CS 480

Introduction to Artificial Intelligence

November 9th, 2021

Announcements / Reminders

- Quiz #02: due on Sunday (11/14/21) at 11:00 PM CST
- Programming Assignment #02:
  - TBA
- Written Assignment #03:
  - TBA
- Final Exam:
  - Thursday, December 2nd, 2021 (during lecture time)

Plan for Today

- **Casual Introduction to Machine Learning**

Main Machine Learning Categories

Supervised learning	Unsupervised learning	Reinforcement learning
Supervised learning is one of the most common techniques in machine learning. It is based on **known relationship(s) and patterns within data** (for example: relationship between inputs and outputs). Frequently used types: **regression**, and **classification**.	**Unsupervised learning** involves finding underlying patterns within data. Typically used in **clustering** data points (similar customers, etc.)	Reinforcement learning is inspired by behavioral psychology. It is **based on a rewarding / punishing an algorithm**. Rewards and punishments are based on algorithm's action within its environment.

Supervised Learning

Given a training set of N example input-output (feature-label) pairs

$$(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$$

where each pair was generated by some UNKNOWN function

$$y = f(x)$$

discover a function (model) $h(x)$ (hypothesis) that approximates the true function $f(x)$.

Choosing Hypothesis / Model

Given a training set of N example input-output (feature-label) pairs

$$(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$$

where each pair was generated by

$$y = f(x)$$

Ideally, we would like our model $h(x)$ (hypothesis) that approximates the true function $f(x)$ to be:

$$h(x) = y = f(x)$$ **(consistent hypothesis)**

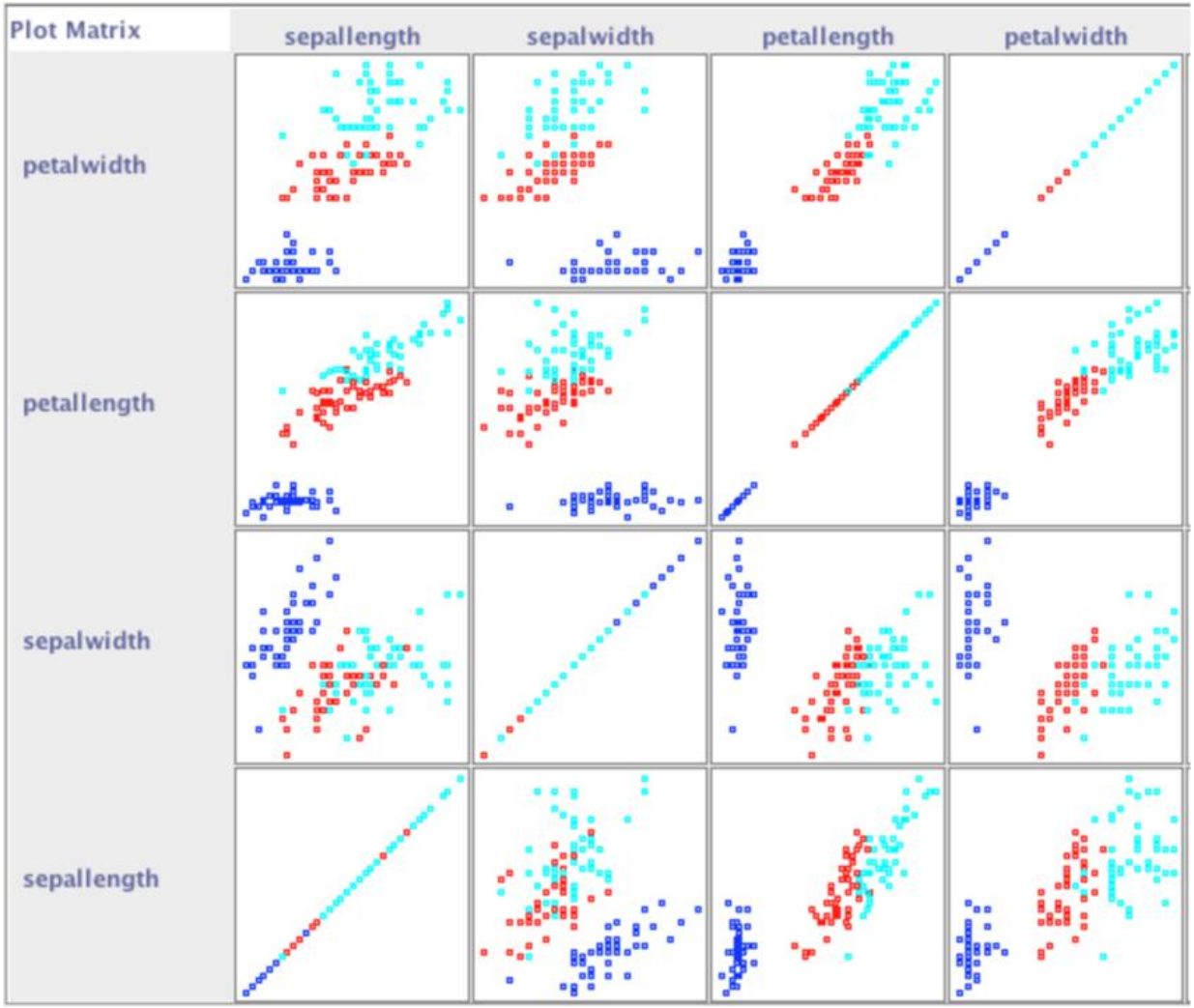
Choosing Hypothesis / Model

Typically consistent hypothesis is impossible or difficult to achieve:

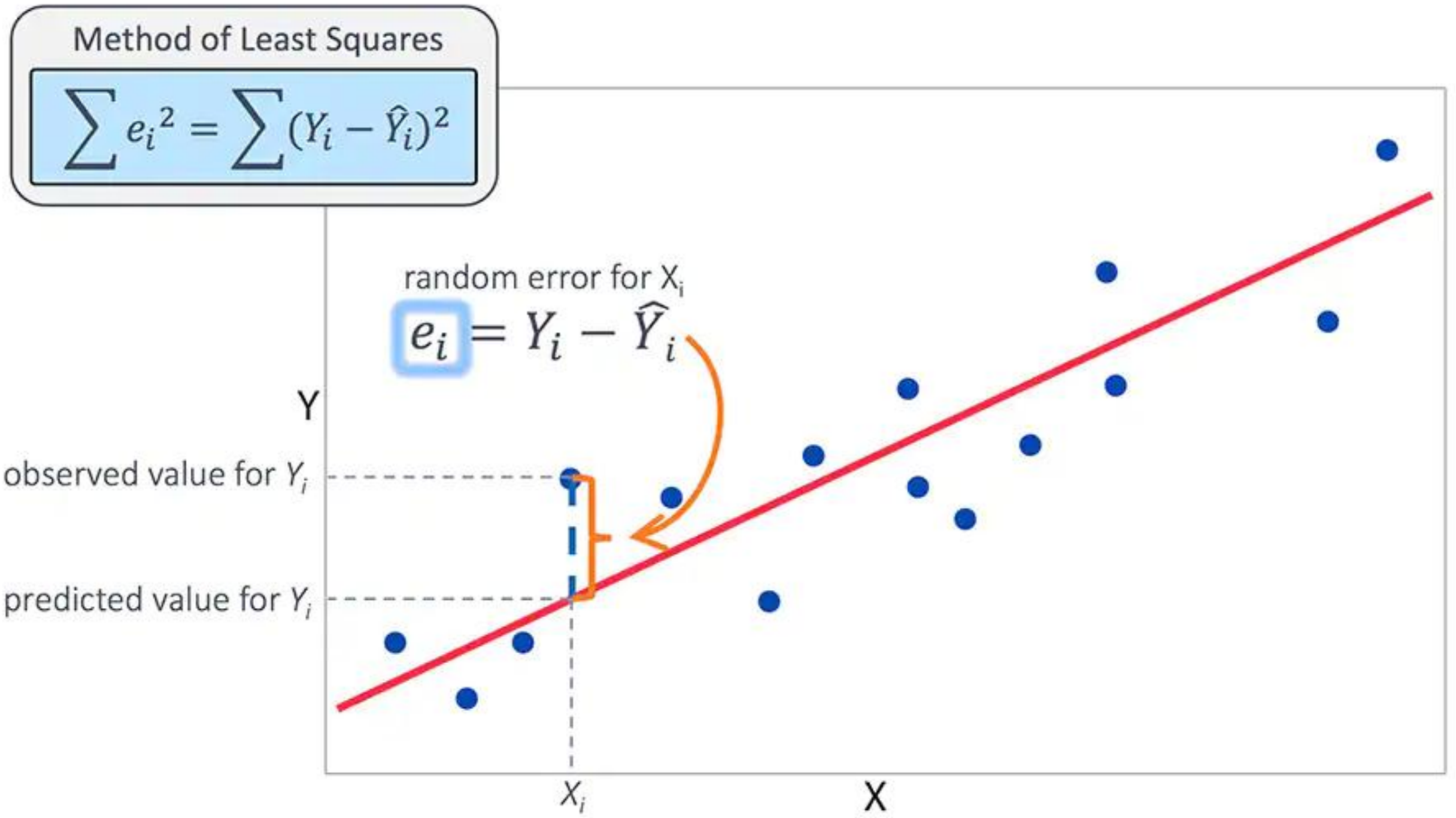
- **use best-fit model / hypothesis**

Our model needs to be tested on the test set inputs (data the model has not "seen" yet) to see how well it generalizes (how accurately it predicts the outputs of the test set).

