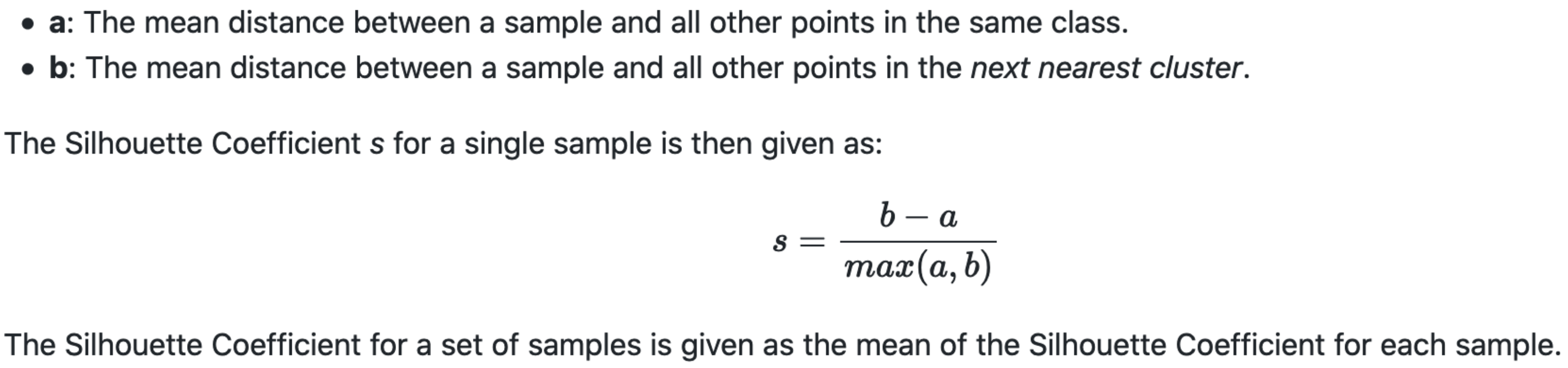
- **v-measure**

$$v = \frac{(1+\beta) \times \mathbf{homogeneity} \times \mathbf{completeness}}{(\beta \times \mathbf{homogeneity} + \mathbf{completeness})}$$

**Content acknowledgement**: Sci-kit: https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation

# Case B: Ground Truth is Unknown

Silhouette Coefficient



- **a**: The mean distance between a sample and all other points in the same class.
- **b**: The mean distance between a sample and all other points in the *next nearest cluster*.

The Silhouette Coefficient *s* for a single sample is then given as:

$$s = \frac{b - a}{max(a, b)}$$

The Silhouette Coefficient for a set of samples is given as the mean of the Silhouette Coefficient for each sample.

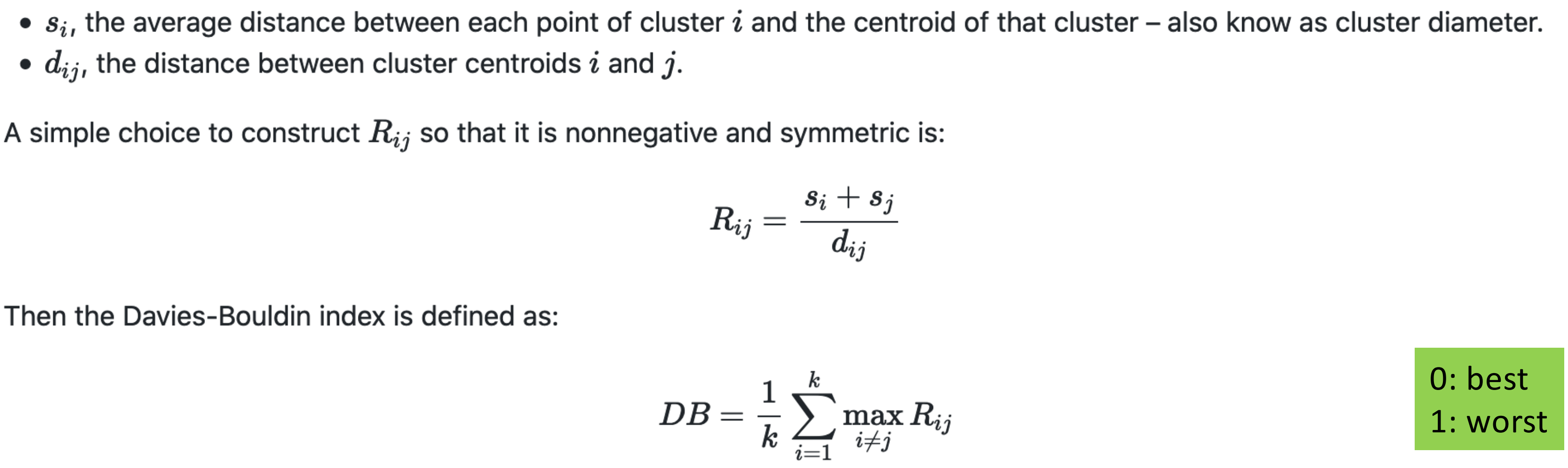
-1: incorrect clustering
+1: highly dense clustering.
Scores around zero indicate overlapping clusters.

