Summer 2023 Training Course:

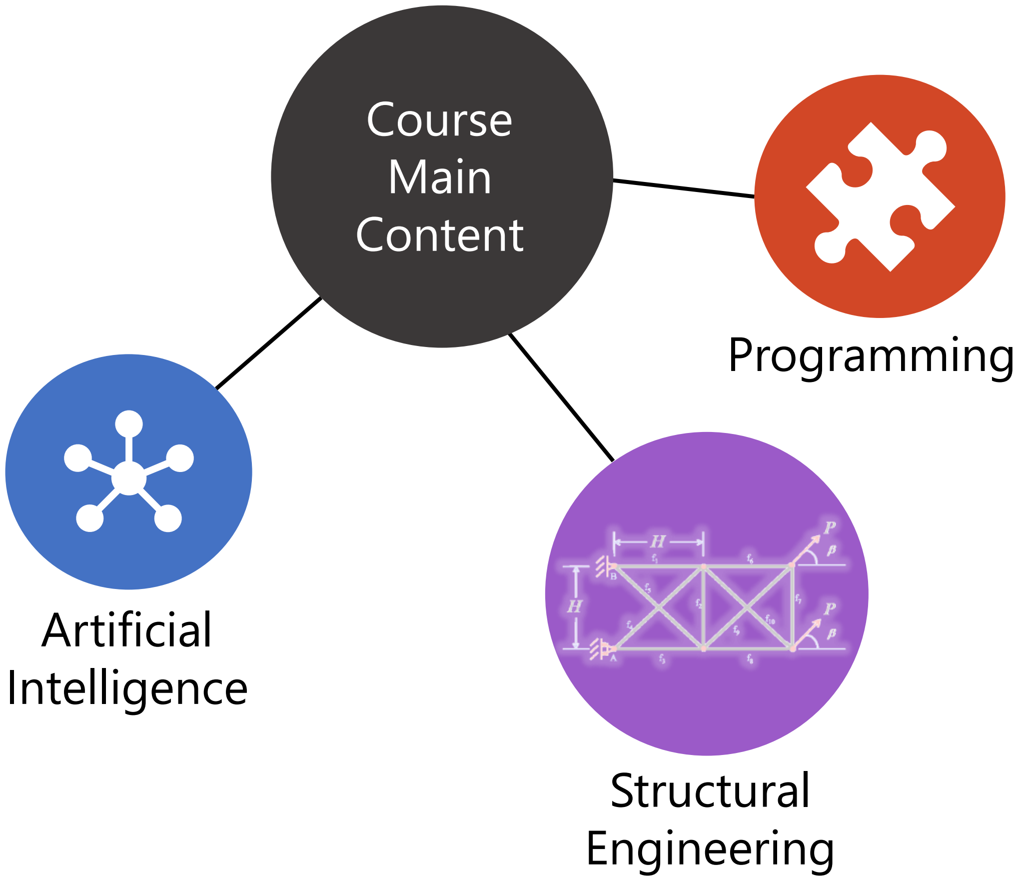
Artificial Intelligence Applications in Structural Engineering

Week 3: **Introduction to ML**

Ahmed A. Torky – Civil Engineering Department

Content

- Artificial Intelligence
- Learning Type
- Clustering Methods
    - K-Means
    - K Nearest Neighbor
    - Examples
- Dense Neural Networks
    - Example
- Convolutional Neural Networks
- Evolutionary Algorithms
    - Particle Swarm Algorithm
    - Genetic Algorithm
    - Examples



Artificial Intelligence?

Artificial Intelligence (AI)

The ability of a computer system to demonstrate and perform cognitive functions associated with the human mind, including perceiving, reasoning, learning, and problem solving.

Machine Learning (ML)

A branch of AI that uses algorithms to find patterns in datasets and subsequently make predictions about the future, enabling machine to learn without receiving explicit programming instruction. The three major types of ML include supervised learning, unsupervised learning, and reinforcement learning.

Deep Learning (DL)

A modern branch of ML that uses artificial neural networks to consume vast amounts of data (i.e. Big Data), often producing more accurate results than traditional ML approaches.

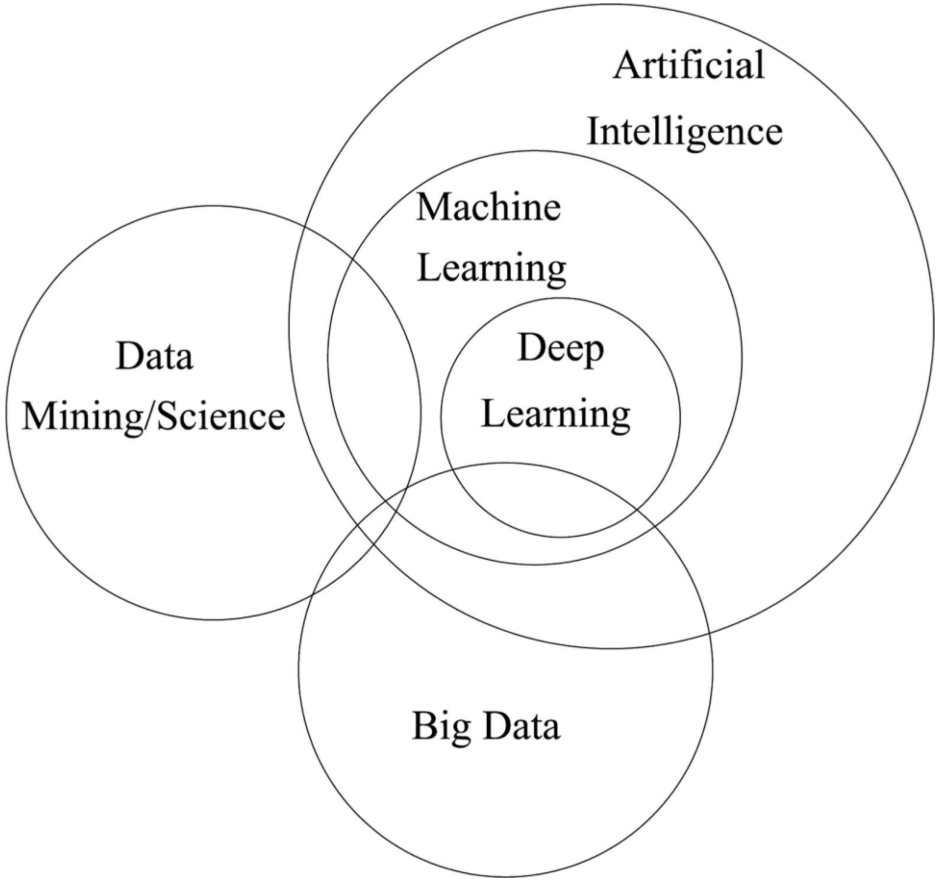
Types of Machine Learning

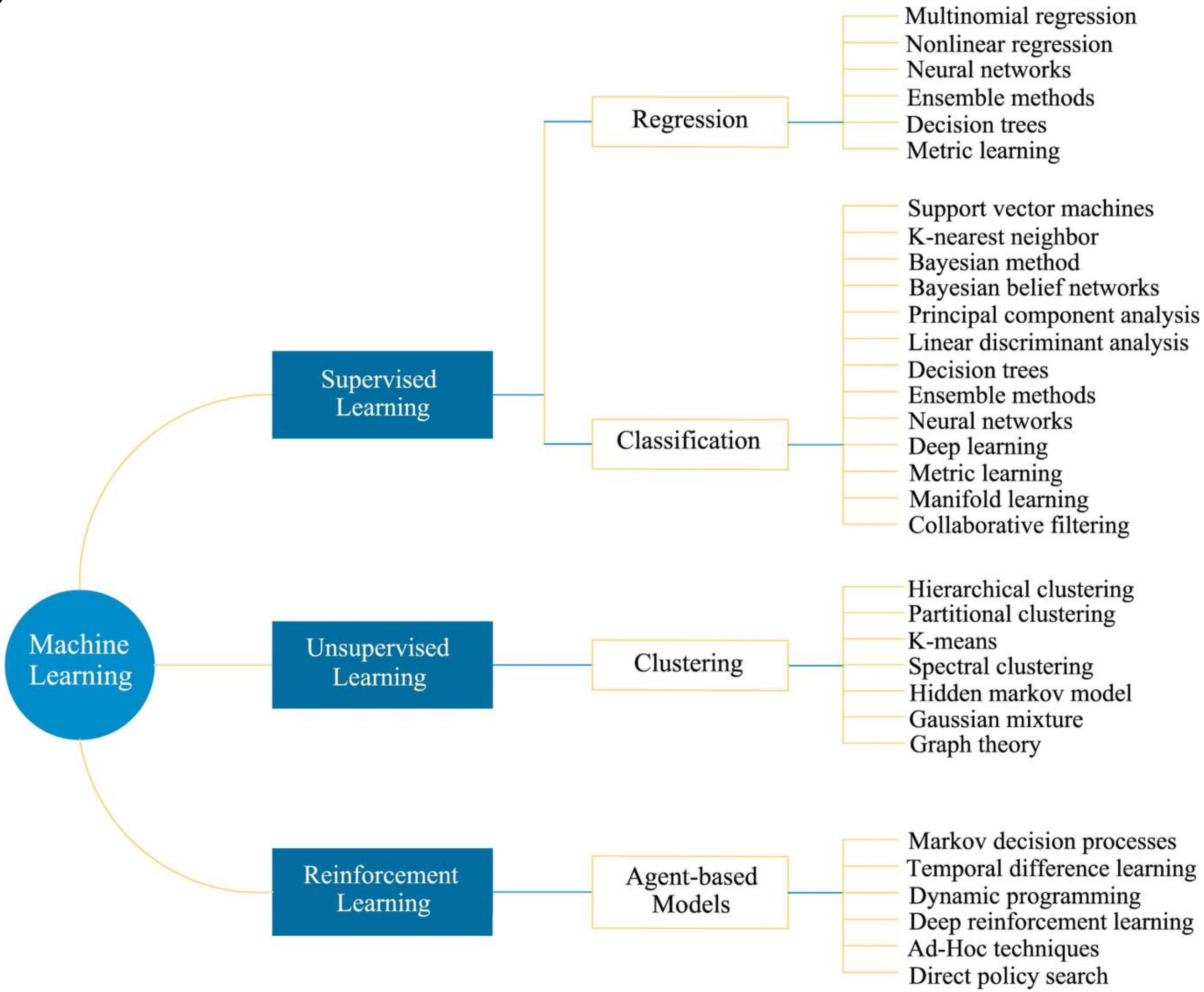
Supervised Learning
Uses training data and feedback from humans to detect patterns and predict next values.