Exploratory Data Analysis



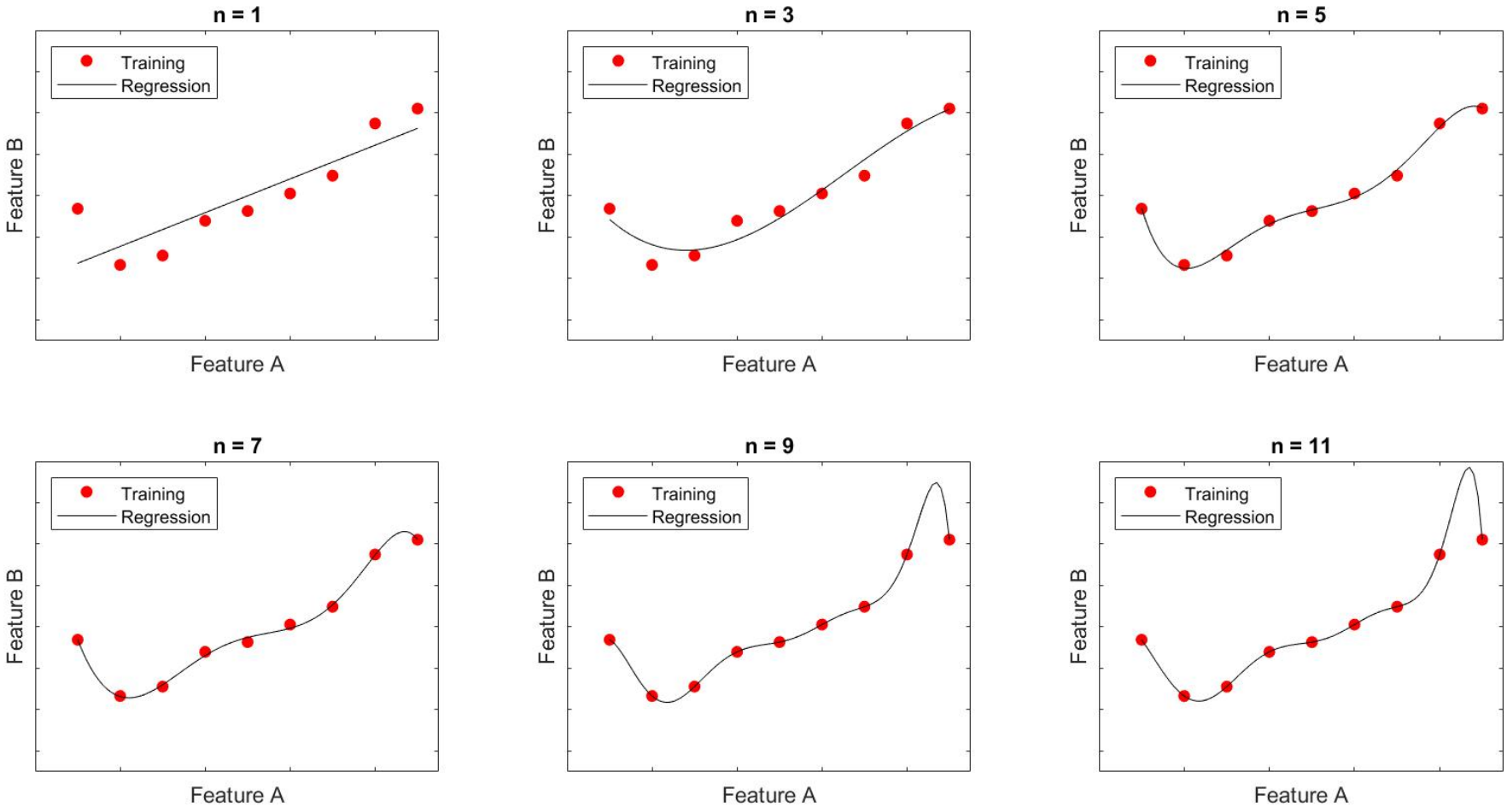
Linear Regression Using Least-Squares



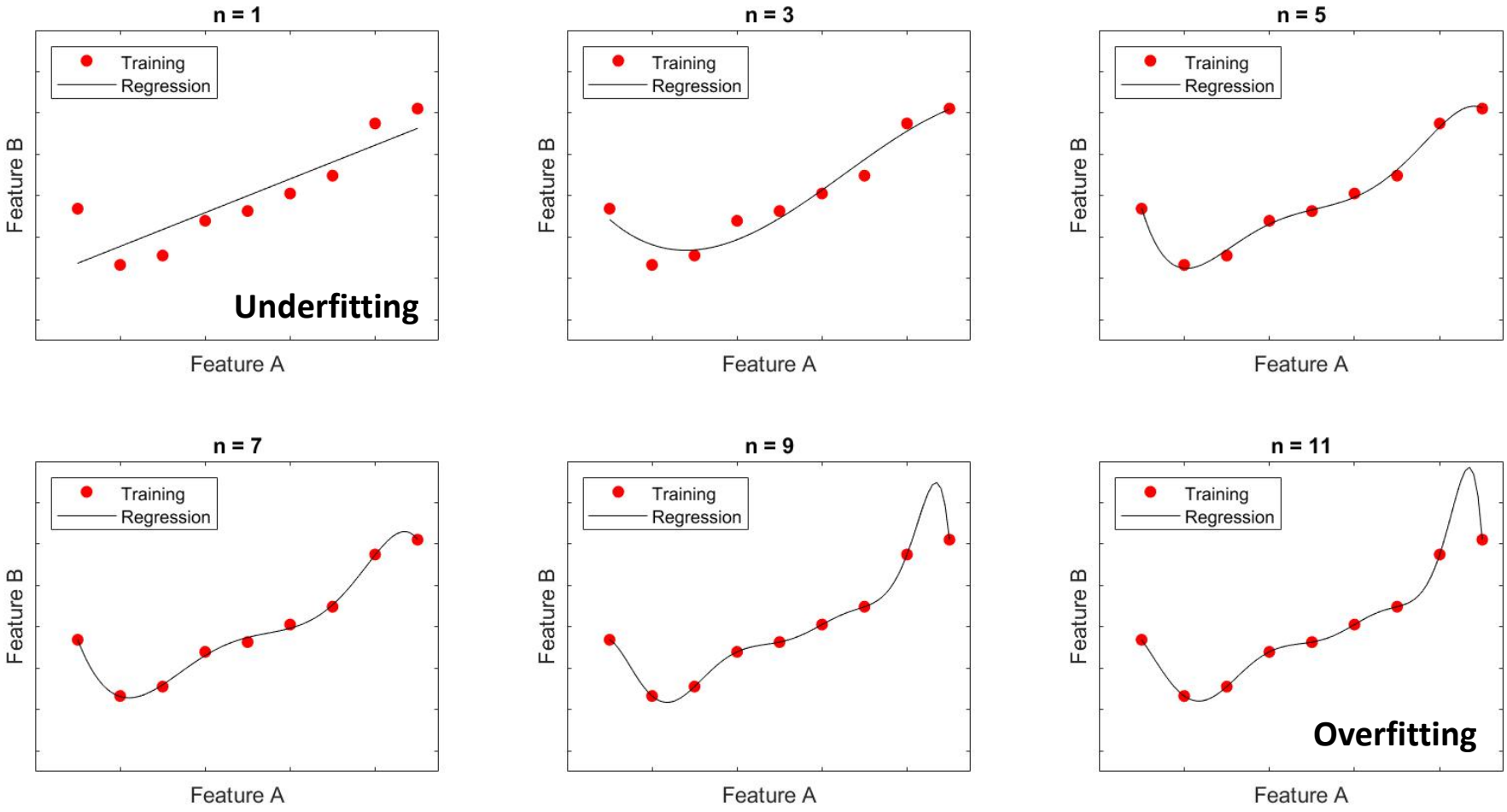
The goal is to find the line y = ax + b that minimizes the amount of error.

Source: https://www.jmp.com/en_us/statistics-knowledge-portal/what-is-multiple-regression/fitting-multiple-regression-model.html

Regression

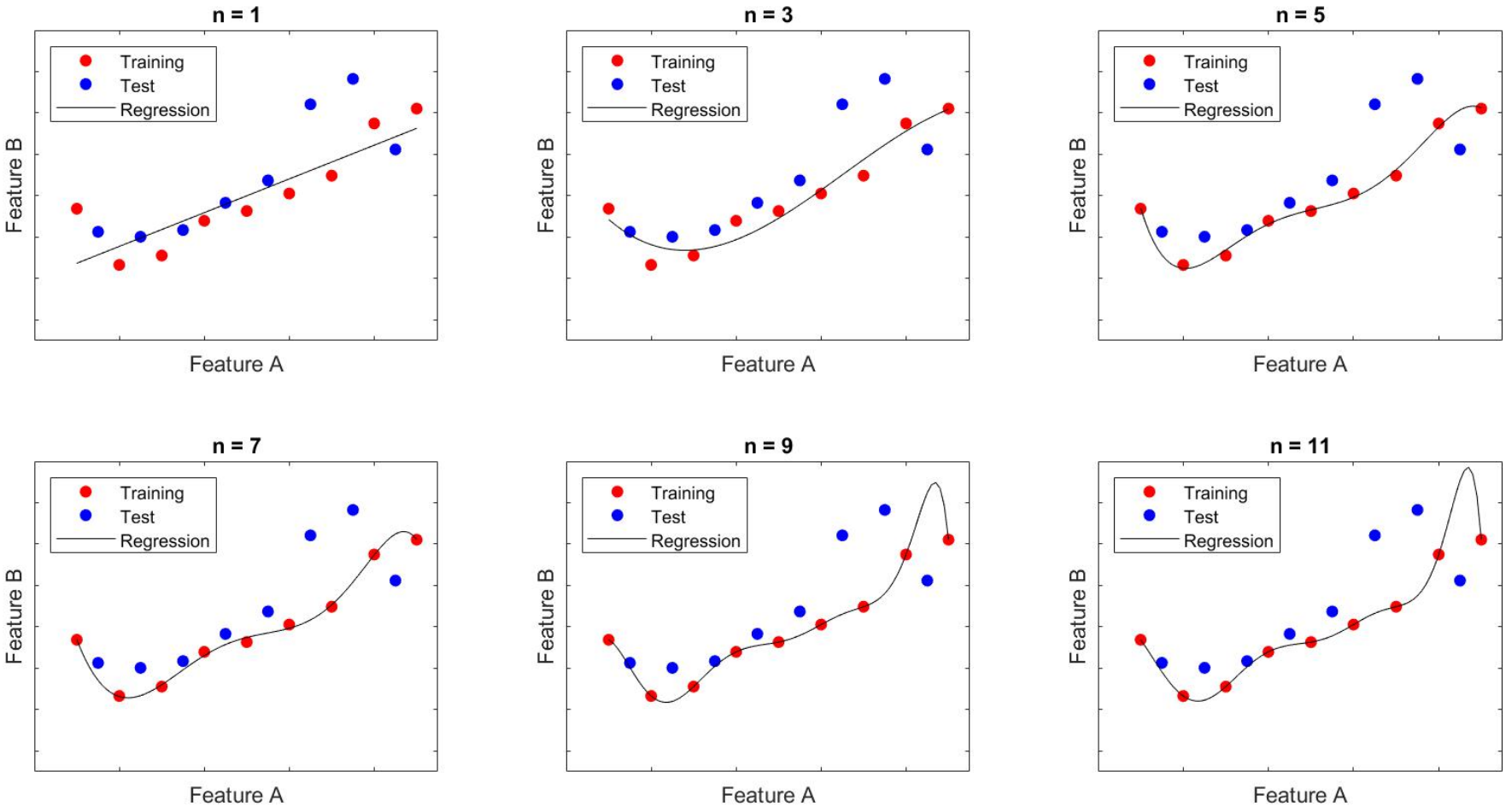


Regression: Underfitting / Overfitting

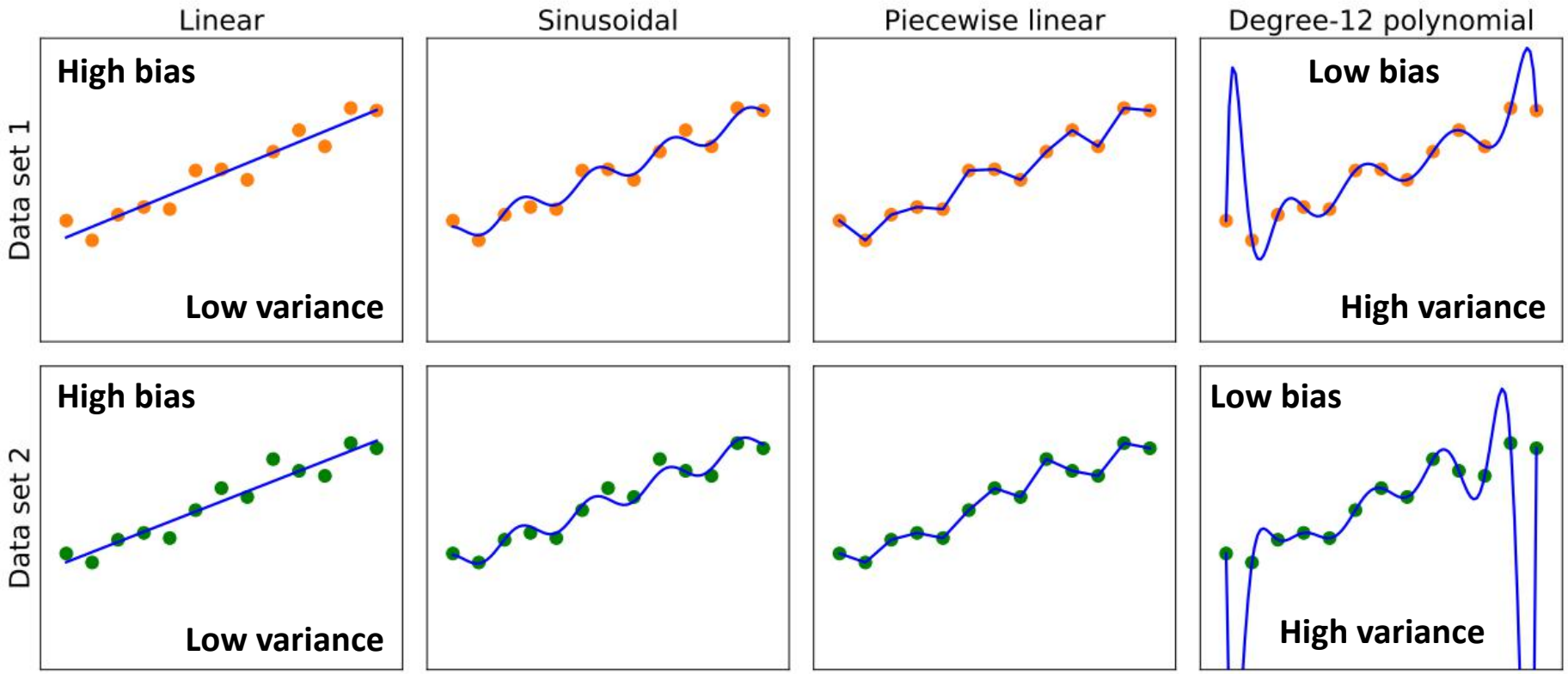


Underfitting: "failing" to find pattern in the data.