**Question**: can you calculate when all data is in one cluster?

**Content acknowledgement**: Sci-kit: https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation

# Case B: Ground Truth is Unknown

Davies-Bouldin Index



- $s_i$, the average distance between each point of cluster $i$ and the centroid of that cluster – also know as cluster diameter.
- $d_{ij}$, the distance between cluster centroids $i$ and $j$.

A simple choice to construct $R_{ij}$ so that it is nonnegative and symmetric is:

$$R_{ij} = \frac{s_i + s_j}{d_{ij}}$$

Then the Davies-Bouldin index is defined as:

$$DB = \frac{1}{k} \sum_{i=1}^{k} \max_{i \neq j} R_{ij}$$

0: best
1: worst

Limitation: Needs euclidean distances

**Content acknowledgement**: Sci-kit: https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation

**Content**: Jiawei Han, Micheline Kamber and Jian Pei
Data Mining: Concepts and Techniques, 3rd ed.

# Measuring Clustering Quality

- Two methods: extrinsic vs. intrinsic
- Extrinsic: supervised, i.e., the ground truth is available
  - Compare a clustering against the ground truth using certain clustering quality measure
  - Ex. Recall - precision and recall metrics in classification
- Intrinsic: unsupervised, i.e., the ground truth is unavailable
  - Evaluate the goodness of a clustering by considering how well the clusters are separated, and how compact the clusters are
  - Ex. Silhouette coefficient

**Content**: Jiawei Han, Micheline Kamber and Jian Pei Data Mining: Concepts and Techniques, 3rd ed.

# Measuring Clustering Quality: Extrinsic Methods

- Clustering quality measure: $Q(C, C_g)$, for a clustering $C$ given the ground truth $C_g$.
- $Q$ is good if it satisfies the following **4** essential criteria
  - Cluster homogeneity: the purer, the better
  - Cluster completeness: should assign objects belong to the same category in the ground truth to the same cluster
  - Rag bag: putting a heterogeneous object into a pure cluster should be penalized more than putting it into a *rag bag* (i.e., “miscellaneous” or “other” category)
  - Small cluster preservation: splitting a small category into pieces is more harmful than splitting a large category into pieces

**Content**: Jiawei Han, Micheline Kamber and Jian Pei
Data Mining: Concepts and Techniques, 3rd ed.

# Summary

- Cluster analysis groups objects based on their similarity  and has wide applications
- Measure of similarity can be computed for various types of data
- Clustering algorithms can be categorized into partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods
- K-means and K-medoids algorithms are popular partitioning-based clustering algorithms
- Birch and Chameleon are interesting hierarchical clustering algorithms, and there are also probabilistic hierarchical clustering algorithms
- DBSCAN, OPTICS, and DENCLU are interesting density-based algorithms
- STING and CLIQUE are grid-based methods, where CLIQUE is also a subspace clustering algorithm
- Quality of clustering results can be evaluated in various ways

# Code Examples

- Clustering quality
  - https://github.com/biplav-s/course-d2d-ai/blob/main/sample-code/l10-11-unsupervised-ml/clustering-quality-measures.ipynb
- Clustering methods
  - https://github.com/biplav-s/course-d2d-ai/blob/main/sample-code/l10-11-unsupervised-ml/Cluster-exploration-syntheticdata.ipynb

# Exercise: Weka

- Pick a data-set with at least 5 attributes
- Cluster with 2 methods
- Review cluster quality

# Explaining Clusters

- How to describe them ?
  - Centroid
  - Exemplars

- What name to give them ?
  - Using features of the members
  - Algorithm may produce (Concept Clustering)

- Explanations can be based on domain specific rules

# Lecture 7: Concluding Comments

- Understood Clustering problem
- Understood k-means
- A range of clustering methods
- Measuring cluster quality
- Explaining clusters
- Working with Weka, scikit and python code samples