Unsupervised Learning
Explores unlabeled input data without being provided an explicit output variable to detect patterns

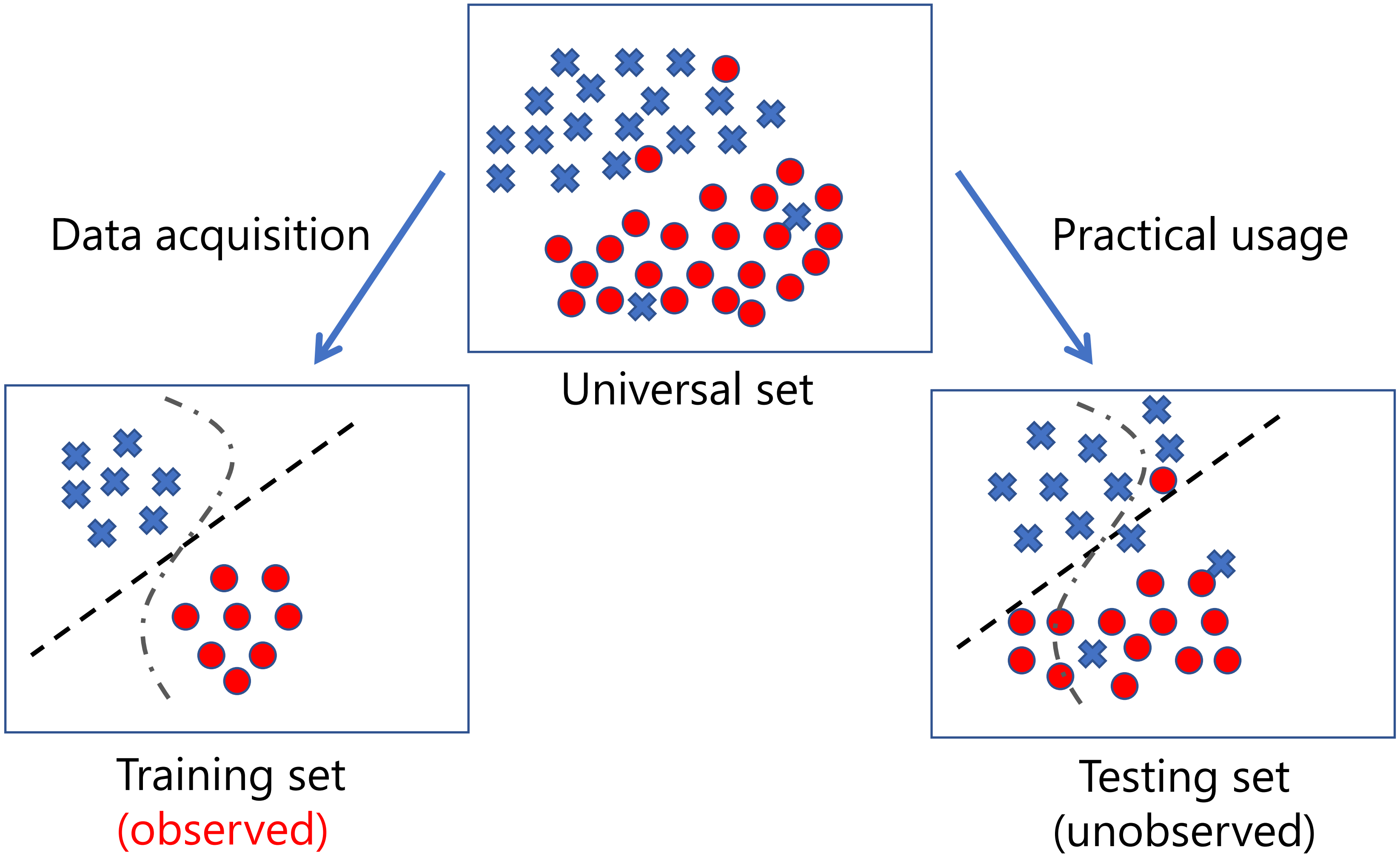
Reinforcement Learning
Uses training data and feedback from humans to detect patterns and predict next values.

AI Learning Types

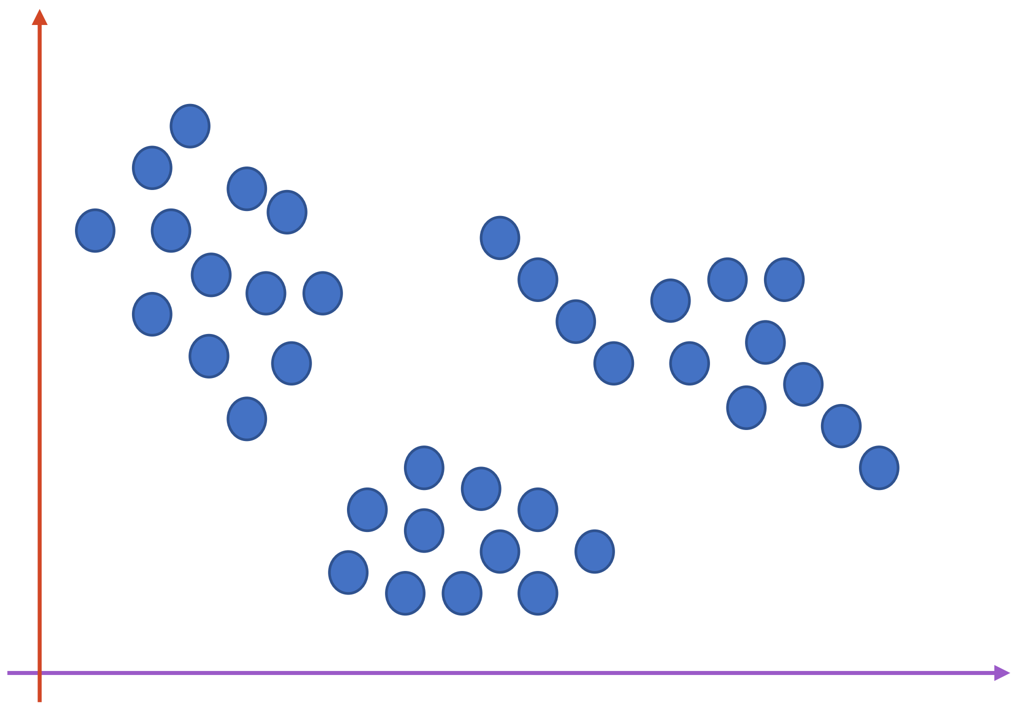




Data Form (Classification)

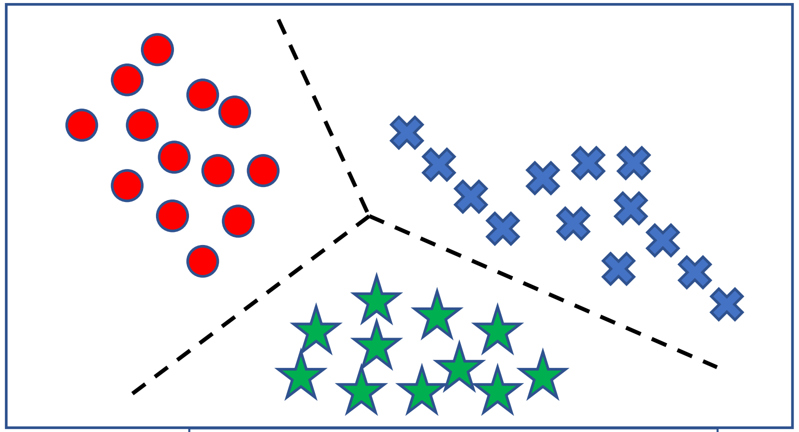


Problem?

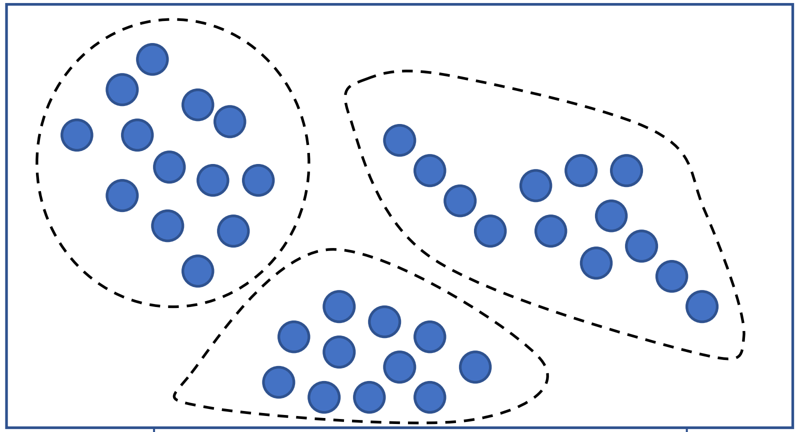


Data Classification is Required!
How could we separate our data into clusters?