Regression


n = 1
n = 3
n = 5
n = 7
n = 9
n = 11
Training
Test
Regression
Feature A
Feature B

Bias vs. Variance



Bias: the tendency of a predictive hypothesis to deviate from the expected value when averaged over different training sets

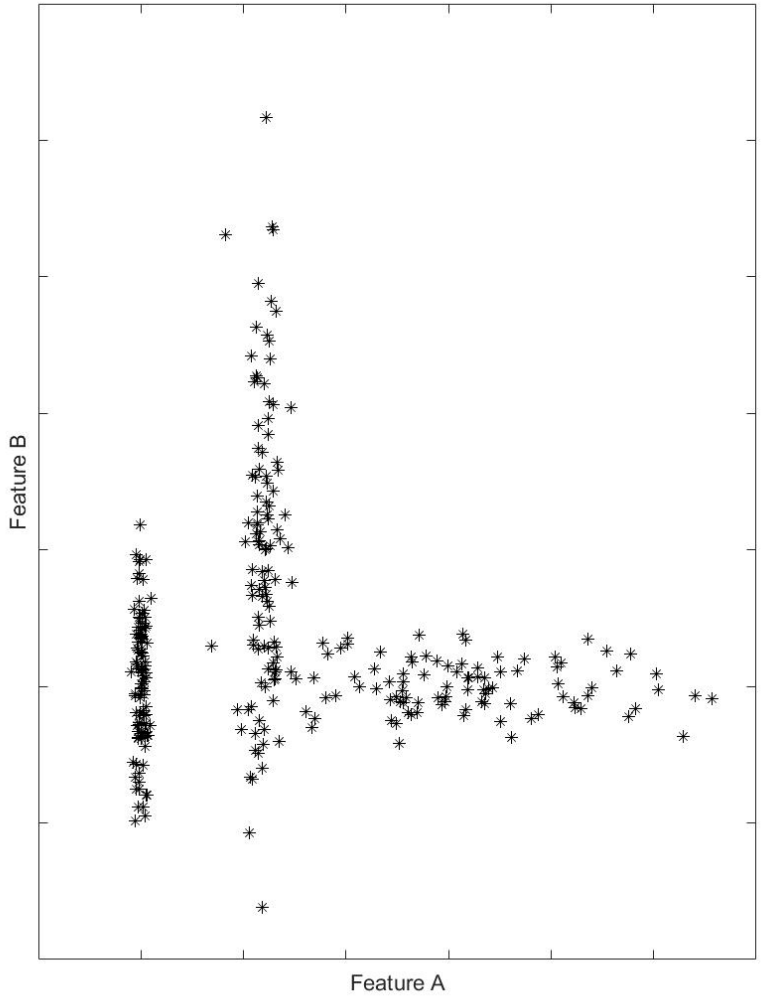
Variance: the amount of change in the hypothesis (model) due to fluctuations in training data

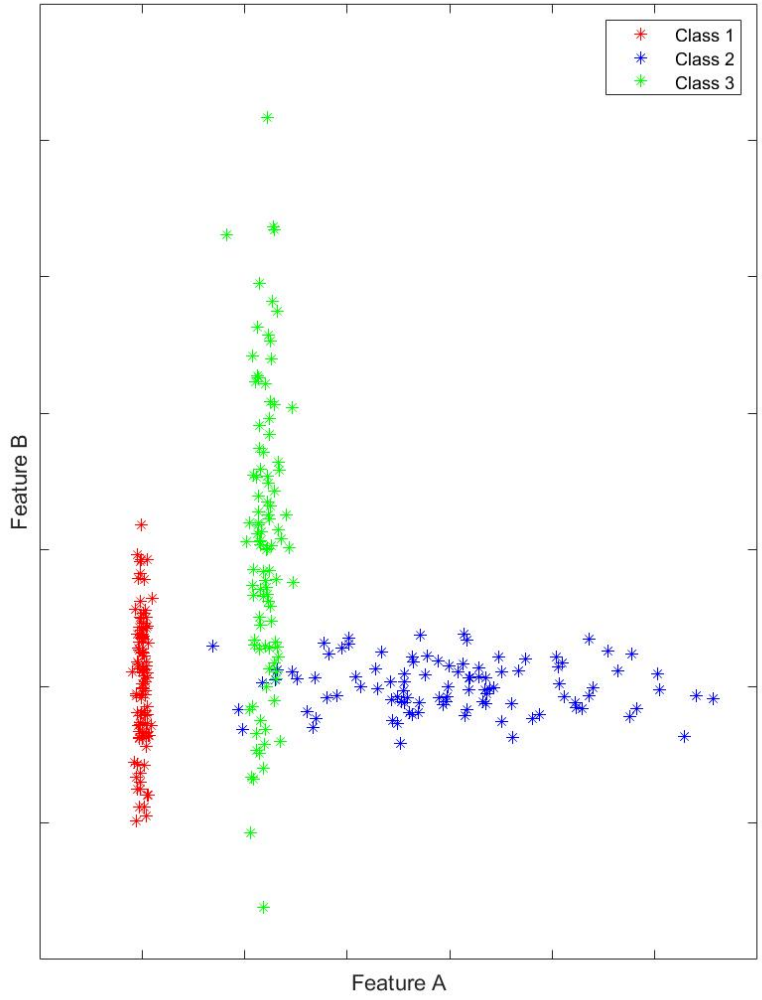
Training / Validation / Test Sets

In order to create the best model possible, given some (relatively large data set), we should divid it into:

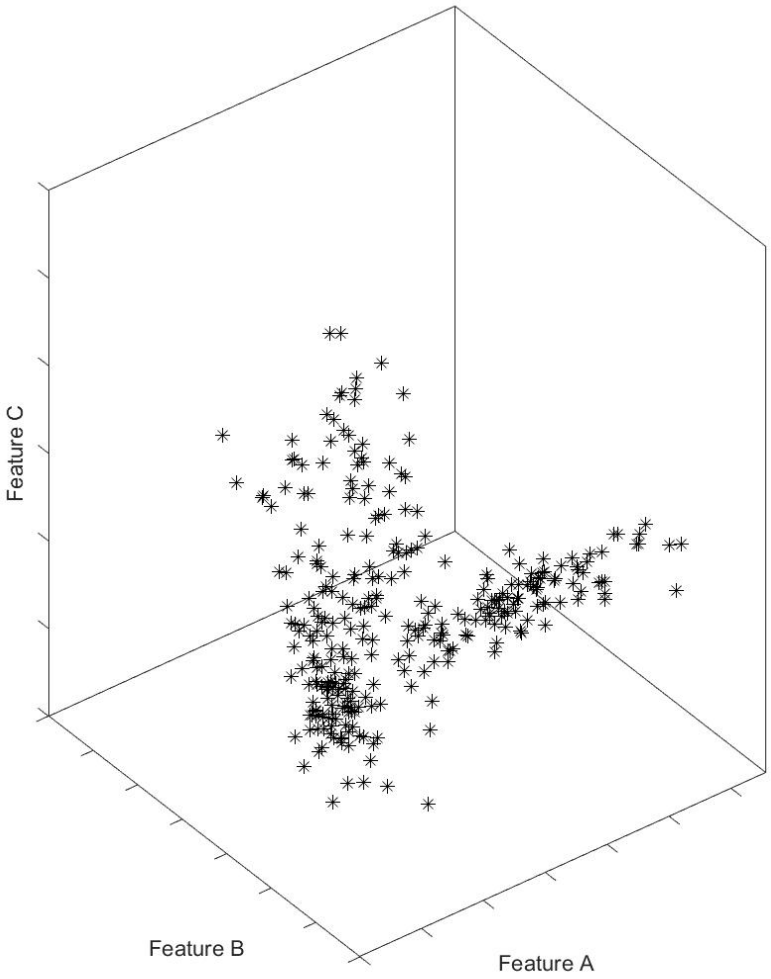
- training set: to train candidate models
- validation set: to evaluate candidate models and pick the best one
- test set: to do the final evaluation of the model