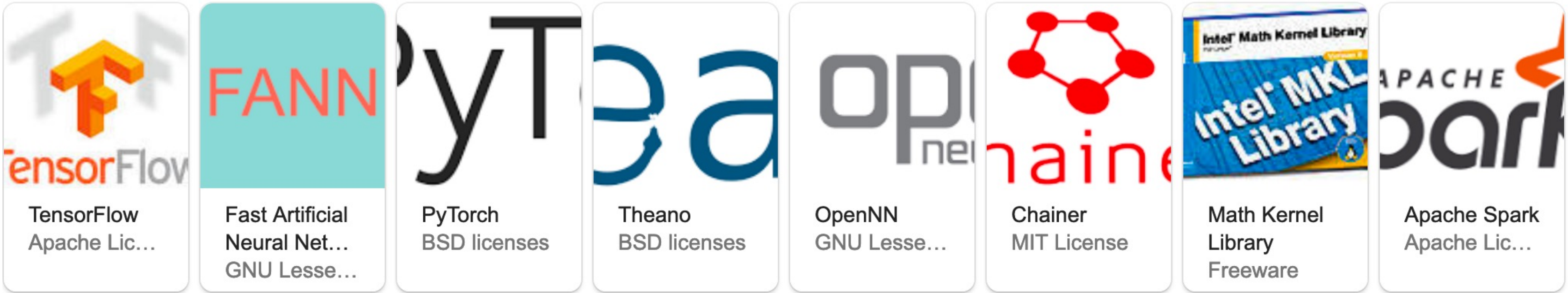
# Lecture 8: NN, Deep Learning

# Machine Learning – Insights from Data

- Descriptive analysis
  - Describe a past phenomenon
  - **Methods**: classification (feedback from label), clustering, dimensionality reduction, anomaly detection, neural methods, reinforcement learning (feedback from hint/ reward)
- Predictive analysis
  - Predict about a new situation
  - **Methods**: time-series, neural networks
- Prescriptive analysis
  - What an agent should do
  - **Methods**: simulation, reinforcement learning, reasoning

- New areas
  - Counterfactual analysis
  - Causal Inferencing
  - Scenario planning
  - Representation learning

# Neural Network Methods



TensorFlow
Apache Lic…

Fast Artificial Neural Net…
GNU Lesse…

PyTorch
BSD licenses

Theano
BSD licenses

OpenNN
GNU Lesse…

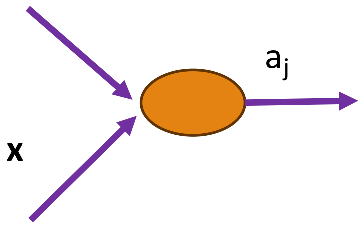
Chainer
MIT License

Math Kernel Library
Freeware

Apache Spark
Apache Lic…

# Node (Unit) of a NN

- Notations and meanings
  - $a_j$: output of a unit j
  - $w_{i,j}$: weight of link from unit i to unit j
  - $a_{j} = g_j\ (\Sigma\ w_{i,j}\, a_i\ )$, where $g_j$ is a nonlinear activation function
- $a_{j} = g_j\ (\mathbf{w}^T\ \mathbf{x})$, where **w** is vector of weights leading into unit j and **x** is the inputs to unit j



# Major Types of NNs

## Spiking NN

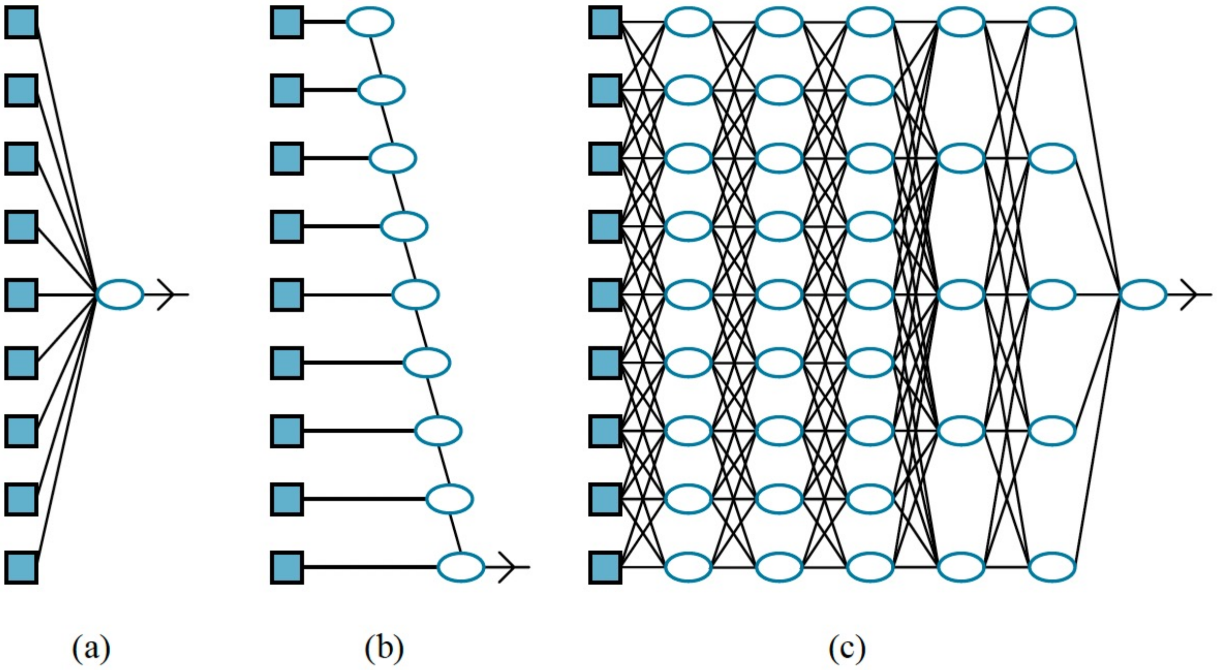
- The idea is that neurons do not transmit information at each propagation cycle only when a membrane potential—an intrinsic quality of the neuron related to its membrane electrical charge—reaches a specific value, called the threshold.
- When the membrane potential reaches the threshold, the neuron fires, and generates a signal that travels to other neurons which, in turn, increase or decrease their potentials in response to this signal.
- A neuron model that fires at the moment of threshold crossing is also called a spiking neuron model.[3]