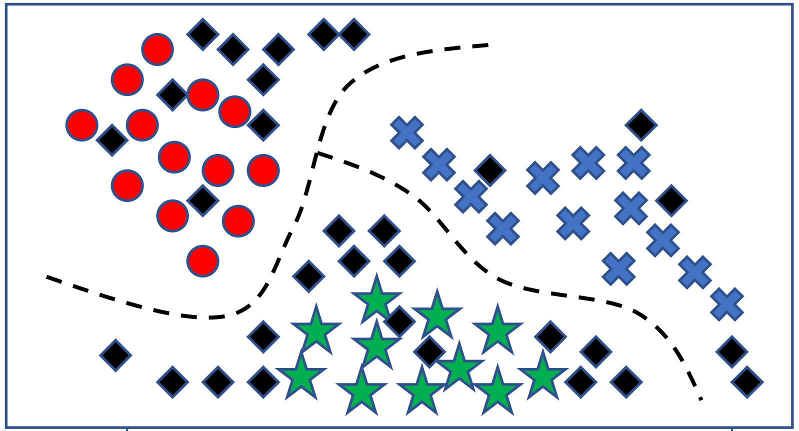
ML Algorithms



Supervised learning



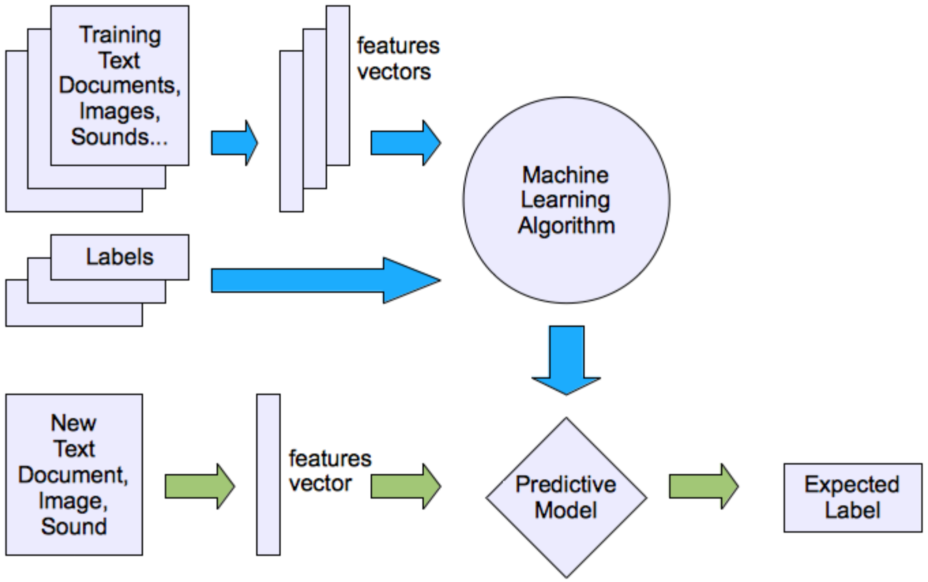
Unsupervised learning



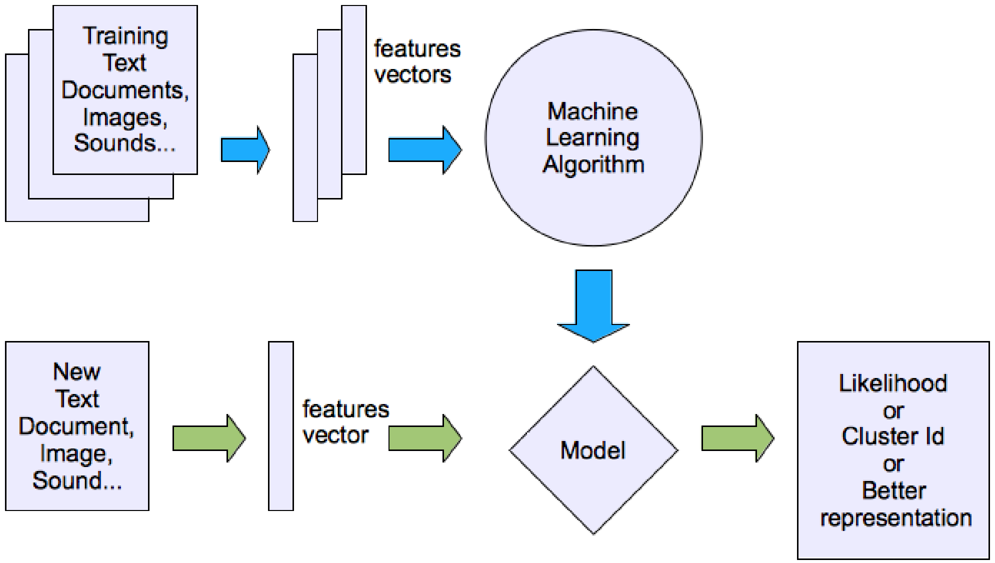
Semi-supervised learning

Supervised Vs Unsupervised

Supervised Learning



Un-Supervised Learning



Before Applying Machine Learning

1. Define your problem,
2. Collect your data,
3. Organize your data,
4. Explore your data and filter/select features (Preprocessing),
5. Decide on your Input X and Output Y,
6. Split your data to training, validation, and testing datasets.

Machine Learning Pipeline

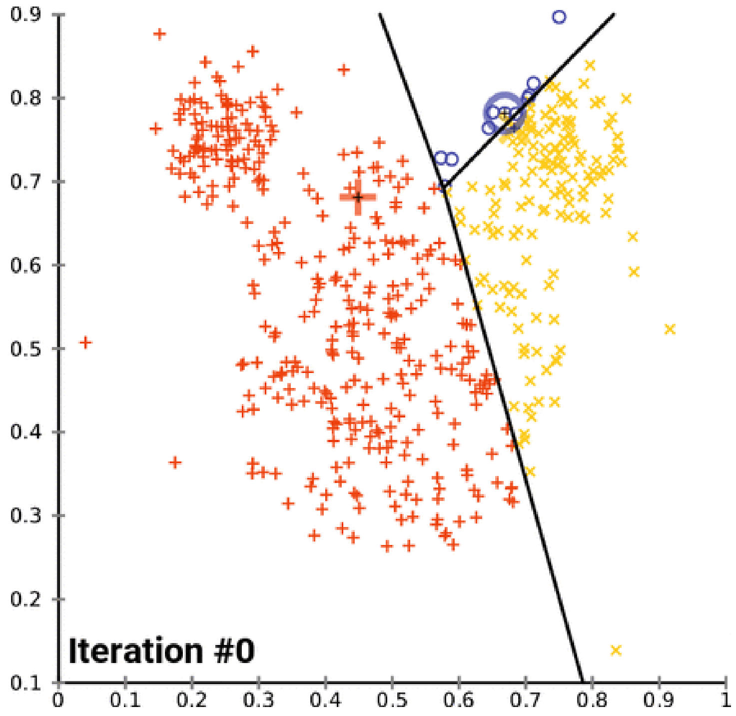
1. **Data Collection and handling** from previous slide.
2. **Model Selection**: Choose the machine learning algorithm that you want to use. This choice is based on the type of problem (classification, regression, clustering, etc.), the nature of the data, and the requirement of the problem.
3. **Training the Model**: Use your selected algorithm to build a model using your training data.
4. **Model Evaluation**: Evaluate the performance of your model using suitable metrics. This usually involves using a validation set of data that the model hasn't seen before.
5. **Parameter Tuning**: Fine-tune the parameters of your model based on the results of your evaluation. This could involve techniques like cross-validation and grid search.
6. **Testing the Model**: Once you're satisfied with the performance of your model on the validation data, test the model with the test data set.
7. **Deployment**: If the model's performance is satisfactory, it is deployed in the real-world setting for making predictions on unseen data.
8. **Monitoring and Updating the Model**: The performance of the model needs to be monitored over time. The model could also be updated or retrained as new data becomes available.

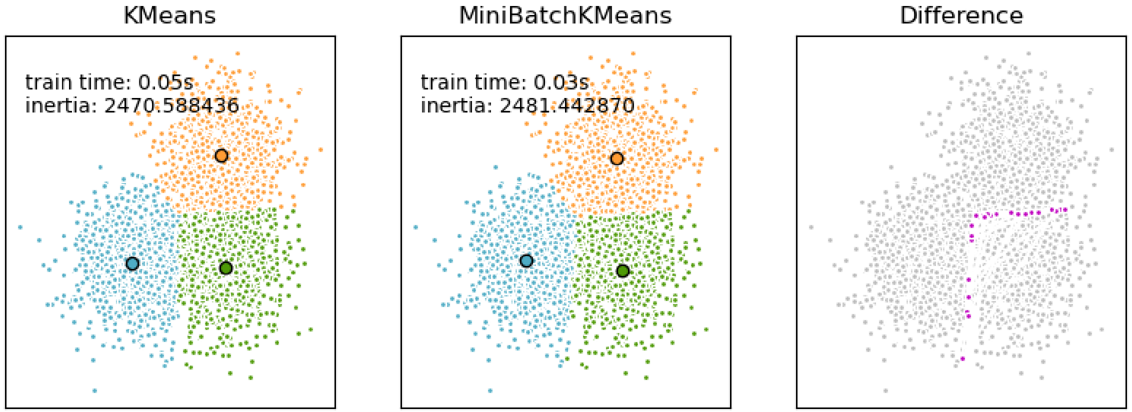
K-Means Clustering Method

It is a method of vector quantization, that aims to partition ***n*** observations into ***k*** clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid).

Given a set of observations *{x1, x2, ..., xn}*, where each observation is a **d**-dimensional real vector, ***k-means*** clustering aims to partition the ***n*** observations into **k** (≤ **n**) sets

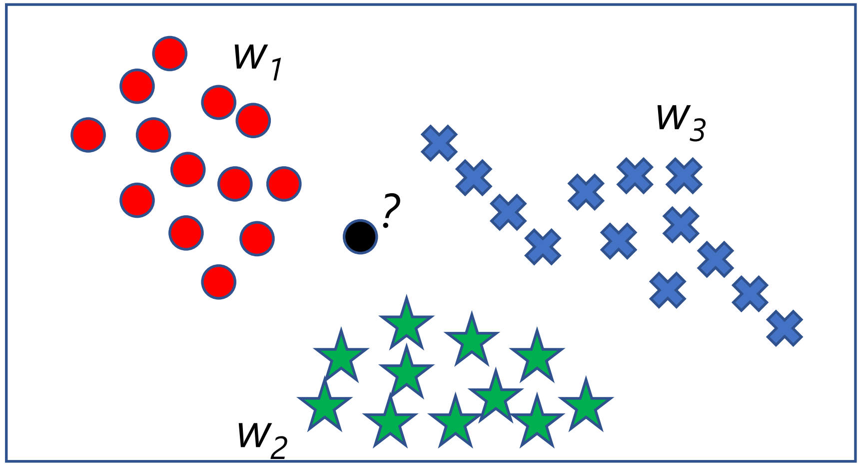
S = *{S1, S2, ..., Sk}* so as to minimize the within-cluster sum of squares (WCSS) (most notably is variance).

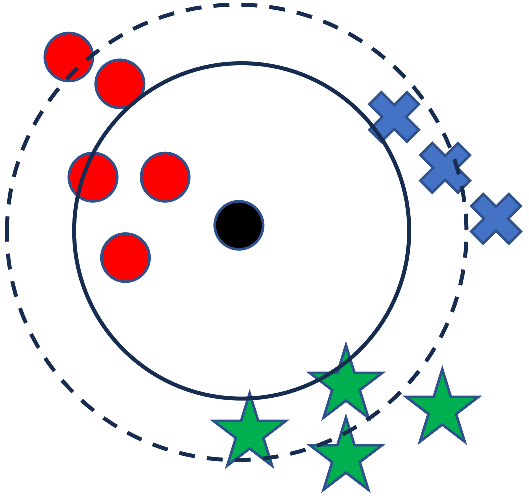




K-Nearest Neighbor Classification Method

In pattern recognition, the **k-nearest neighbours** algorithm (**k-NN**) is a non-parametric method used for classification and regression. In both cases, the input consists of the ***k*** closest training examples in the feature space.