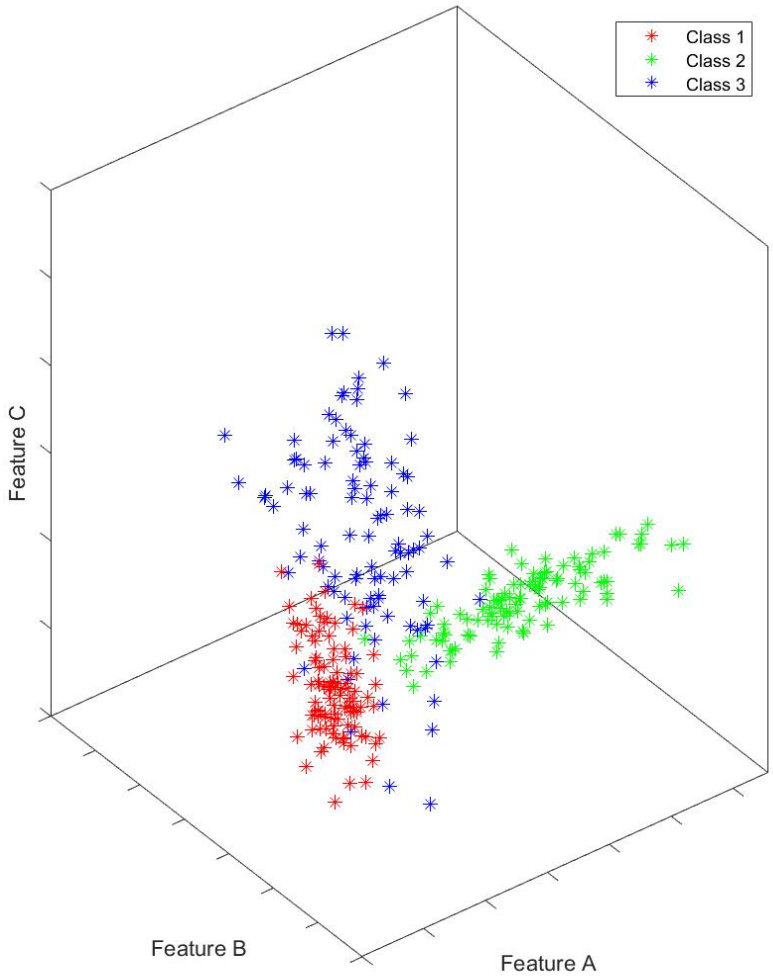
Supervised Learning: Classification



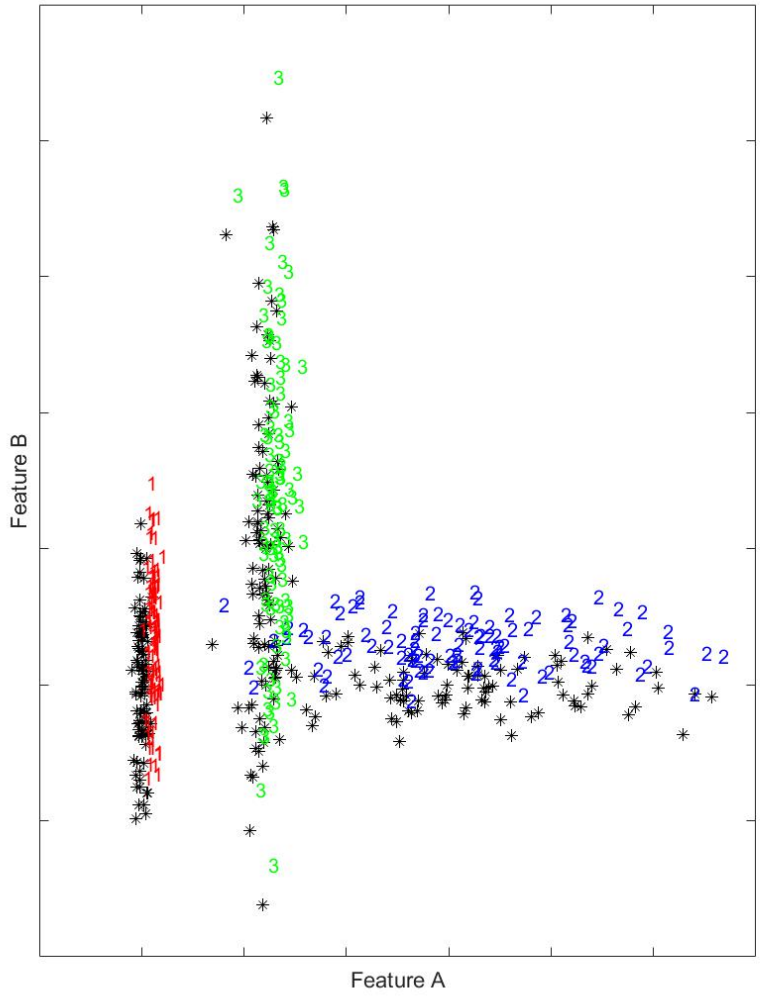


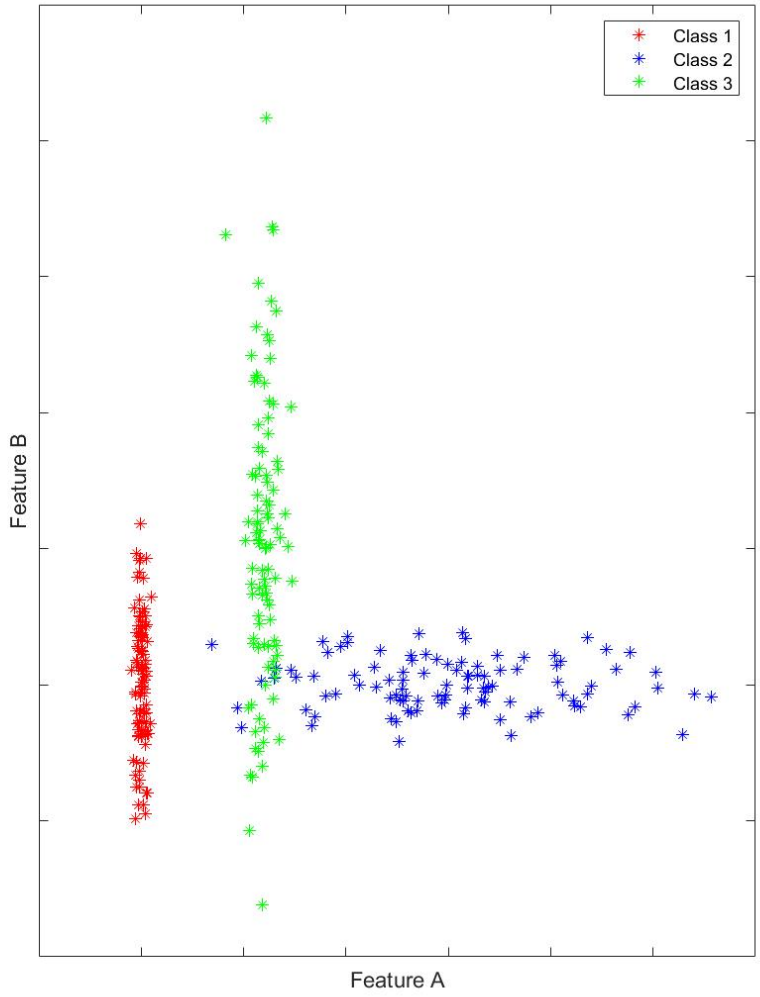
Supervised Learning: Classification



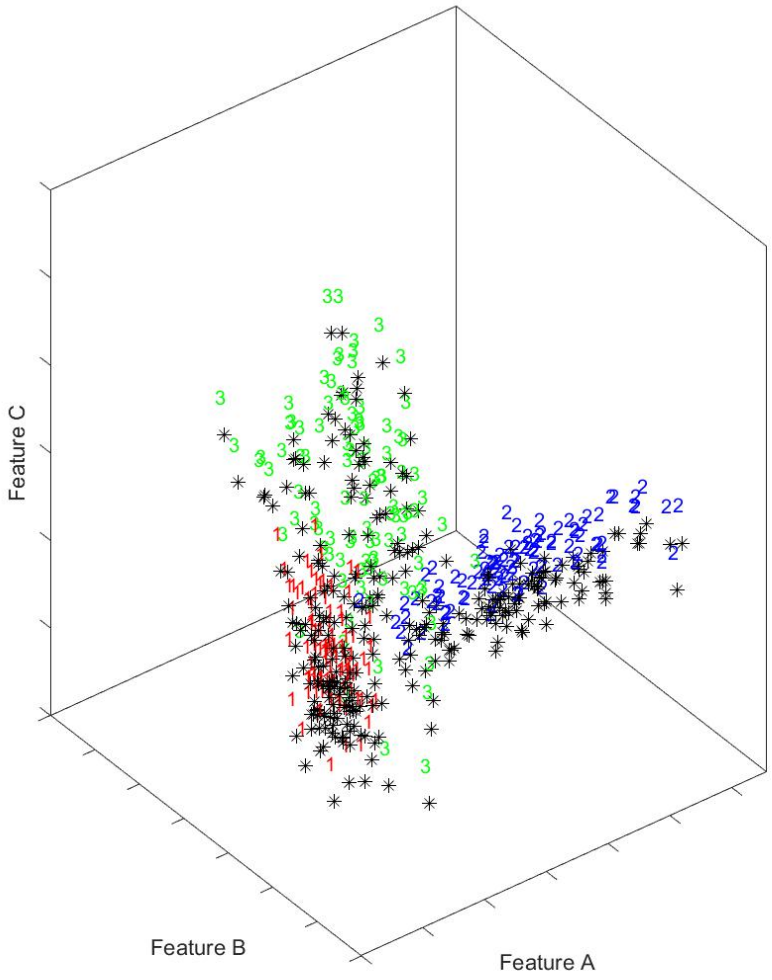


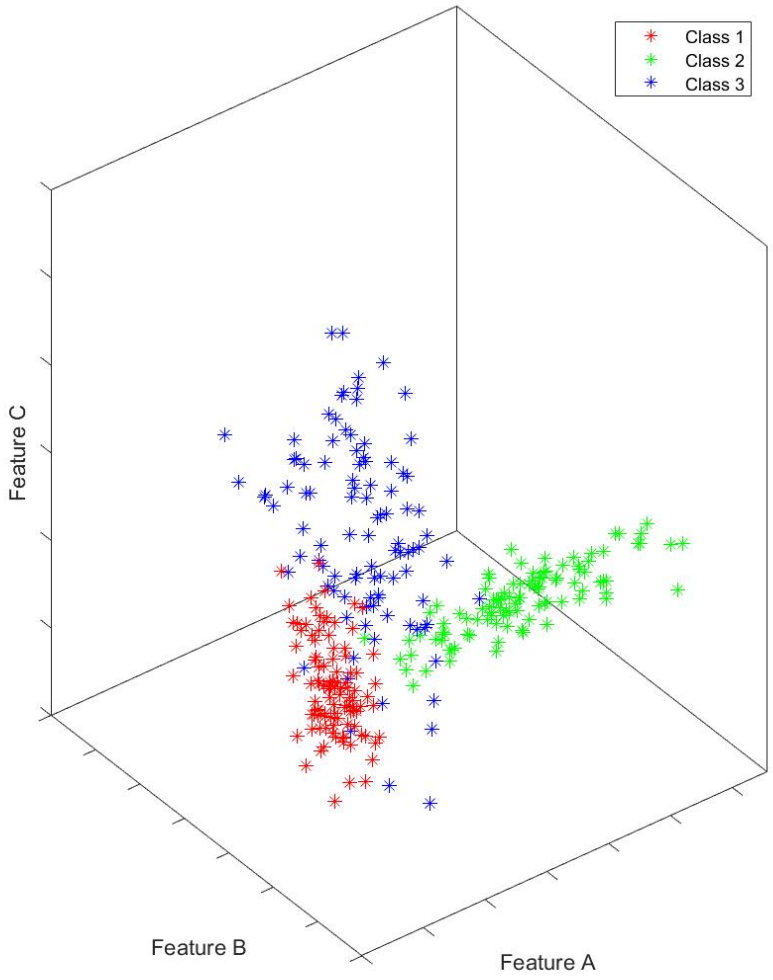
Data Set: Labeled Data





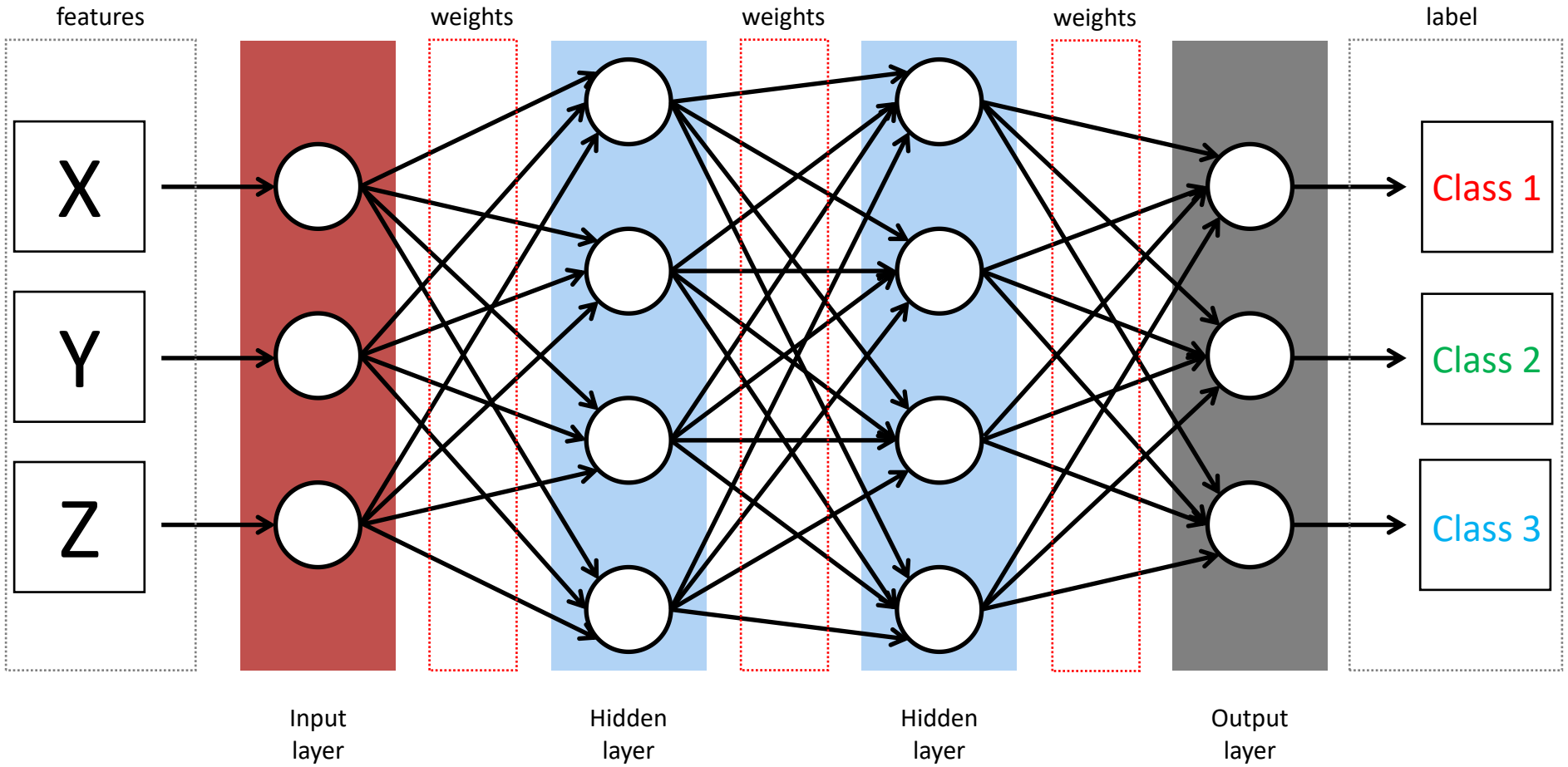