## Artificial NN

- Artificial Neurons are highly abstracted out
  - no modeling of action potentials or spikes
  - no accounting of temporality or spatiality and they are stateless
- Artificial Neurons are simple non-linear activation functions
  - e.g. ReLU, Sigmoid, etc.
- Real-valued output of artificial neurons is analogous to the firing rate of spiking neurons

**Credit**: https://en.wikipedia.org/wiki/Spiking_neural_network

# Model Depth and Learning Ability



(a) A shallow model, such as linear regression, has short computation paths between inputs and output. (b) A decision list network has some long paths for some possible input values, but most paths are short. (c) A deep learning network has longer computation paths, allowing each variable to interact with all the others.

Adapted from:
Russell & Norvig, AI: A Modern Approach

# NN Background

- Neural networks were first proposed in 1944 by Warren McCullough and Walter Pitts,
- The first trainable neural network, the Perceptron, was demonstrated by the Cornell University psychologist Frank Rosenblatt in 1957.

Details and Credits:

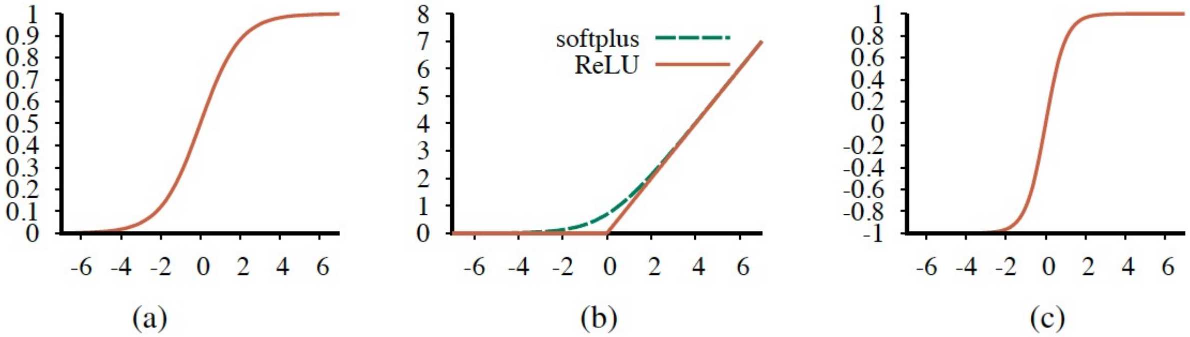
a) https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414

b) Sarker I. H. (2021). Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN computer science*, *2*(6), 420. https://doi.org/10.1007/s42979-021-00815-1

# Popular Activation Functions

- Logistics or sigmoid function: $\sigma(x) = 1/(1 + e^{-x})$
- ReLU (rectified linear unit): max (0, x)
- Softplus function: $\log(1 + e^{x})$
  - Smooth version of ReLU
- $\tanh(x) = (e^{2x} - 1) / (e^{2x} + 1)$
  - Scaled and shifter version of sigmoid; $\tanh(x) = 2\sigma(2x) - 1$

a) the logistic or sigmoid function
b) the ReLU function and the softplus function
c) the tanh function.