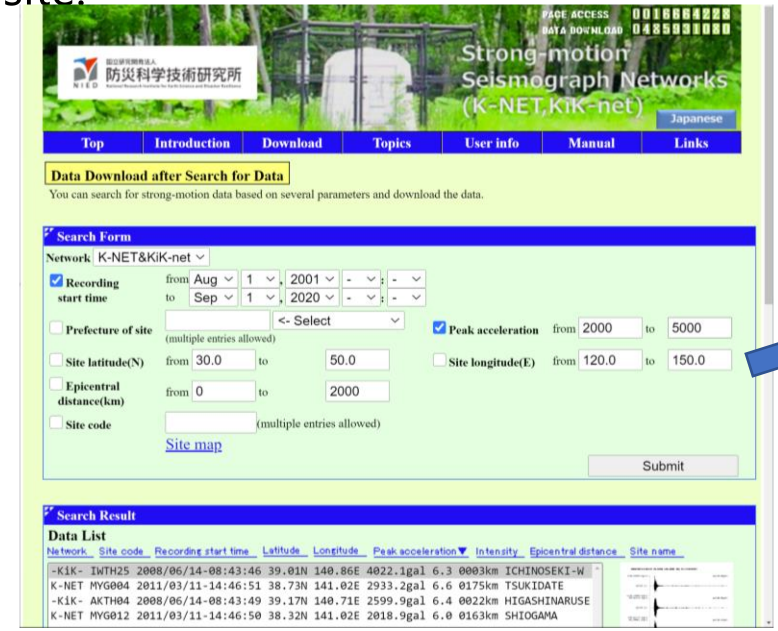




Exercise Data

I extracted information on earthquakes in Japan from: https://www.kyoshin.bosai.go.jp/.

This data includes PGA >= 60gal from all sensors since the year 2000. You can get info on the event and info on the site.



	A	B	C	D	E	F	G	H	I	J
1	HypLatitude(N)	HypLongitude(E)	Depth(km)	Magnitude(M)	NumberOfSites	Latitude(N)	Longitude(E)	PeakAcceleration(gal)	Intensity	EpicentralDistance(km)
2	39.03	140.88	8	7.2	655	39.01	140.86	4022.1	6.3	3
3	38.1	142.86	24	9	1226	38.73	141.02	2933.2	6.6	175
4	39.03	140.88	8	7.2	655	39.17	140.71	2599.9	6.4	22
5	38.1	142.86	24	9	1226	38.32	141.02	2018.9	6	163
6	38.1	142.86	24	9	1226	36.59	140.65	1845.2	6.4	258
7	38.1	142.86	24	9	1226	38.27	140.93	1807.8	6.3	170
8	42.69	142.01	37	6.7	473	42.87	141.82	1796.4	6.4	26
9	38.1	142.86	24	9	1226	36.16	140.49	1762.3	6.4	301
10	37.29	138.87	13	6.8	613	37.13	138.75	1750.2	6.2	21
11	32.74	130.81	11	6.5	337	32.8	130.82	1579.7	6.4	6
12	38.81	141.68	71	7	794	38.3	141.5	1571.3	6.2	59
13	36.16	137.46	4	5	180	36.04	137.49	1538.7	5.8	13
14	42.69	142.01	37	6.7	473	42.87	141.82	1504.8	6.7	26
15	37.29	138.87	13	6.8	613	37.31	138.79	1501.9	6.7	7
16	38.2	141.92	66	7.1	924	38.31	141.5	1495.8	5.4	38
17	35.38	133.85	11	6.6	594	35.43	133.83	1494	5.8	6
18	38.2	141.92	66	7.1	924	38.32	141.02	1480.6	5.8	80
19	38.1	142.86	24	9	1226	36.73	139.72	1444	6.2	317
20	38.1	142.86	24	9	1226	37.12	140.19	1425.3	6.1	259
21	39.03	140.88	8	7.2	655	38.97	141	1372.1	6	12
22	32.75	130.76	12	7.3	698	32.8	130.82	1362.1	6.5	7
23	38.1	142.86	24	9	1226	37.16	140.09	1335.4	6	266
24	38.1	142.86	24	9	1226	36.55	140.41	1311.9	6	277
25	38.1	142.86	24	9	1226	36.55	140.08	1304.8	6.5	301
26	38.81	141.68	71	7	794	39.18	141.39	1304.6	5.8	48
27	36.87	139.41	3	6.3	354	36.88	139.45	1300.3	6	4
28	38.1	142.86	24	9	1226	36.55	140.17	1291.1	6.3	294
29	38.2	141.92	66	7.1	924	38.73	141.02	1258.8	6.1	98
30	38.1	142.86	24	9	1226	37.23	141	1239.9	5.9	190

A1 | HypLatitude(N)

quakes_values

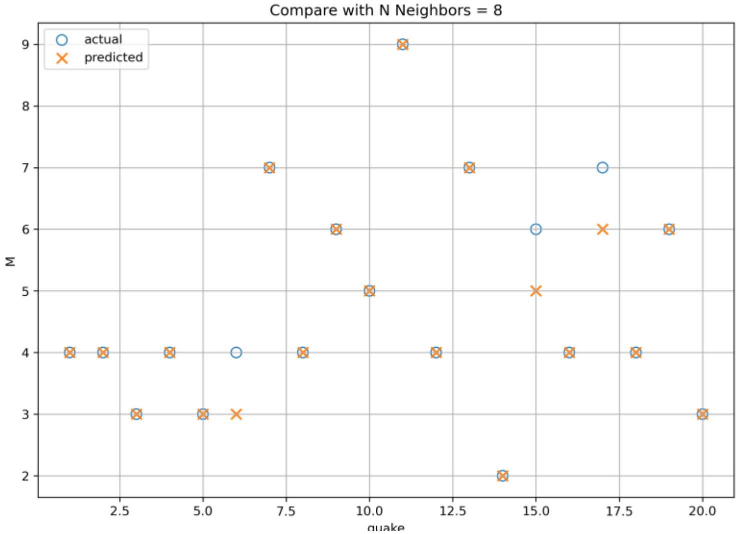
Average: 95.5382184 Count: 156970 Sum: 14995678.76 100%

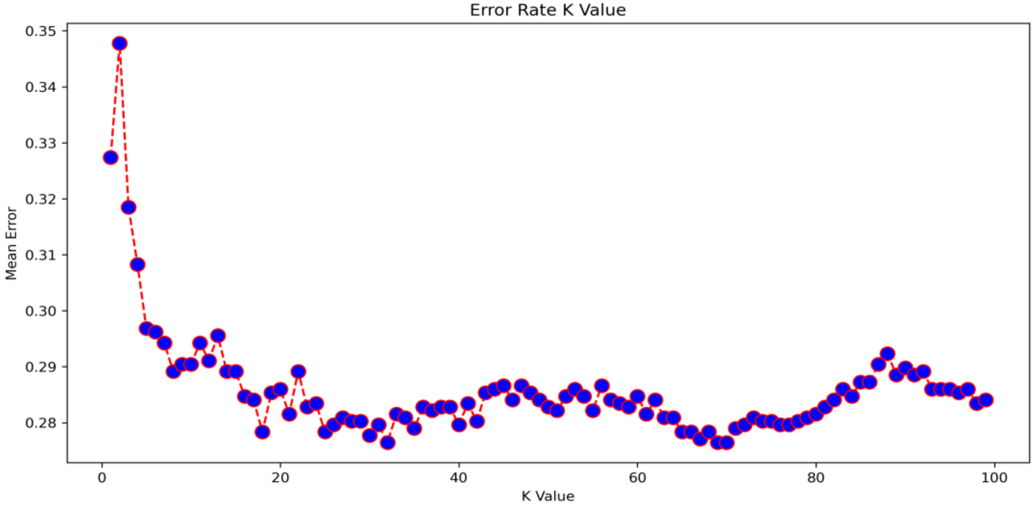
Example KNN

Input was: **NumberOfSites**, **PeakAcceleration**(gal), **Intensity**

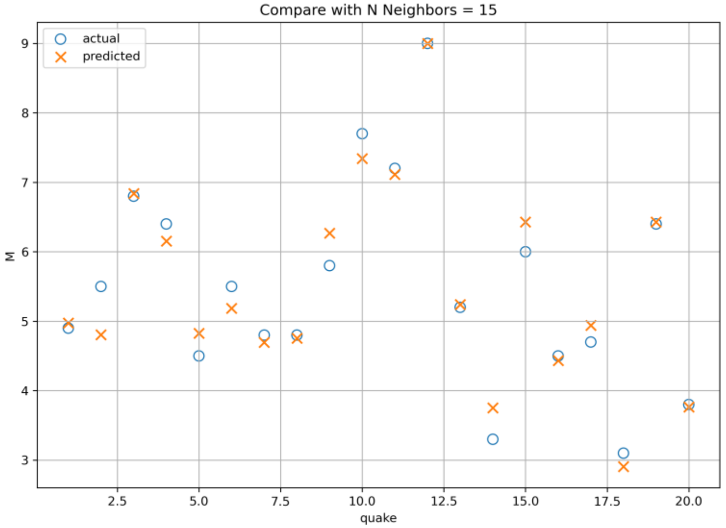
Output was: **Magnitude**

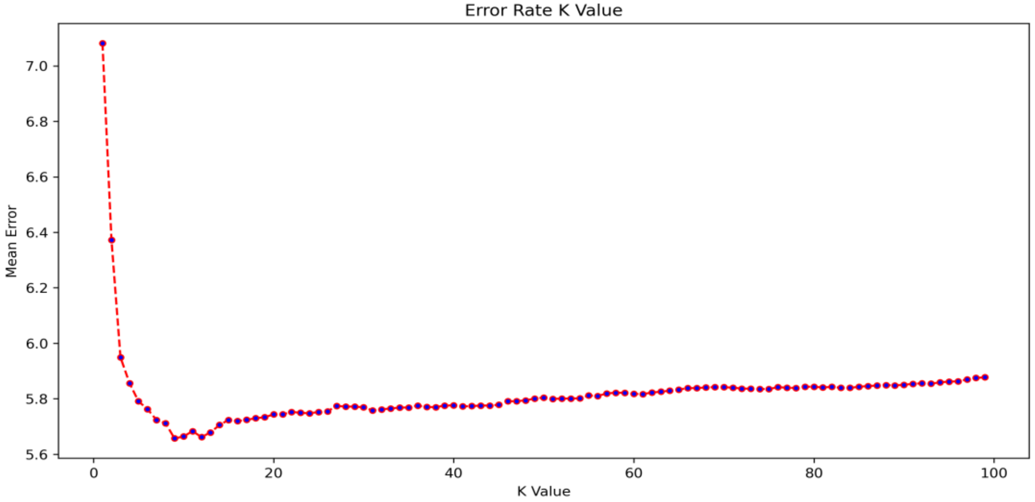
Classification
Output was modified to be integers (by rounding), then to string.