Data Set: Labeled Data





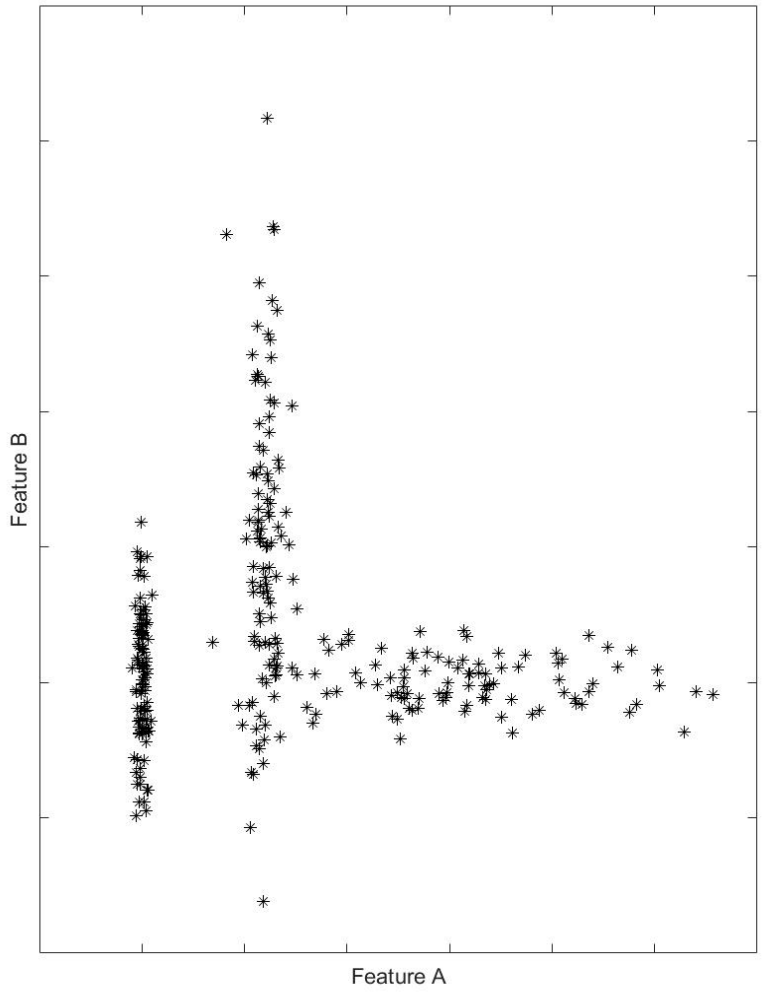
ANN: Supervised Learning

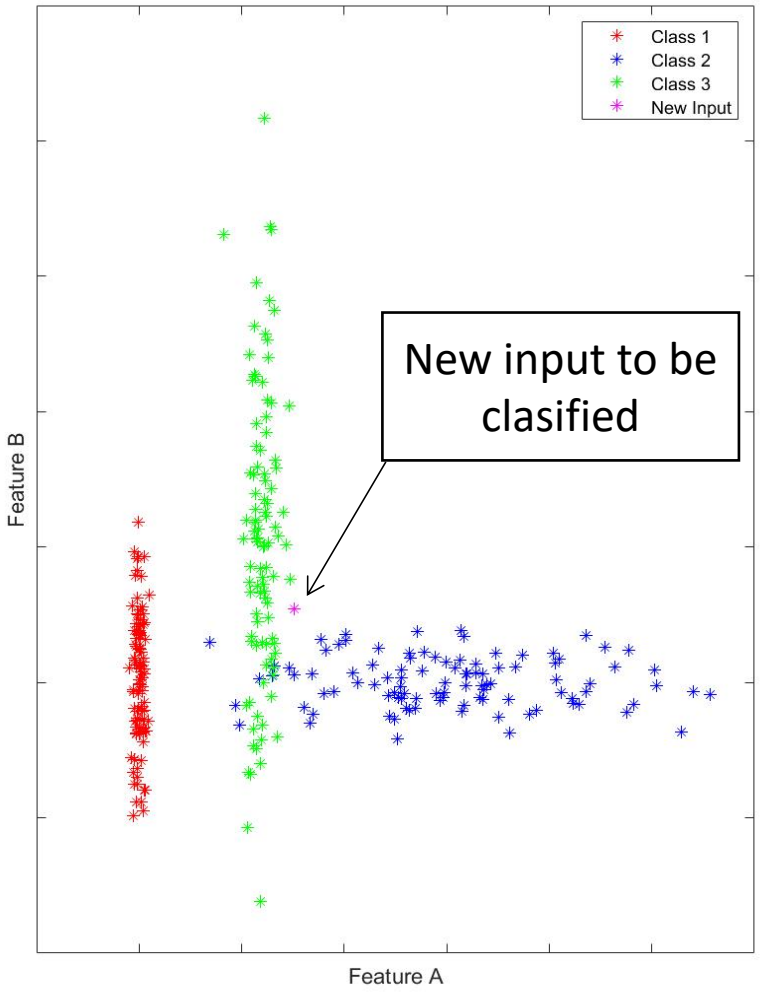
In order to work properly a classifier **needs to be trained** first with **labeled data**.



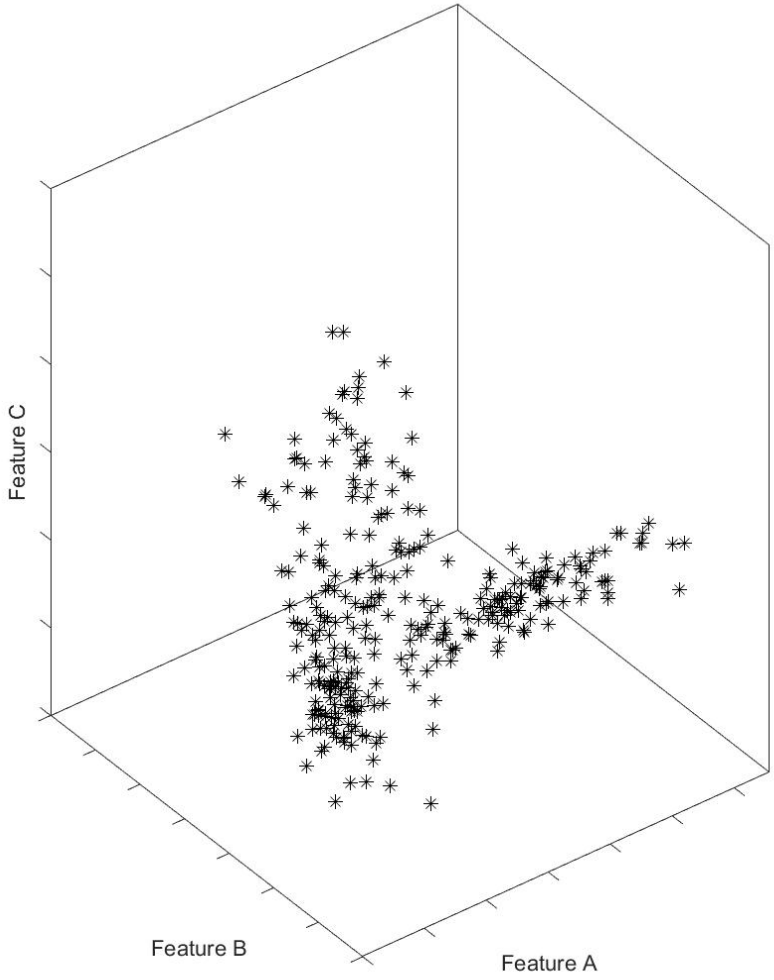
Training will **adjust all the weights** within this artificial neural network.