Adapted from:
Russell & Norvig, AI: A Modern Approach

**Note**: All activation functions are non-linear

# Loss functions

- Mean squared error

$$MSE = \frac{1}{n} \sum_{j=1}^{n} \left[ f(X_{j\cdot}) - y_j \right]^2$$

- Categorical Cross Entropy

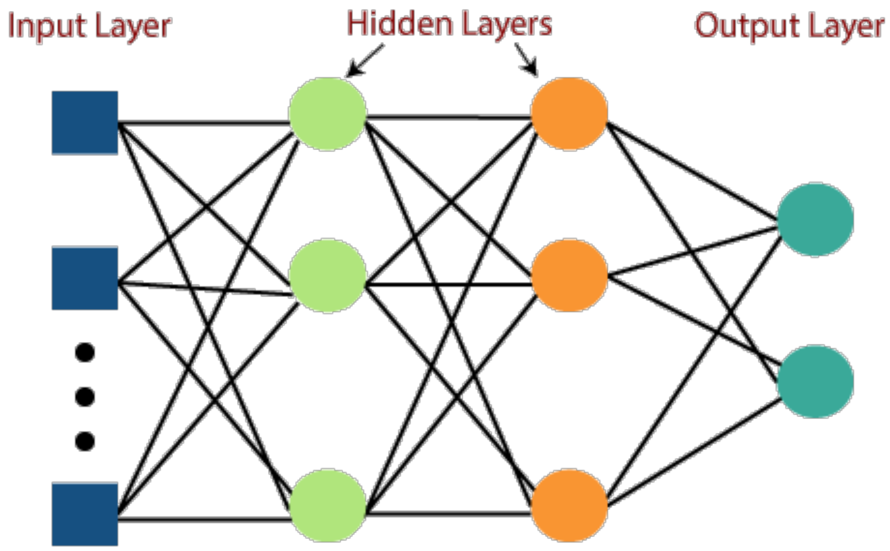
$$\text{Cost} = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} [y_{ij} \log(\hat{y}_{ij})]$$

k is classes,
y = actual value
**Ŷ =** prediction

- More loss functions:

https://www.analyticsvidhya.com/blog/2022/06/understanding-loss-function-in-deep-learning/

# NN – Multi Layer Perceptron



Content and Image Courtesy:
https://github.com/Thanasis1101/MLP-from-scratch

# (Stochastic) Gradient Descent

Gradient Descent

```
$\mathbf{w} \leftarrow$ any point in the parameter space
While not converged do:
    For each $w_i$ in $\mathbf{w}$ do:
        $w_i \leftarrow w_i - \alpha \; (\partial \;/\; \partial w_i \,) \; Loss\,(\mathbf{w})$
```

Calculate the gradient of the loss function with respect to the weights along the gradient direction to reduce the loss.

Stochastic Gradient Descent (SGD)

Randomly select a small number of training examples at each step

**Sources**:

- https://en.wikipedia.org/wiki/Stochastic_gradient_descent
- Russell & Norvig, AI: A Modern Approach, Chapter 19

# Logistic Regression in a Slide

Function estimate (linear)

W: weight, b: bias

$$f(X_j) = X_j W + b$$

Update Weight

$$W^* = W - \eta \frac{dL}{dW}$$

Error Term (mean squared error)

$$MSE = \frac{1}{n} \sum_{j=1}^{n} \left[ f(X_{j\cdot}) - y_j \right]^2$$

**Common Code Pattern**

```
y = tf.matmul(x, W) + b
loss = tf.reduce_mean(tf.square(y - y_label))
```

# NN Concepts

- **Epoch**: The number of times the learning algorithm will iterate over the entire dataset
- **Batch**: how many samples are processed before updating the model's internal parameters.
  - **Batch Gradient Descent**: Batch Size = Size of Training Set
  - **Stochastic Gradient Descent**: Batch Size = 1
  - **Mini-Batch Gradient Descent**: 1 < Batch Size < Size of Training Set