Regression
Output was not modified.

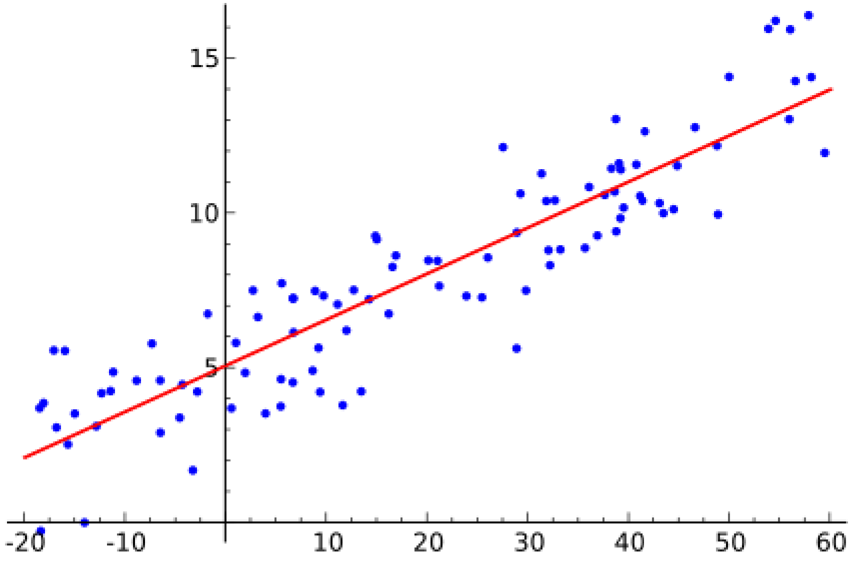




Linear Regression

$$y = ax + b$$

which describes a line with slope a and y -intercept b. In general, such a relationship may not hold exactly for the largely unobserved population of values of the independent and dependent variables.



This is just a 2D Linear Problem.

How about higher dimensions with non-linear relationships?

Dense Neural Networks

Forward Propagation

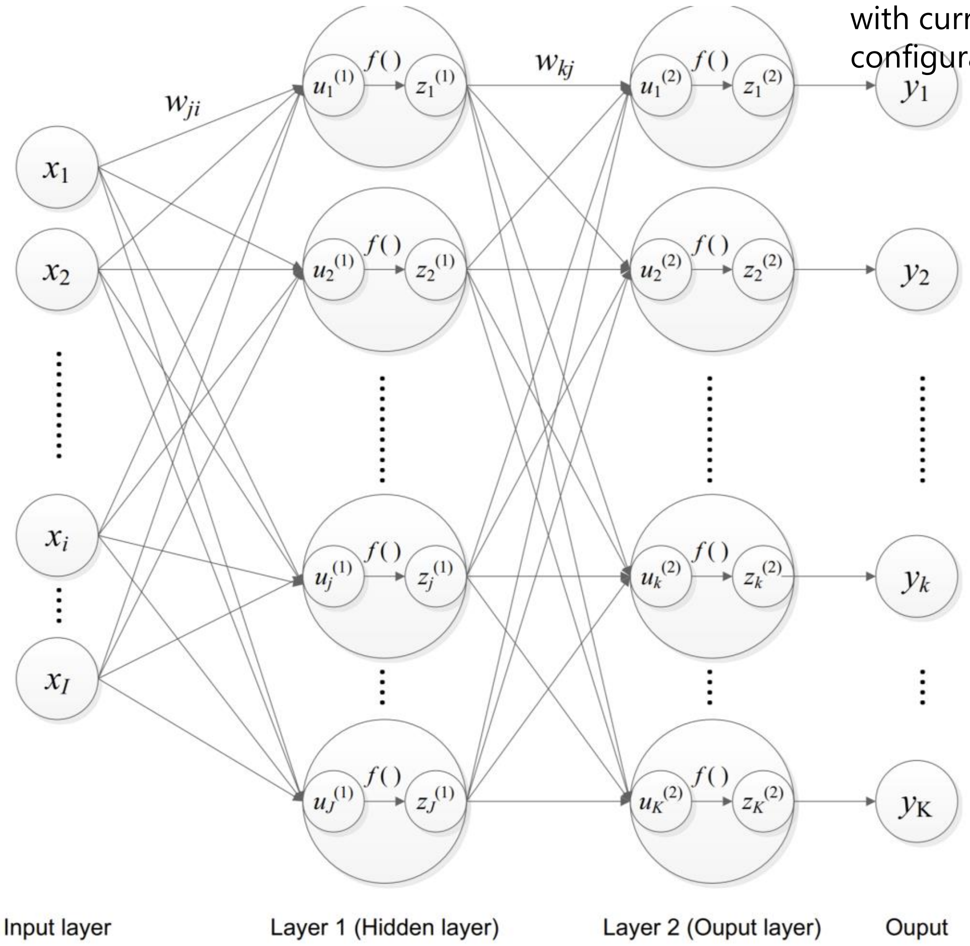
To calculate output with current network configuration.

A deep learning neural network comprises of at least one input layer, one output layer and several hidden layers in between. The input layer and its nodes represent the predictive features, whereas the output layer and its nodes are the target predictions.

While inputs and outputs can be physical values that correspond to actual data, the values in the hidden layer aren't something to observe directly. Nevertheless, each node in any hidden layer represents an aggregation of information to capture interactions from input data.

Hyperparameters:

1. Epochs count
2. Number of hidden layers
3. Number of neurons
4. Learning rate and training optimization algorithm
5. Neuron activation functions
6. Batch size



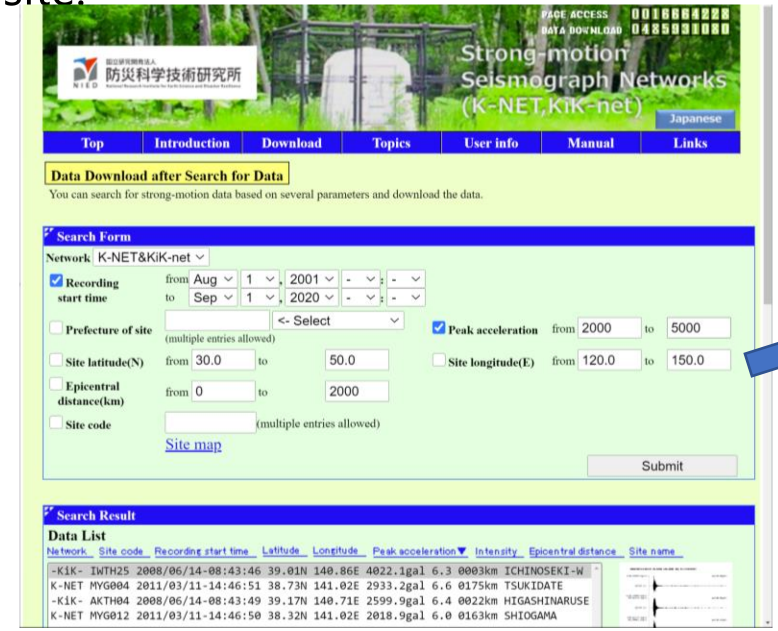
To update weights & biases to improve current configuration.

Backward Propagation

Exercise Data

I extracted information on earthquakes in Japan from: https://www.kyoshin.bosai.go.jp/.

This data includes PGA >= 60gal from all sensors since the year 2000. You can get info on the event and info on the site.



A1 | fx HypLatitude(N)

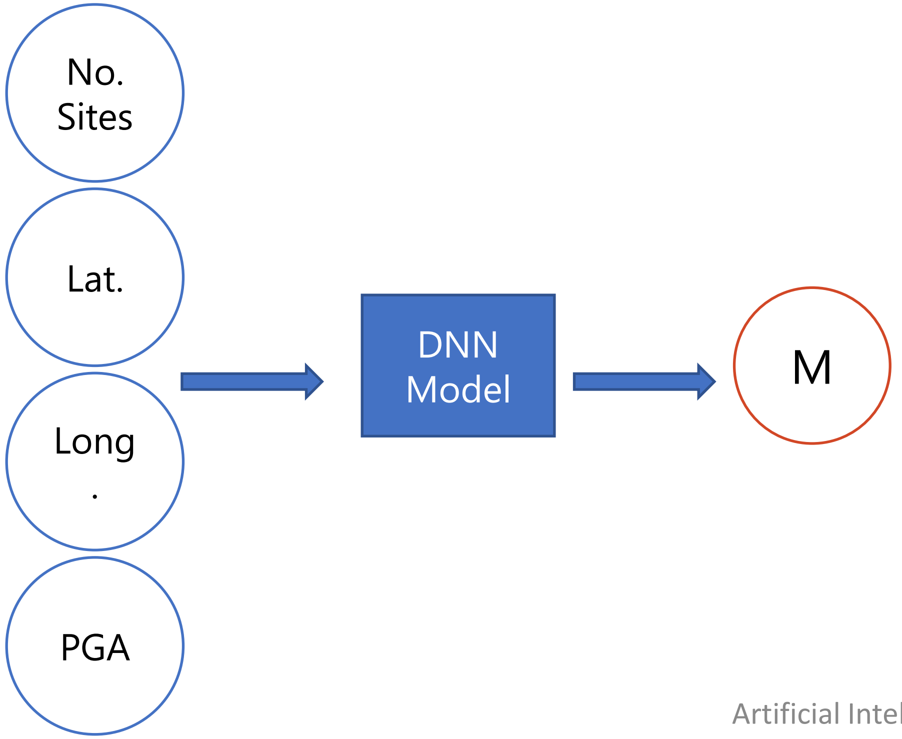
	A	B	C	D	E	F	G	H	I	J
1	HypLatitude(N)	HypLongitude(E)	Depth(km)	Magnitude(M)	NumberOfSites	Latitude(N)	Longitude(E)	PeakAcceleration(gal)	Intensity	EpicentralDistance(km)
2	39.03	140.88	8	7.2	655	39.01	140.86	4022.1	6.3	3
3	38.1	142.86	24	9	1226	38.73	141.02	2933.2	6.6	175
4	39.03	140.88	8	7.2	655	39.17	140.71	2599.9	6.4	22
5	38.1	142.86	24	9	1226	38.32	141.02	2018.9	6	163
6	38.1	142.86	24	9	1226	36.59	140.65	1845.2	6.4	258
7	38.1	142.86	24	9	1226	38.27	140.93	1807.8	6.3	170
8	42.69	142.01	37	6.7	473	42.87	141.82	1796.4	6.4	26
9	38.1	142.86	24	9	1226	36.16	140.49	1762.3	6.4	301
10	37.29	138.87	13	6.8	613	37.13	138.75	1750.2	6.2	21
11	32.74	130.81	11	6.5	337	32.8	130.82	1579.7	6.4	6
12	38.81	141.68	71	7	794	38.3	141.5	1571.3	6.2	59
13	36.16	137.46	4	5	180	36.04	137.49	1538.7	5.8	13
14	42.69	142.01	37	6.7	473	42.87	141.82	1504.8	6.7	26
15	37.29	138.87	13	6.8	613	37.31	138.79	1501.9	6.7	7
16	38.2	141.92	66	7.1	924	38.31	141.5	1495.8	5.4	38
17	35.38	133.85	11	6.6	594	35.43	133.83	1494	5.8	6
18	38.2	141.92	66	7.1	924	38.32	141.02	1480.6	5.8	80
19	38.1	142.86	24	9	1226	36.73	139.72	1444	6.2	317
20	38.1	142.86	24	9	1226	37.12	140.19	1425.3	6.1	259
21	39.03	140.88	8	7.2	655	38.97	141	1372.1	6	12
22	32.75	130.76	12	7.3	698	32.8	130.82	1362.1	6.5	7
23	38.1	142.86	24	9	1226	37.16	140.09	1335.4	6	266
24	38.1	142.86	24	9	1226	36.55	140.41	1311.9	6	277
25	38.1	142.86	24	9	1226	36.55	140.08	1304.8	6.5	301
	38.81	141.68	71	7	794	39.18	141.39	1304.6	5.8	48
27	36.87	139.41	3	6.3	354	36.88	139.45	1300.3	6	4
28	38.1	142.86	24	9	1226	36.55	140.17	1291.1	6.3	294
29	38.2	141.92	66	7.1	924	38.73	141.02	1258.8	6.1	98
30	38.1	142.86	24	9	1226	37.23	141	1239.9	5.9	190