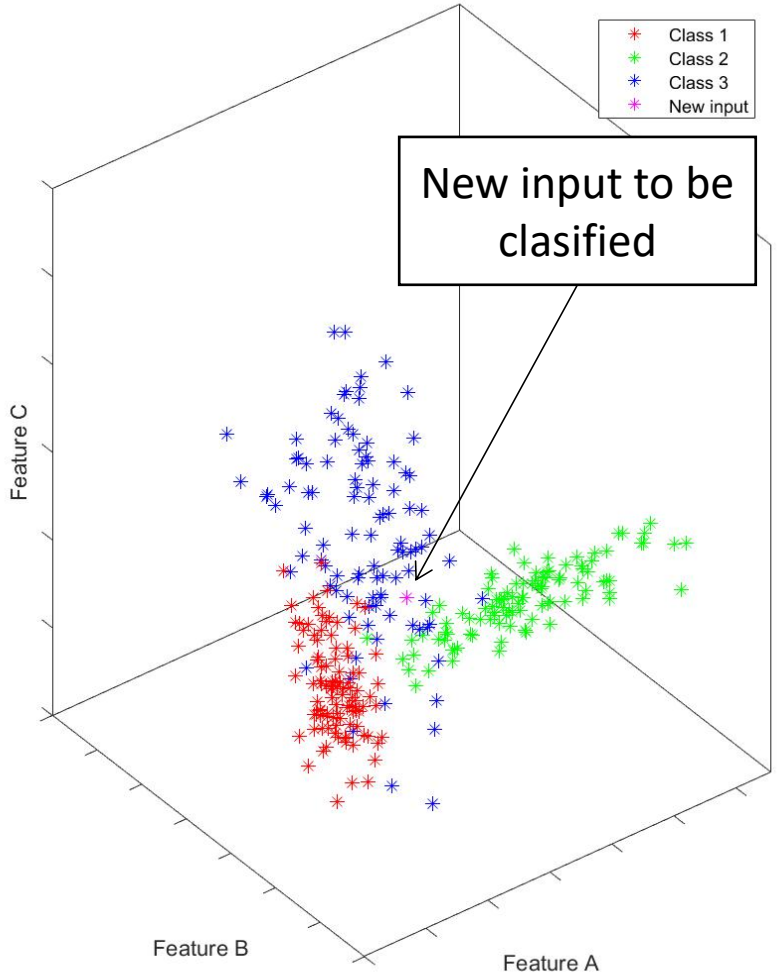
Supervised Learning: New Input



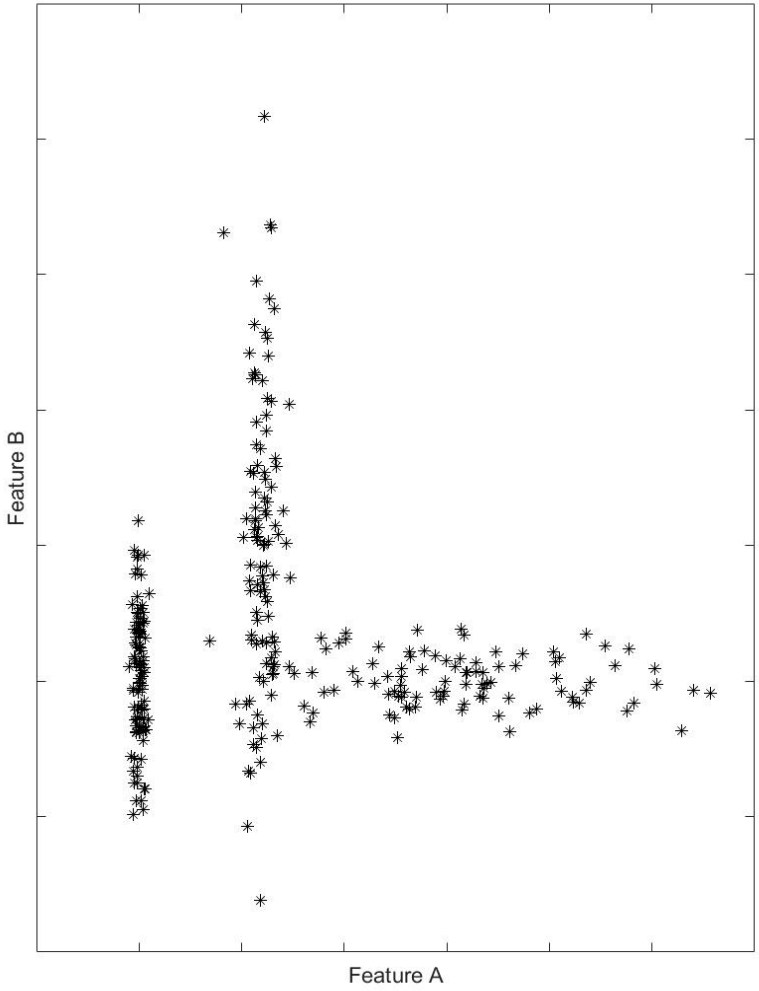


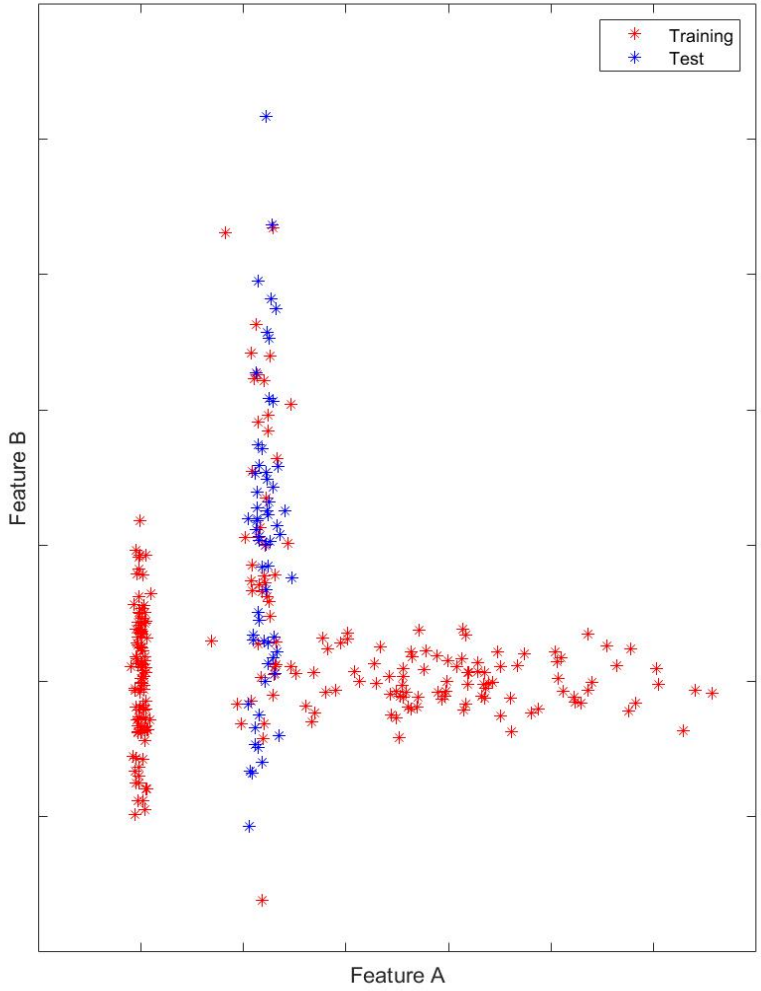
Supervised Learning: New Input



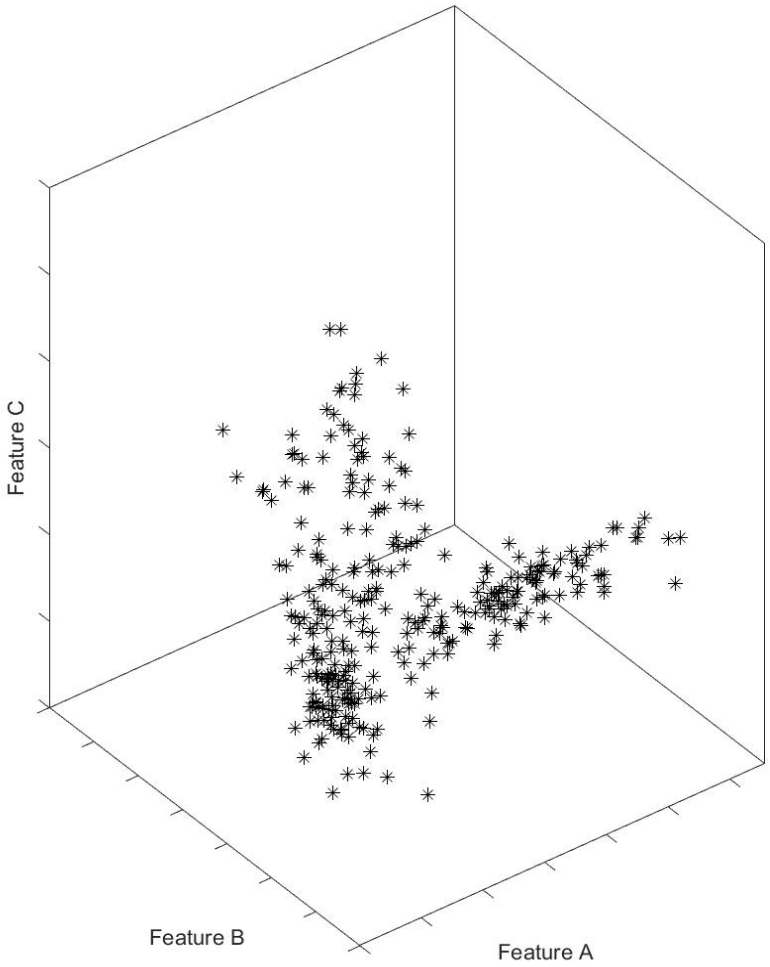


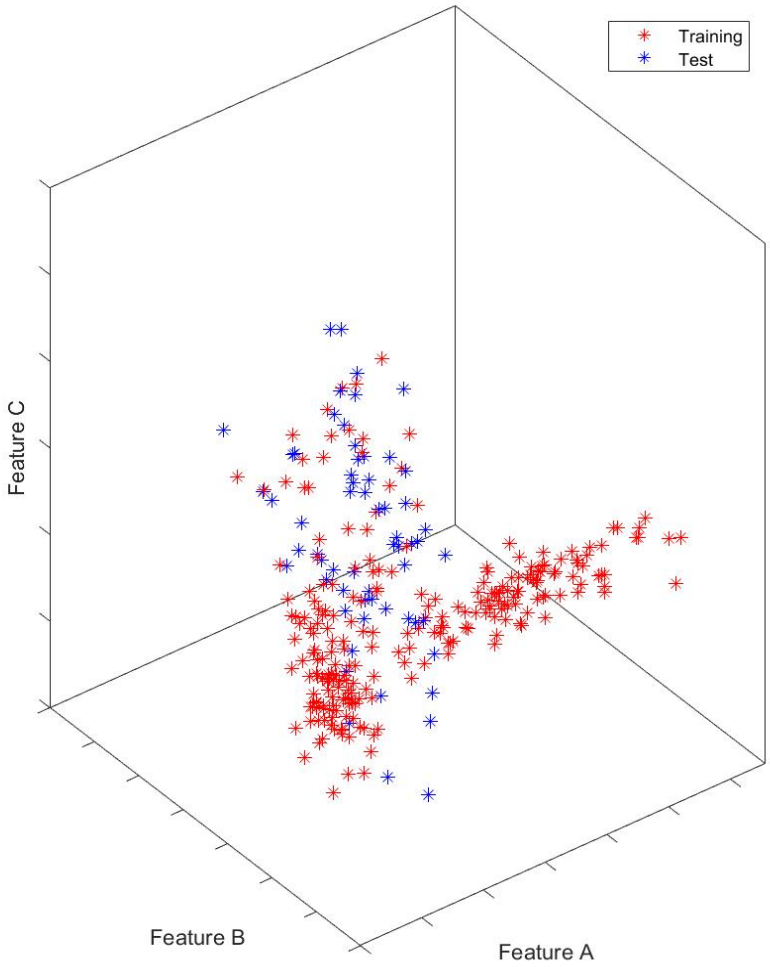
Training / Test Sets: Poor Split



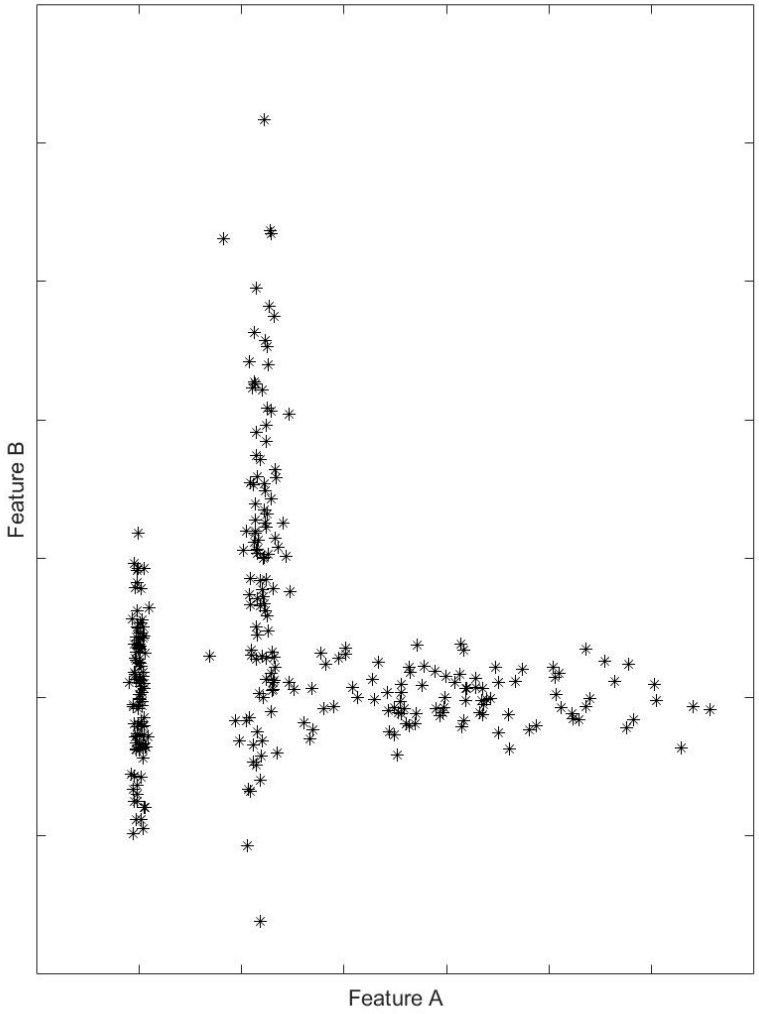


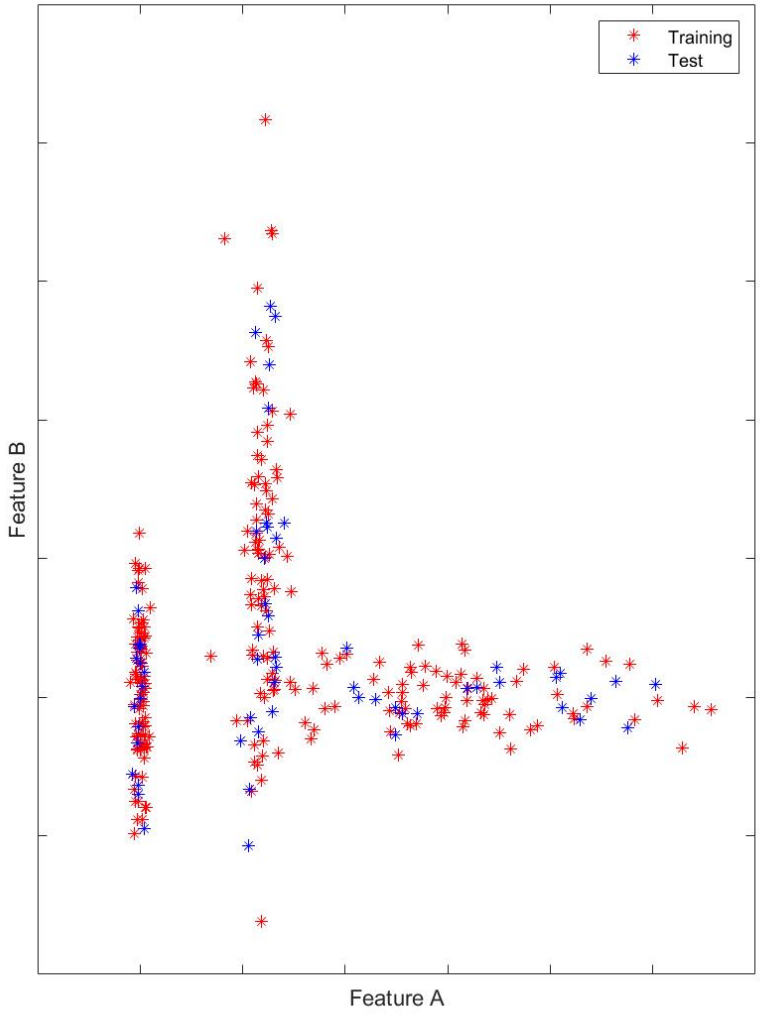
Training / Test Sets: Poor Split



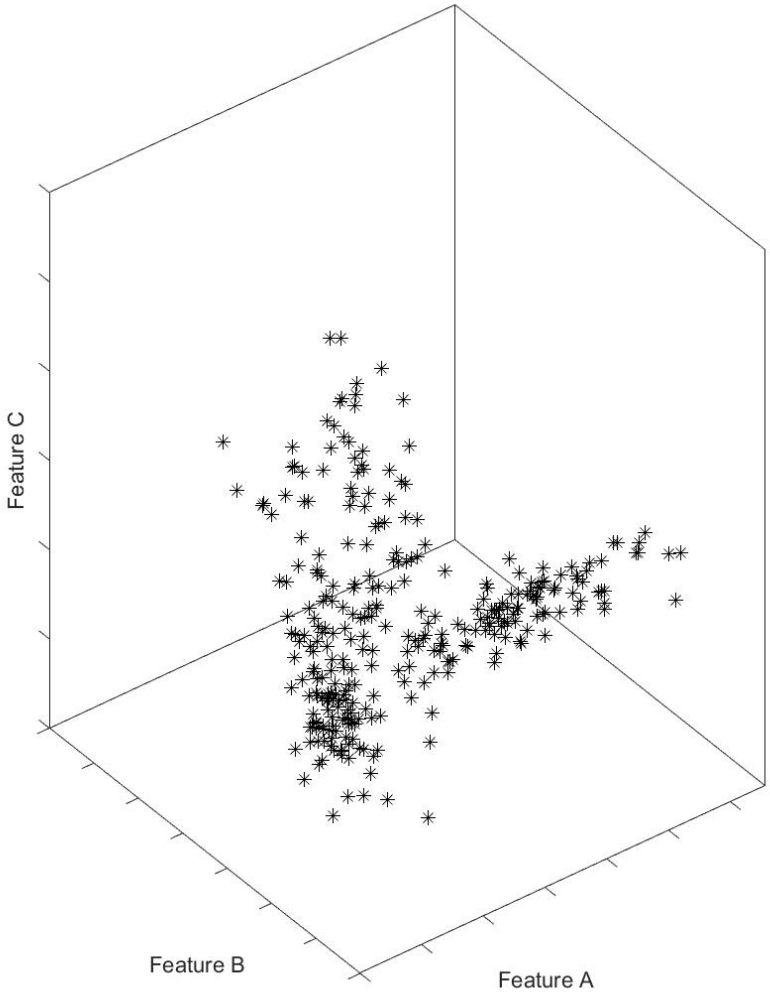