**Credit**: https://rentry.org/llm-training

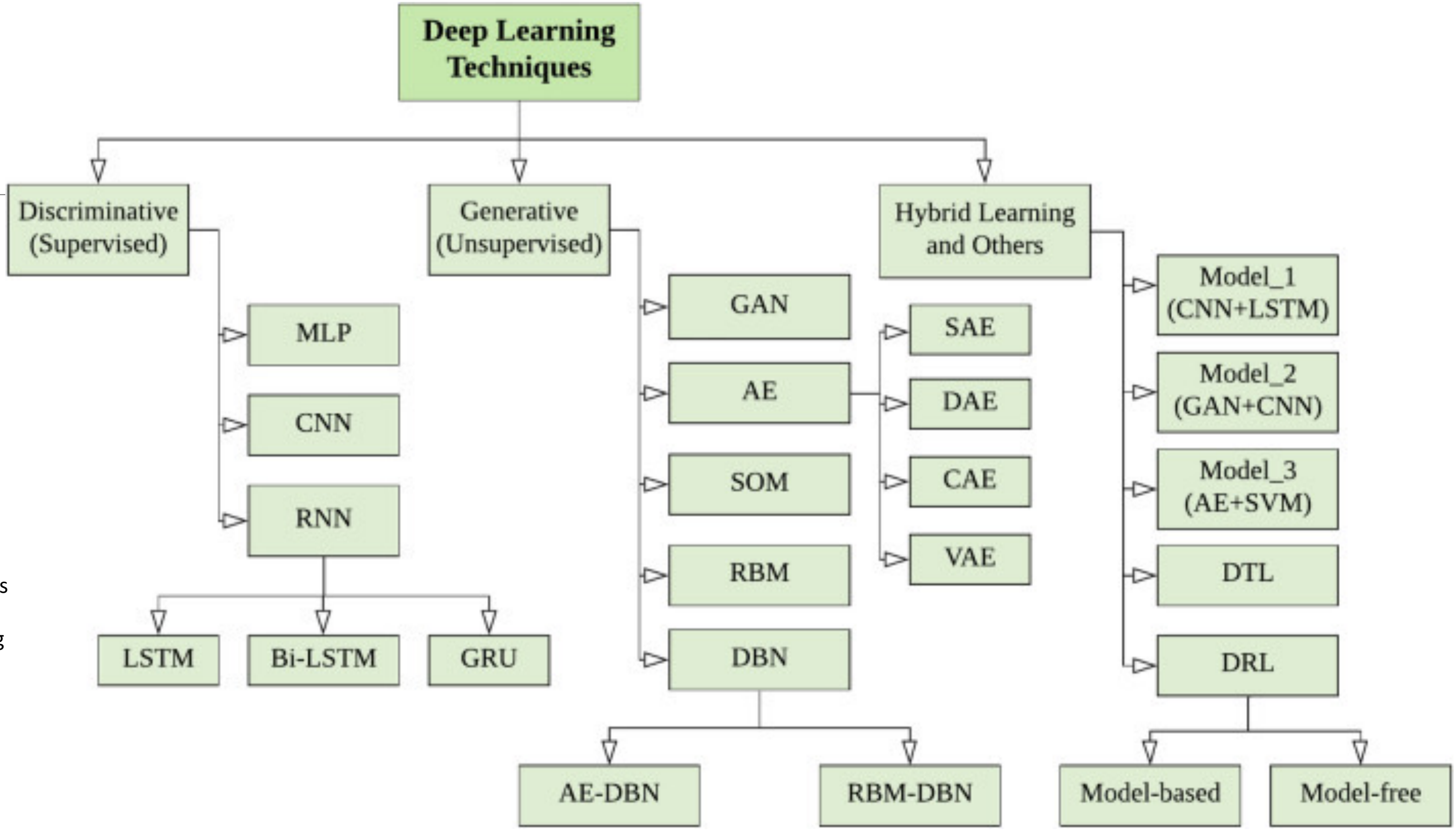
# Universal Approximation Theorem

- A network with just two layers of computation units, first nonlinear, and the second linear, can approximate any continuous function to an arbitrary degree of accuracy.
- **Why**: a sufficiently large network can implement a lookup table for continuous functions
  - Nonlinear layer is the key

**Sources**:

- https://en.wikipedia.org/wiki/Universal_approximation_theorem
- Russell & Norvig, AI: A Modern Approach, Chapter 21

# Deep NN Types



A taxonomy of DL techniques, broadly divided into three major categories (i) deep networks for supervised or discriminative learning, (ii) deep networks for unsupervised or generative learning, and (ii) deep networks for hybrid learning and relevant others

Details and Credits:

a) https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414

b) Sarker I. H. (2021). Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN computer science*, *2*(6), 420. https://doi.org/10.1007/s42979-021-00815-1

# Datasets

- In keras, https://keras.io/api/datasets/
  - boston_housing
  - cifar10 module, cifar100, fashion_mnist, mnist
  - imdb module
  - reuters module

- In TF, https://www.tensorflow.org/datasets/catalog/overview#all_datasets

# Keras Walkthrough

- Package: https://keras.io/about/
- Example model:
  - Sequential: https://keras.io/guides/sequential_model/
- Many examples: classification, image, text, audio
  - https://keras.io/examples/
- Future Keras: https://keras.io/keras_core/
  - Keras Core - run Keras workflows on top of TensorFlow, JAX, and PyTorch; preview of Keras 3.0