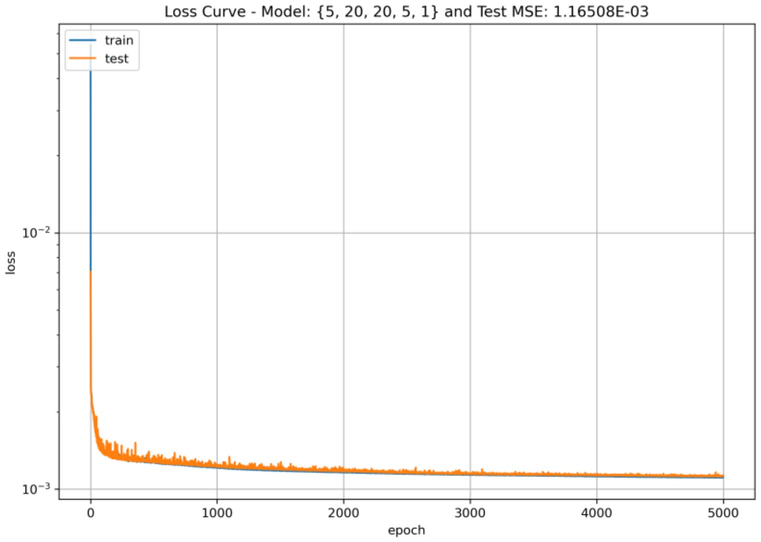
quakes_values

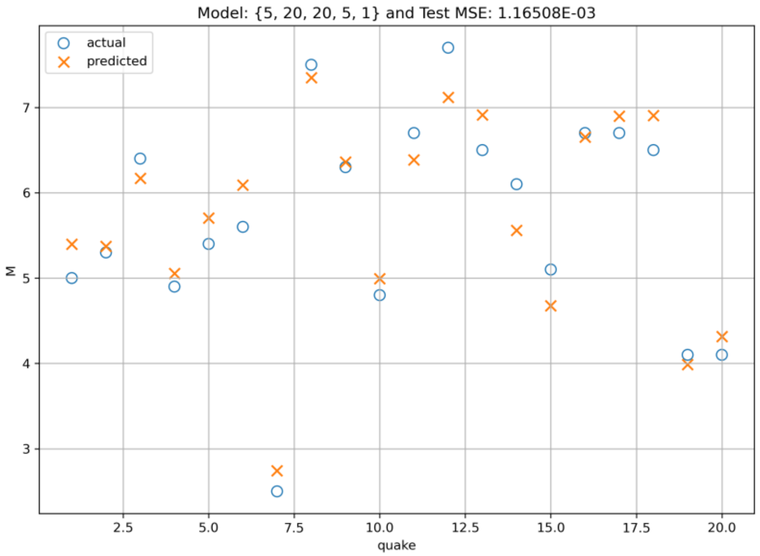
Average: 95.5382184 Count: 156970 Sum: 14995678.76 100%

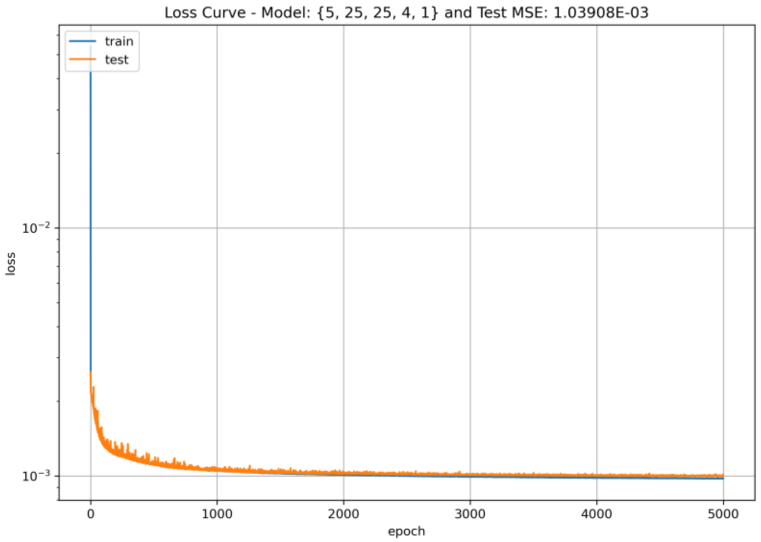
DNN Example 1

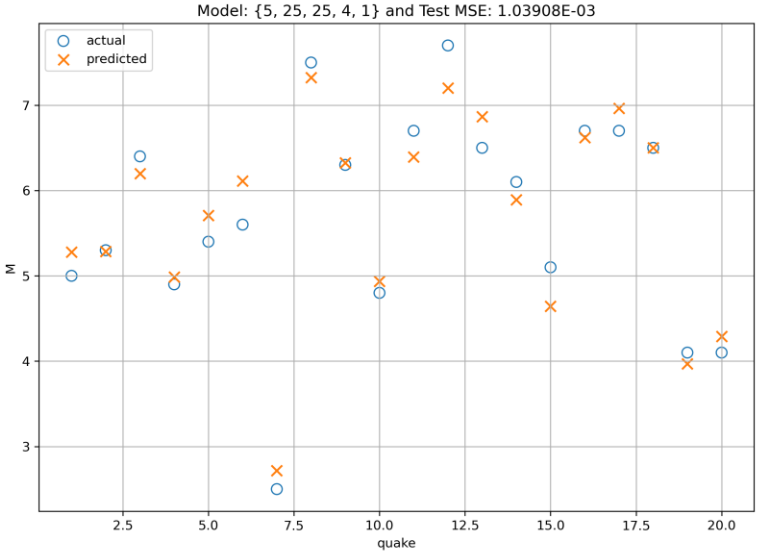
First model was simple, I tried to predict the Magnitude of the earthquake from the number of sites that seismic activity has reached, and information from just one sensor (Lat, Long, and PGA).