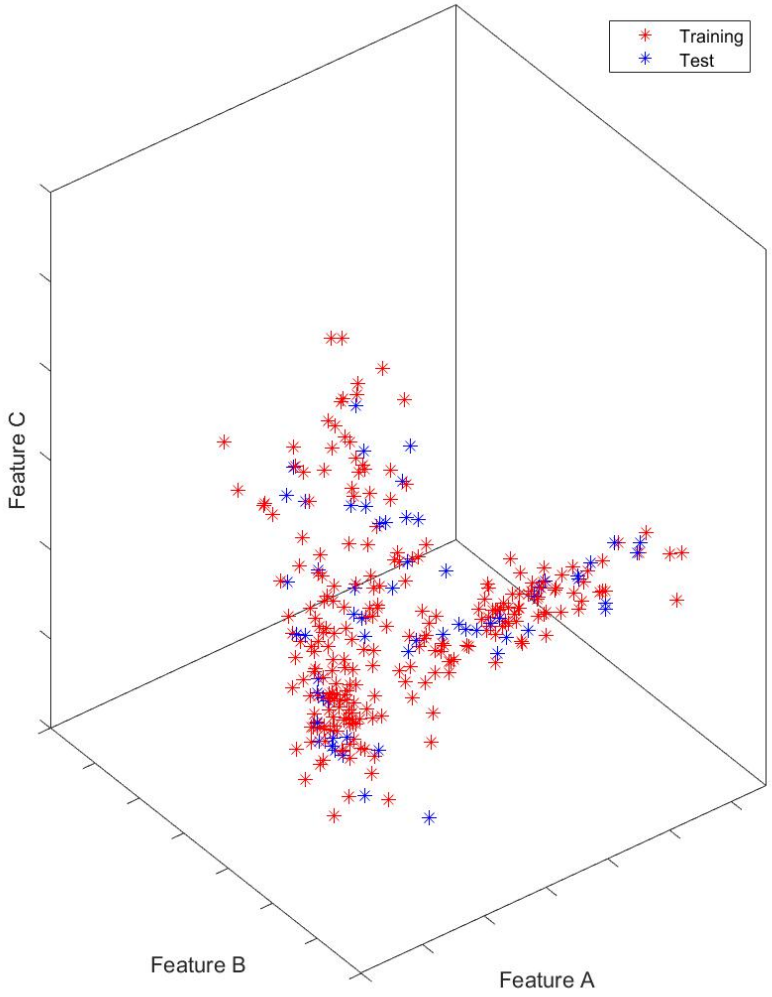
Training / Test Sets: Better Split



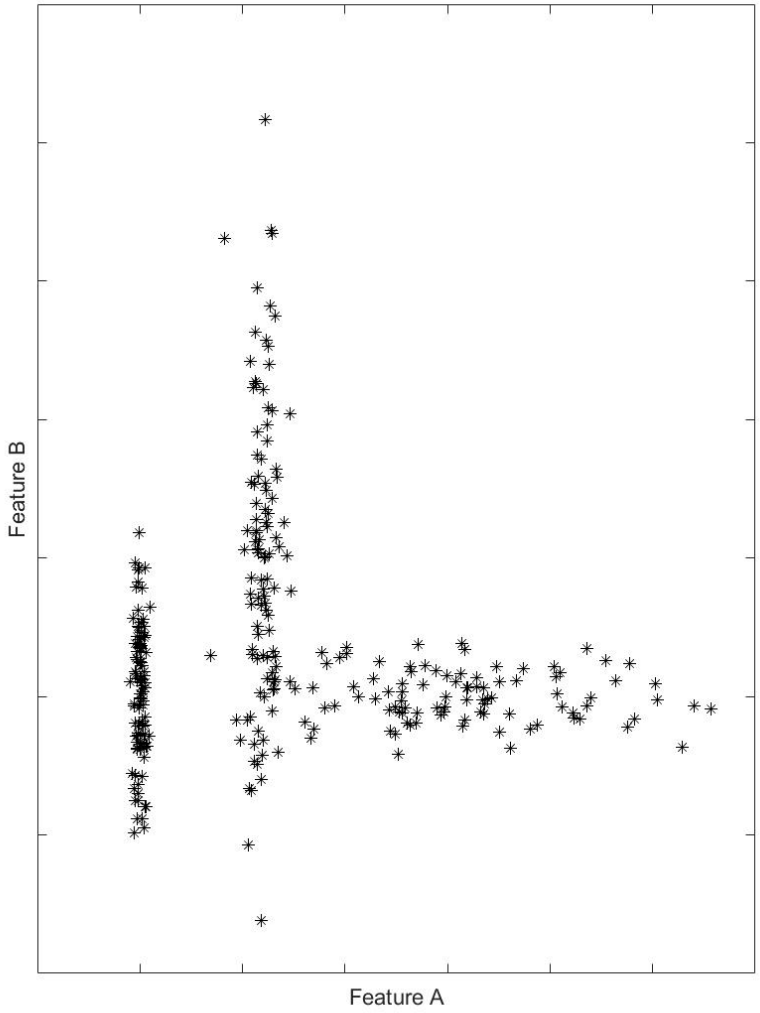


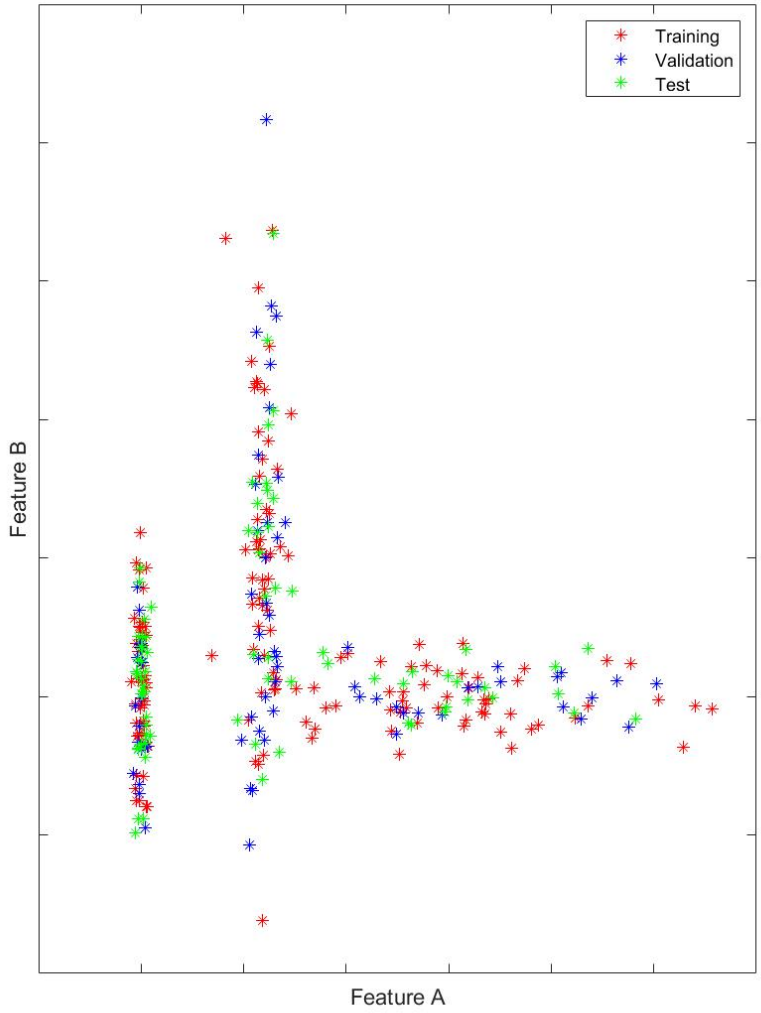
Training / Test Sets: Better Split



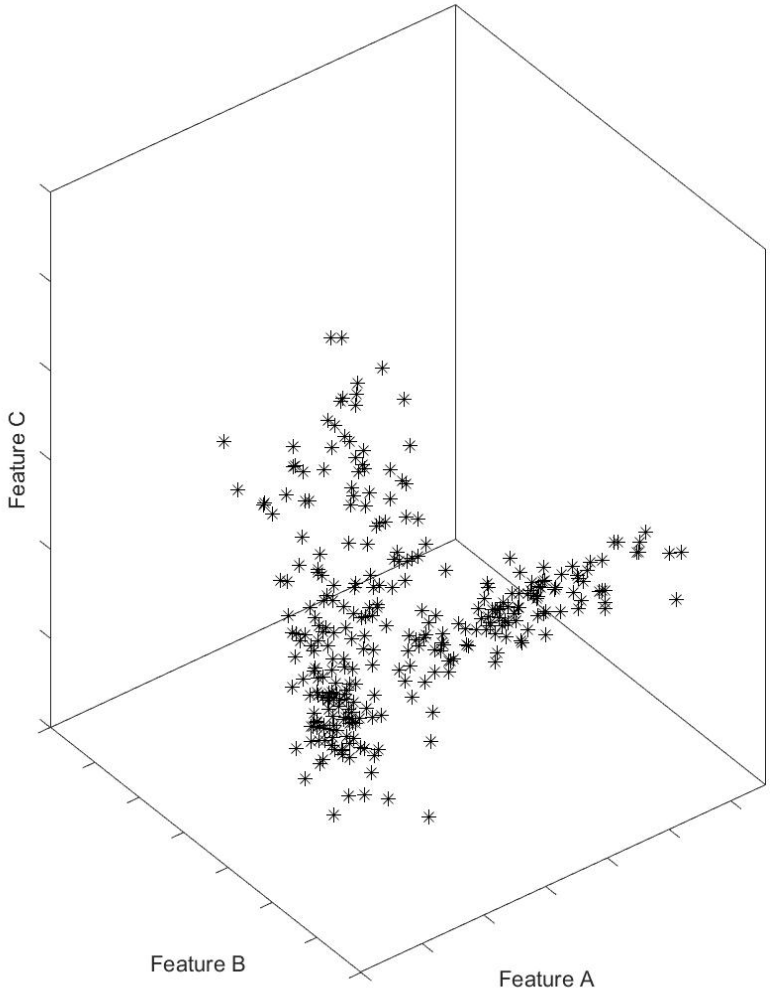


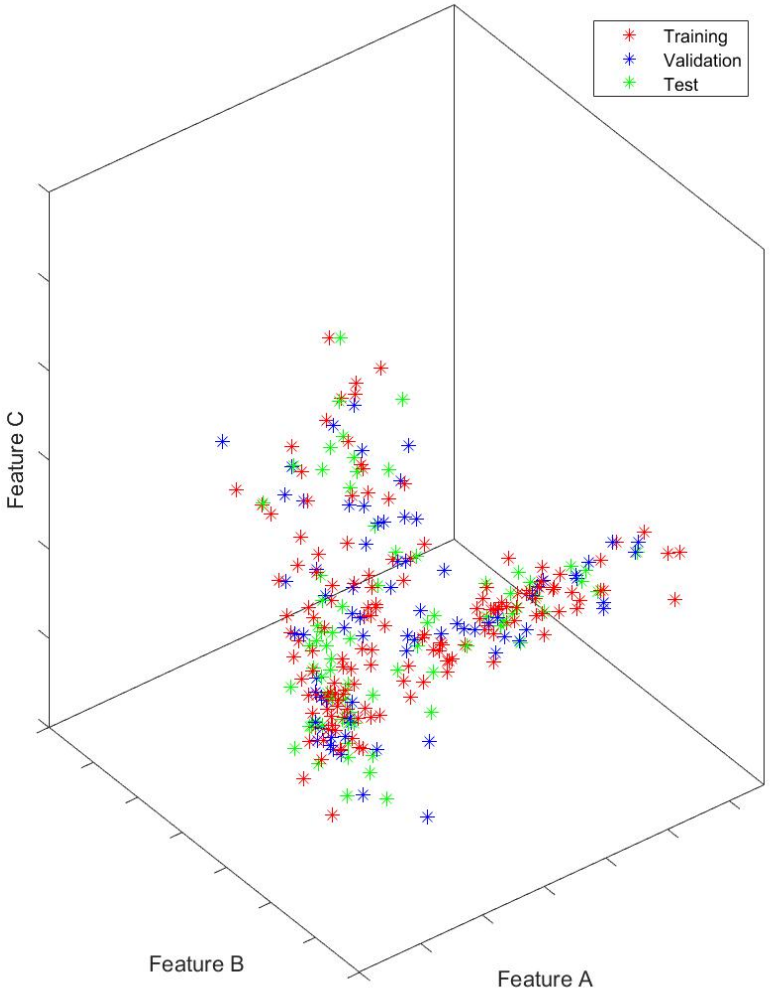
Training / Validation / Test Sets



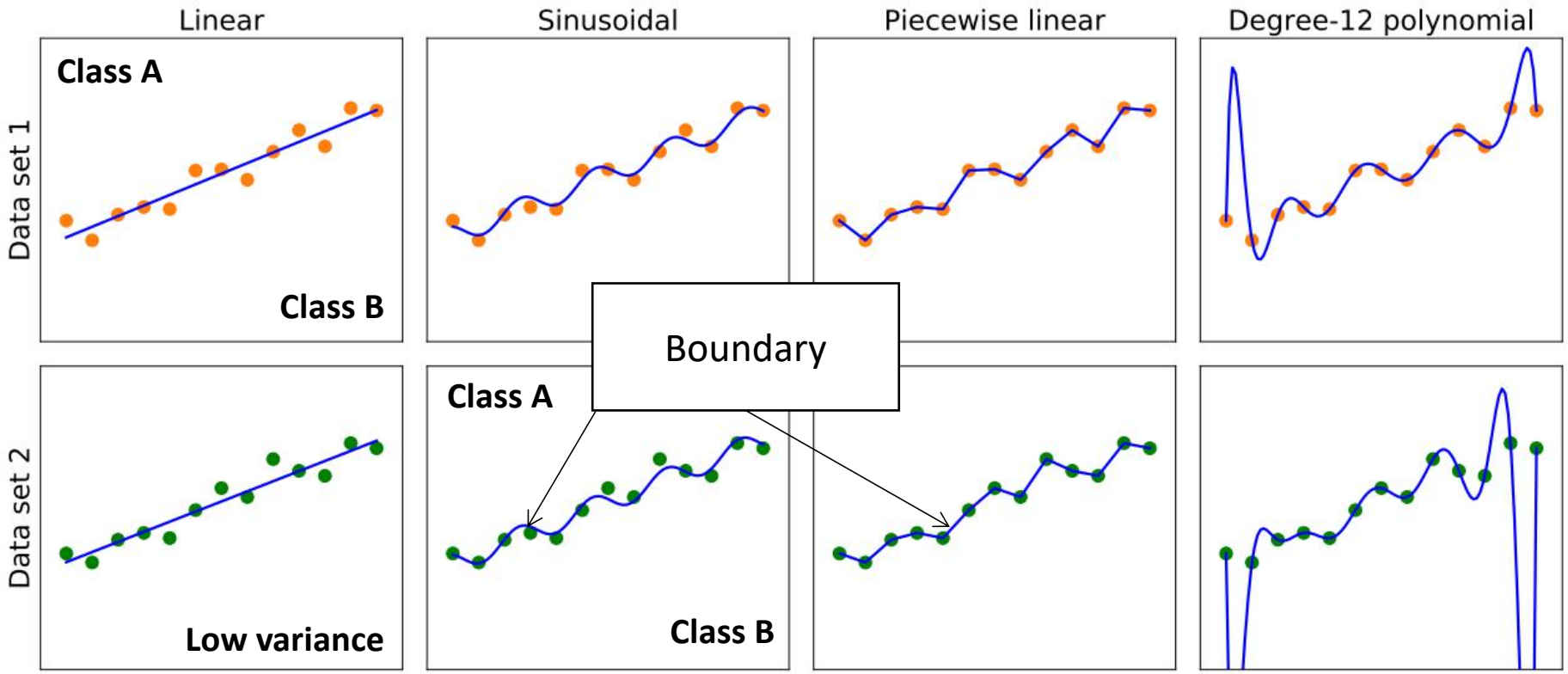


Training / Validation / Test Sets

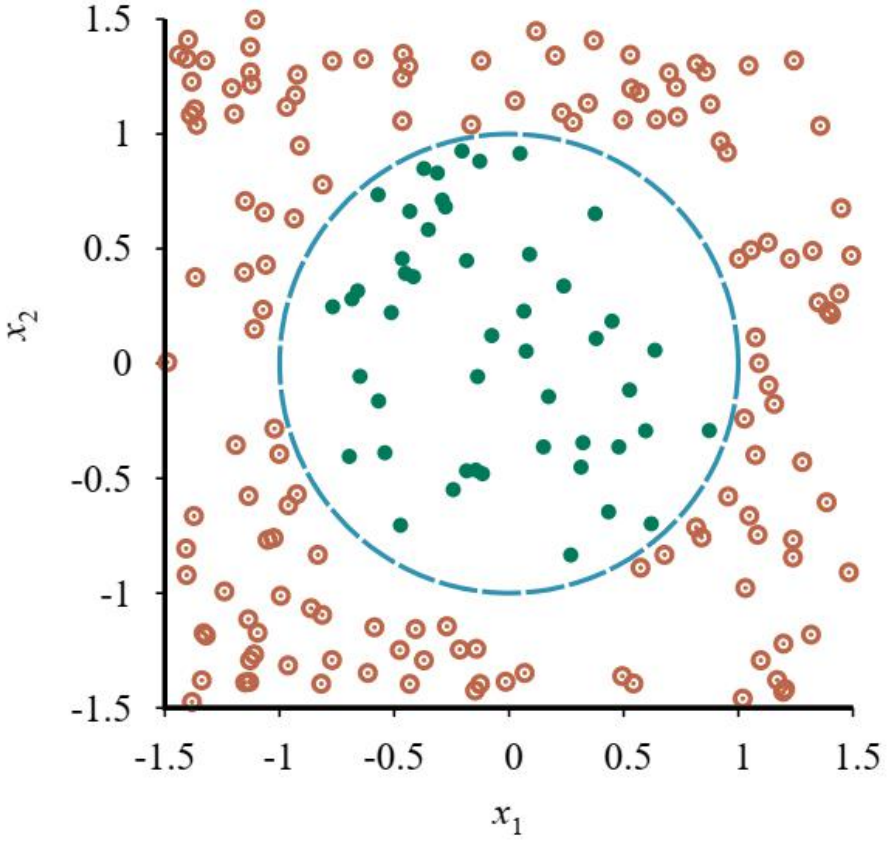


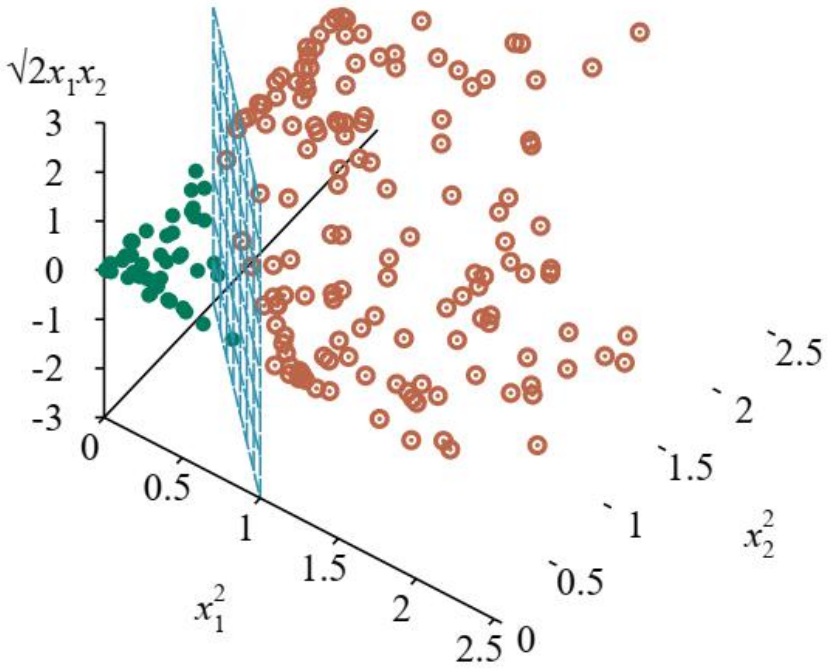