# Code Examples With Keras and TF

**1. Classification – diabetes**

2. Try code
   - Play with hyper-parameters
   - Look at keras features used

Code location:
https://github.com/biplav-s/course-ai-tai-f23/tree/main/sample-code/Class19-To-21-DL

# Discussion

- Impact of network structure:
  - Nodes / layer:
  - Layers:
  - Inter-connection structure:
- Impact of hyper-parameters:
  - Epochs:
  - Batch size:

# Code Examples With Keras and TF

1. Classification – diabetes

**2. Prediction/ representation learning – autoencoder**

Code location:
https://github.com/biplav-s/course-ai-tai-f23/tree/main/sample-code/Class19-To-21-DL

# Discussion

- Impact of network structure:
  - Nodes / layer:
  - Layers:
  - Inter-connection structure:
- Impact of hyper-parameters:
  - Epochs:
  - Batch size:

# Code Examples With Keras and TF

1. Classification – diabetes

2. Prediction/ representation learning – autoencoder

**3. Classification – MNIST**

Code location:
https://github.com/biplav-s/course-ai-tai-f23/tree/main/sample-code/Class19-To-21-DL

# Discussion

- Impact of network structure:
  - Nodes / layer:
  - Layers:
  - Inter-connection structure:
- Impact of hyper-parameters:
  - Epochs:
  - Batch size:

# Keras and TensorFlow

- By Example:
  - https://github.com/biplav-s/course-nl-f22/blob/main/sample-code/l11-nn-dl/Basic%20TensorFlow%20and%20Keras.ipynb
  - TensorFlow's NMIST tutorial
  - https://www.tensorflow.org/tutorials/quickstart/beginner

- More examples
  - Number Addition by sequence learning: https://keras.io/examples/nlp/addition_rnn/
  - AutoEncoder: https://machinelearningmastery.com/lstm-autoencoders/

# NN/ MLP

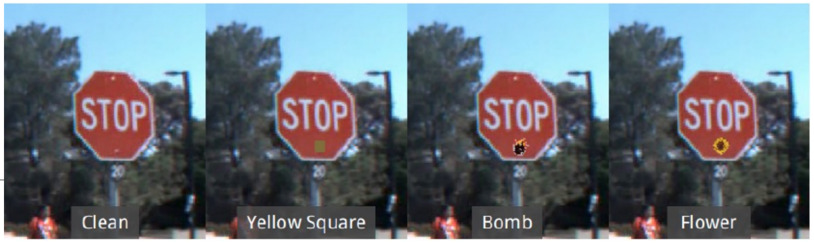
- Code examples:
  - https://github.com/biplav-s/course-d2d-ai/blob/main/sample-code/l6-l7-l8-supervised-ml/Supervised-NaiveBayes-GradientBoost-NN-Classification.ipynb
- Scikit Library:
  - MLP: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

# Consideration: Which NN/DL Tool to Use

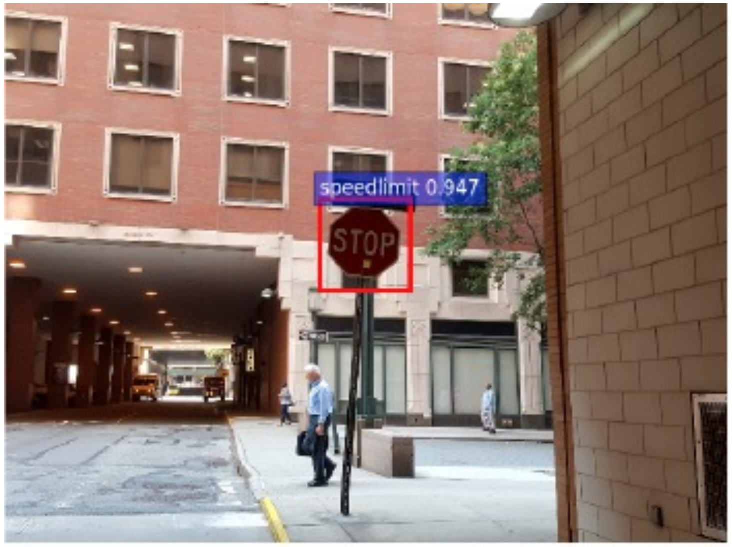
- See:
  - https://www.simplilearn.com/keras-vs-tensorflow-vs-pytorch-article
  - In theory, keras supports all major ones
    - Pytorch used in academic research more
    - TF used in production systems