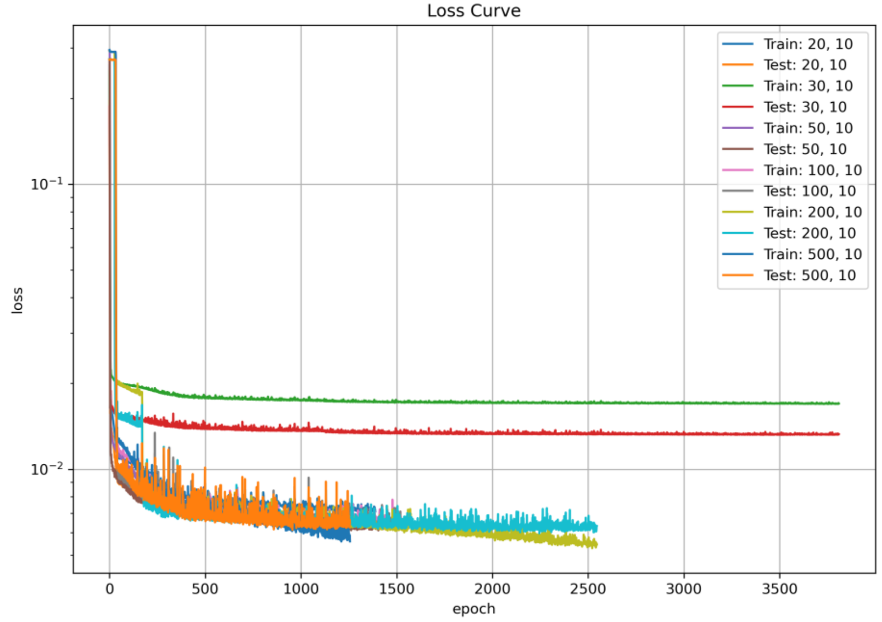


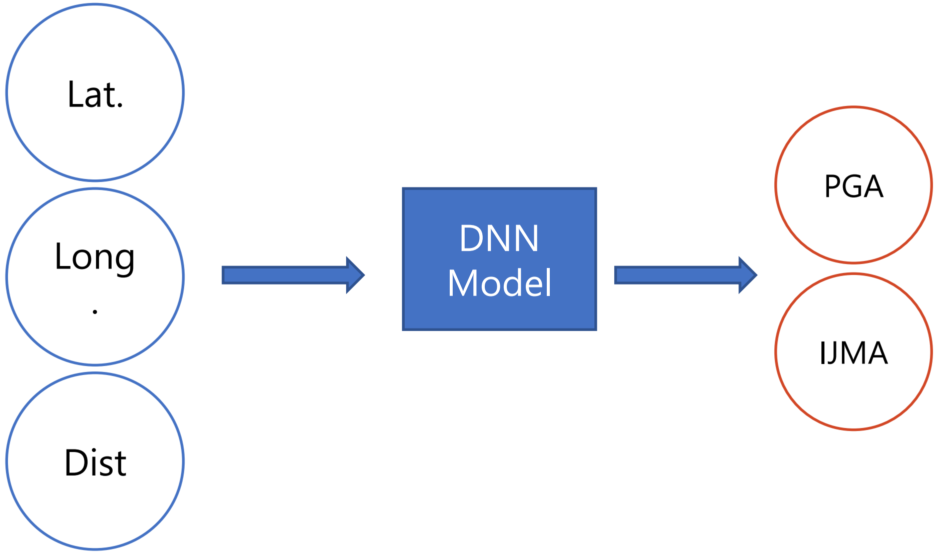


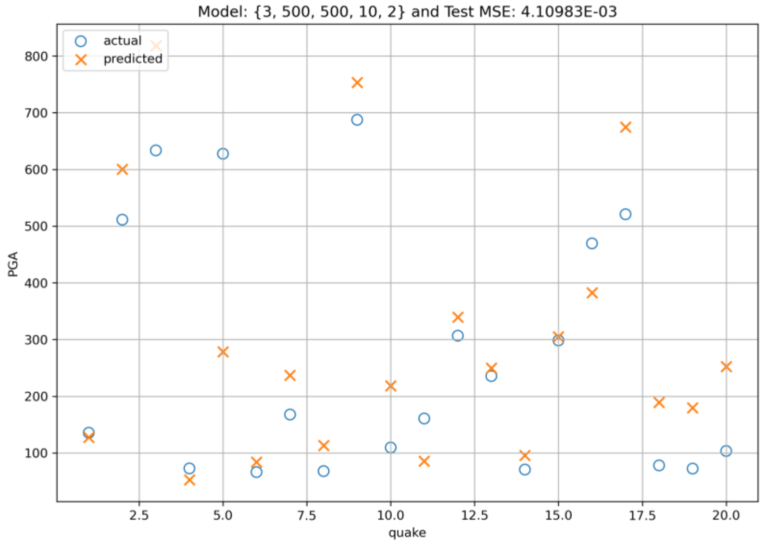


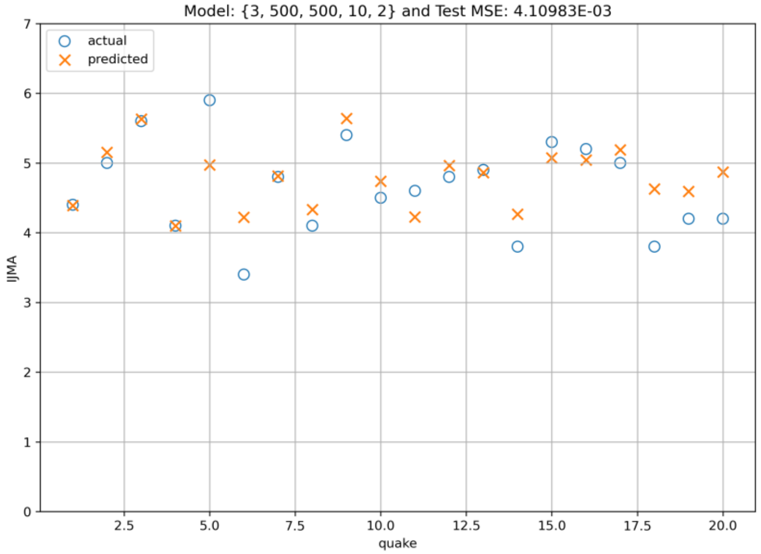
DNN Example 2

Second model I tried to predict the PGA and IJMA of the site for one earthquake event from the (Lat, Long) of the site and the epicentral distance.

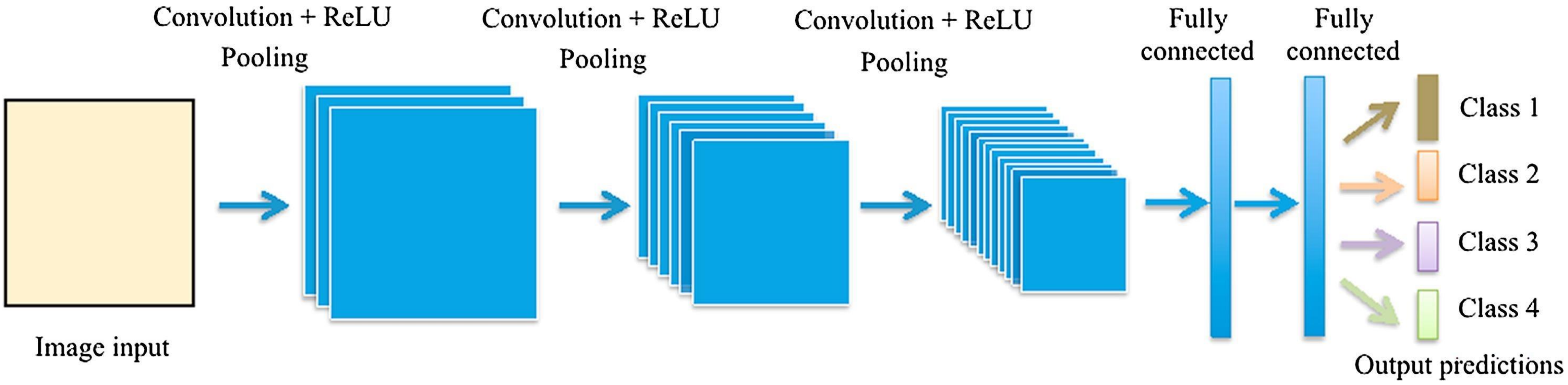








Convolutional Neural Network



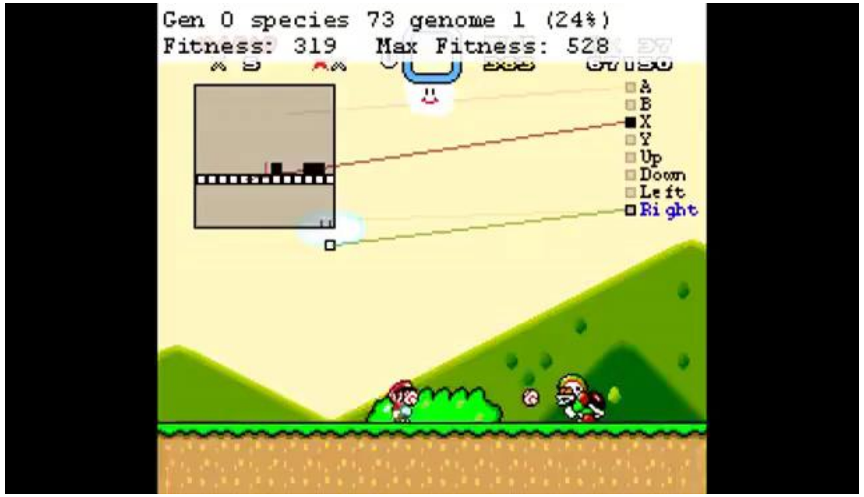
Evolutionary Algorithms

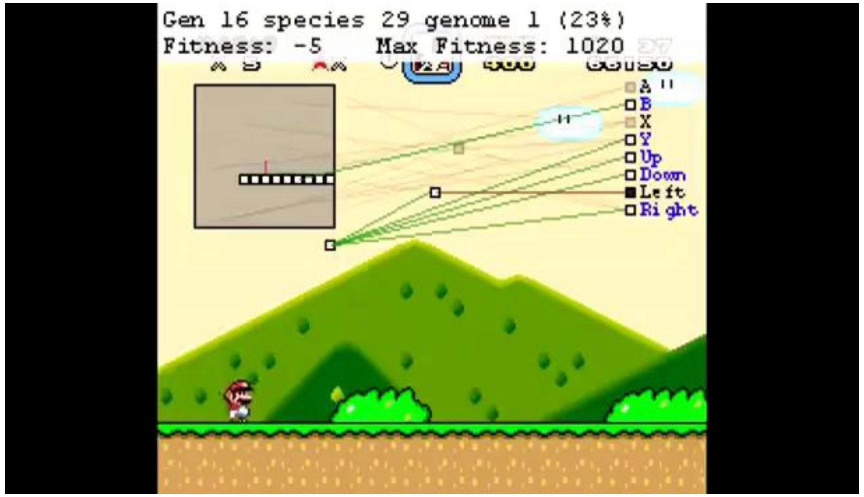
An ***evolutionary algorithm*** is a generic population-based metaheuristic optimization algorithm. An EA uses mechanisms inspired by biological evolution, such as reproduction, mutation, recombination, and selection.

They are a family of population-based trial and error problem solvers with a stochastic optimization character. ***The population will gradually evolve to increase in fitness.***

Types of EA are:

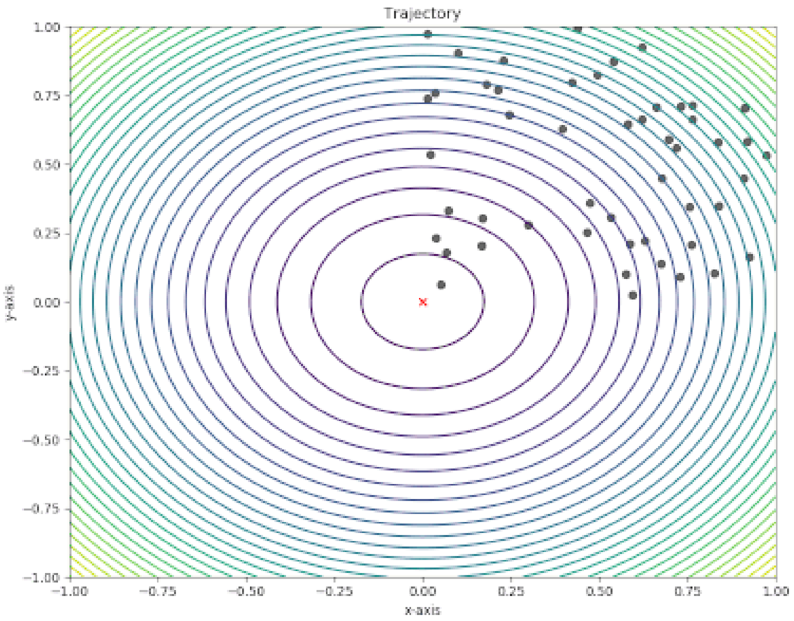
- **Genetic Algorithms**
- **Particle Swarm Optimization**