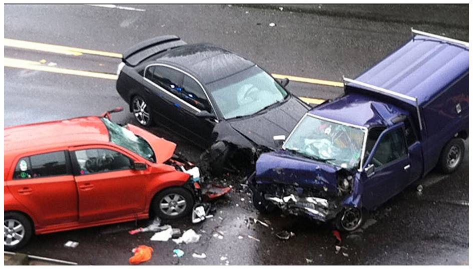
# Trust: Adversarial Attacks

Example (Gu et al. 2017)



- **ML Application**: Detect and classify street signs in images
- **Poisoning method**: Insert images where a special sticker is added to stop signs and the label changed to speed limit
- **Backdoor**: Adversaries ensure that any stop sign is misclassified simply by placing a sticker on it





# Trust: Adversarial Attacks

- Cat and mouse on attacks and defenses
  - Example code: https://github.com/Trusted-AI/adversarial-robustness-toolbox/blob/main/notebooks/adversarial_training_mnist.ipynb
- Tools
  - Adversarial Robustness Toolbox (ART) - Python library for Machine Learning Security, https://github.com/Trusted-AI/adversarial-robustness-toolbox

# Trust Issues with NN

- Robustness: can the model give the results in the presence of (input) perturbation? Noise?
- Computation/ footprint: why does the learning take so much compute resources?
- Data: is the data representative? How was the data obtained?
- Explainability: why does the model work?
- Fairness: Is the output fair to user groups?

# Resources and Books

- Understanding Deep Learning, https://udlbook.github.io/udlbook/
- Deep Learning, Ian Goodfellow, Yoshua Bengio and Aaron Courville, https://www.deeplearningbook.org/
- AI – A Modern Approach, Russell & Norvig, https://aima.cs.berkeley.edu/
- Websites of libraries – Keras.

# Lecture 8: Concluding Comments

- We talked about
  - Neural Networks
  - Deep Learning
  - Trust Issues
  - Adversarial Attacks

# Week 4: Concluding Comments

- We talked about
  - Unsupervised ML
  - NN and DL
  - Quiz 1

- Week 1: Introduction, Aim: Chatbot / Intelligence Agent
- Weeks 2-3: Data: Formats, Representation and the Trust Problem
- Week 3: Machine Learning – Supervised (Classification)
- Week 4: Machine Learning - Unsupervised (Clustering) –
- Topic 5: Learning neural network, deep learning, Adversarial attacks
- Week 6: Large Language Models – Representation and Usage issues
- Weeks 7-8: Search, Heuristics - Decision Making
- Week 9: Constraints, Optimization – Decision Making
- Topic 10: Markov Decision Processes, Hidden Markov models - Decision making
- Topic 11-12: Planning, Reinforcement Learning – Sequential decision making
- Week 13: Trustworthy Decision Making: Explanation, AI testing
- Week 14: AI for Real World: Tools, Emerging Standards and Laws; Safe AI/ Chatbots

# Quiz 1: Take Home

- Due date: **Tuesday, Sep 16**

# Projects A: Start (4 weeks; 200 points)

- End date: **Thursday, Sep 18**

- See model AI Assignments: http://modelai.gettysburg.edu/
  - Choose a project, preferably within last 5 years (i.e., after 2020).
    - Enter its name in **“Student-InfoShared ..” sheet, column G**
  - Follow instructions and do it alone
  - Submit project outcome
    - Create a folder in your Github called **ProjectA**.
    - Create a file called “ProjectInfo.md” with your name, project chosen and URL/ other details.
    - Put deliverables, as per project description, inside the folder, and commit.
    - Timestamp will be used to confirm that Project-A is done on time

# About Week 5 – Lectures 9 and 10

# Week 5 – Lectures 9 and 10

- Large Language Models (LLMs) / Foundation Models
- Using LLMs; Trust Issues
- Quiz 1 will be due
- Project will be due

- Week 1: Introduction, Aim: Chatbot / Intelligence Agent
- Weeks 2: Data: Formats, Representation, ML Basics
- Week 3: Machine Learning – Supervised (Classification)
- Week 4: Machine Learning - Unsupervised (Clustering) –
- Topic 5: Learning neural network, deep learning, Adversarial attacks
- Week 6: Large Language Models – Representation and Usage issues
- Weeks 7-8: Search, Heuristics - Decision Making
- Week 9: Constraints, Optimization – Decision Making
- Topic 10: Markov Decision Processes, Hidden Markov models - Decision making
- Topic 11-12: Planning, Reinforcement Learning – Sequential decision making
- Week 13: Trustworthy Decision Making: Explanation, AI testing
- Week 14: AI for Real World: Tools, Emerging Standards and Laws; Safe AI/ Chatbots

**Note**: exact schedule changes slightly to accommodate for exams and holidays.