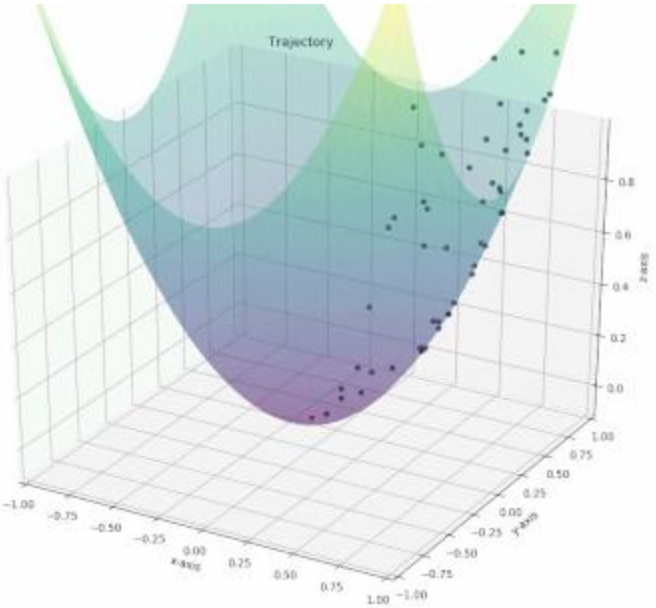


PSO

Particle swarm optimization (PSO) is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. It solves a problem by having a ***population*** of candidate solutions, here dubbed ***particles***, and moving these particles around in the search-space according to simple mathematical formulae over the particle's ***position*** and ***velocity***. Each particle's movement is influenced by its local best known position, but is also guided toward the best known positions in the search-space, which are ***updated as better positions*** are found by other particles. This is expected to move the swarm toward the best solutions.



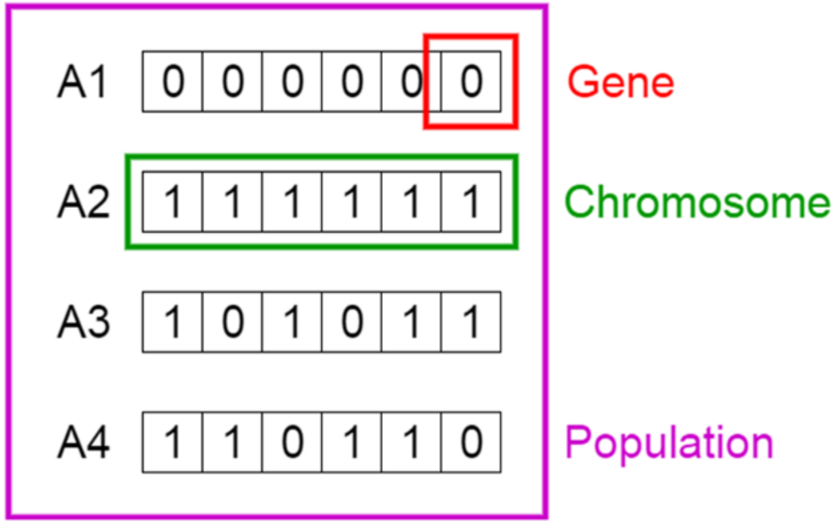
G



GA

Pseudocode for any Genetic Algorithm

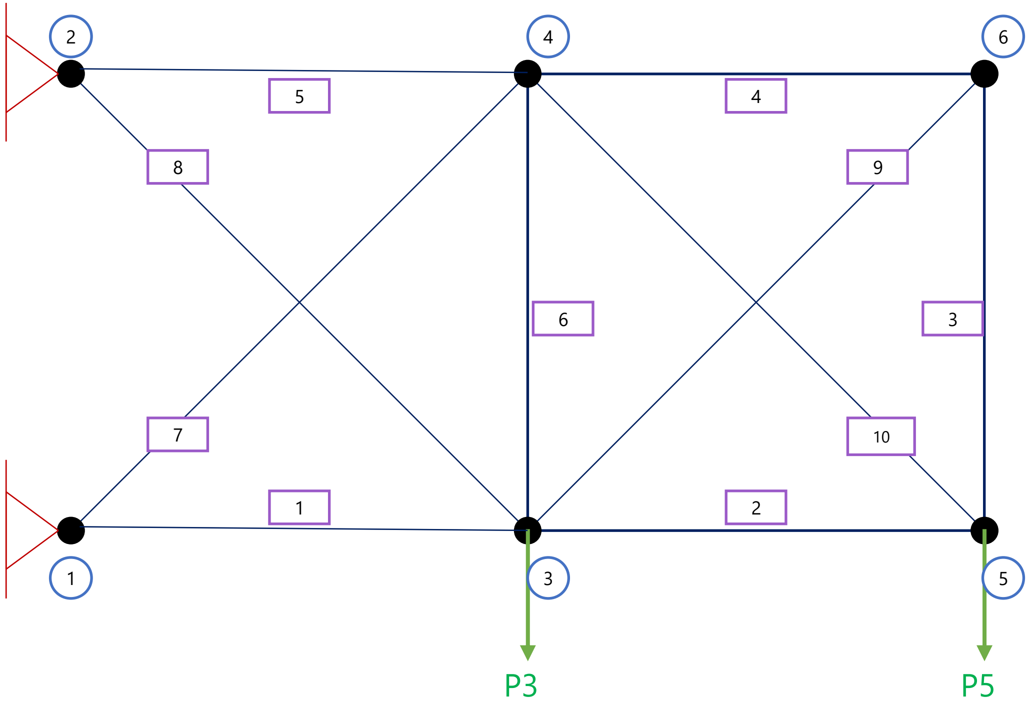
1. produce an initial population of individuals
2. evaluate the fitness of all individuals
3. while termination condition not met do
   a) select fitter individuals for reproduction
   b) recombine between individuals
   c) mutate individuals
   d) evaluate the fitness of the modified individuals
   e) generate a new population
4. End while



Structural Design Problem

Our first example considers a well-known problem corresponding to a 10-bar truss nonconvex optimization (Sunar & Belegundu, 1991). In this problem the cross-sectional area for each of the **10 members** in the structure are being optimized towards the minimization of total weight. The **cross-sectional area** varies between 0.1 to 35.0 in^2. Constraints are specified in terms of **stress** and **displacement** of the truss members. The allowable stress for each member is 25,000 psi for both tension and compression, and the allowable displacement on the nodes is ±2 in, in the x and y directions. The **density of the material** is 0.1 lb/in3, Young's modulus is E = 10^4 ksi and vertical downward loads of 100 kips are applied at nodes 3 and 5. In total, the problem has a variable dimensionality of 10 and constraint dimensionality of 32 (10 tension constraints, 10 compression constraints, and 12 displacement constraints).

The **Weight** should be minimized as possible and still satisfy the **engineering demands** of **displacement** and **stresses**. Therefore, the Weight is set as the objective function, minimizing it as much as possible is the target.

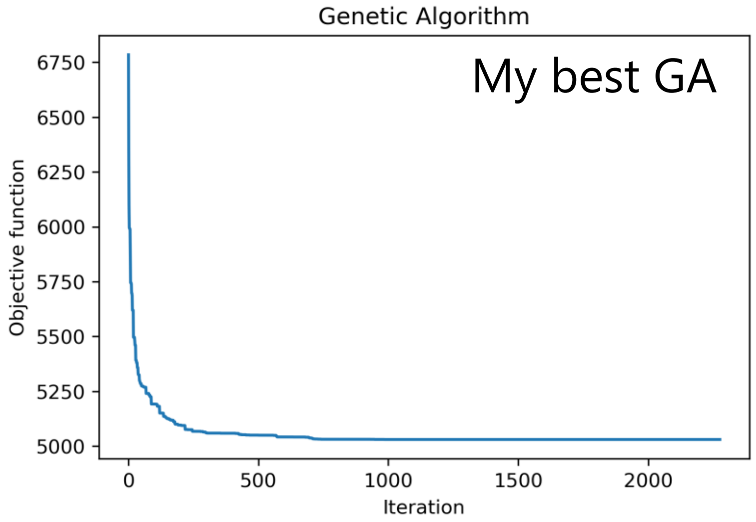


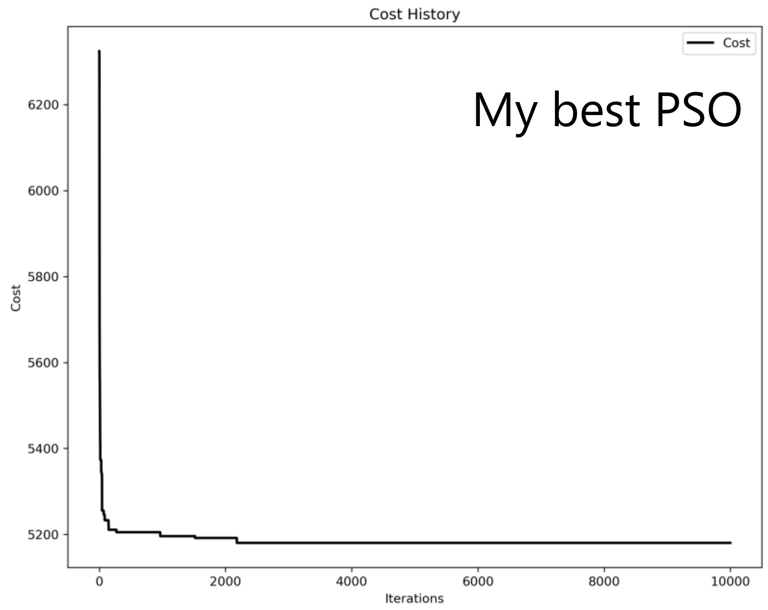
What is the best **combination** for the **areas** of sections **1-10**?

*When the **constraints** are violated, the Weight is set to a **Fixed Penalty** of **100,000 lb** to encourage the algorithms to avoid it.*

Compare PSO and GA

Info	Benchmark (Ghasemi, 1997)	My best GA (From just 10 runs)	My best PSO (From just 10 runs)
Member 1	24.85	22.37	22.27
Member 2	16.35	15.12	18.00
Member 3	0.109	1.034	0.423
Member 4	0.109	0.100	0.118
Member 5	25.73	30.45	29.54
Member 6	0.106	0.103	0.407
Member 7	21.41	21.97	22.65
Member 8	8.700	5.79	8.337
Member 9	0.122	0.11	0.159
Member 10	22.30	21.99	20.55
Total Weight (lb)	**5095.7**	**5030.29**	5180.30
Details		population_size':10000, iterations=10000.	n_particles=10000, iterations=10000.





Assignments & Exercises