Hypothesis: Decision "Boundary"



Hypothesis: Classification "Boundary"





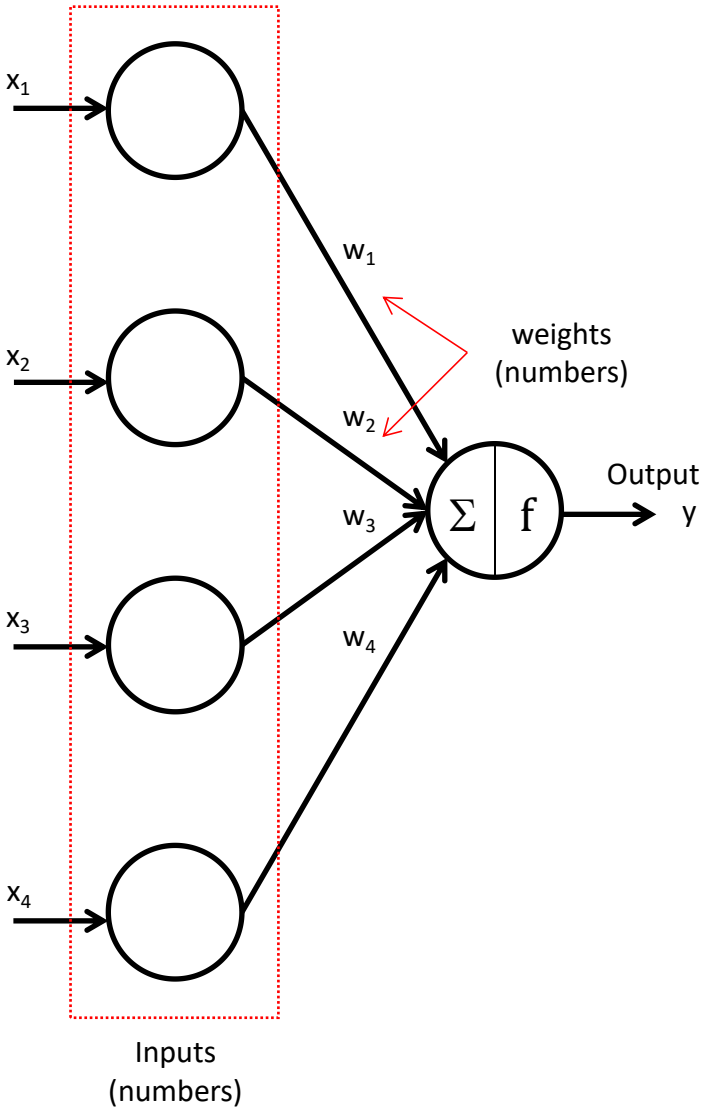
Classifier Evaluation: Confusion Matrix

		Predicted class: Positive	Predicted class: Negative	
Actual class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity $\frac{TP}{TP+FN}$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{TN+FP}$
		Precision $\frac{TP}{TP+FP}$	Negative Predictive Value $\frac{TN}{TN+FN}$	Accuracy $\frac{TP+TN}{TP+TN+FP+FN}$

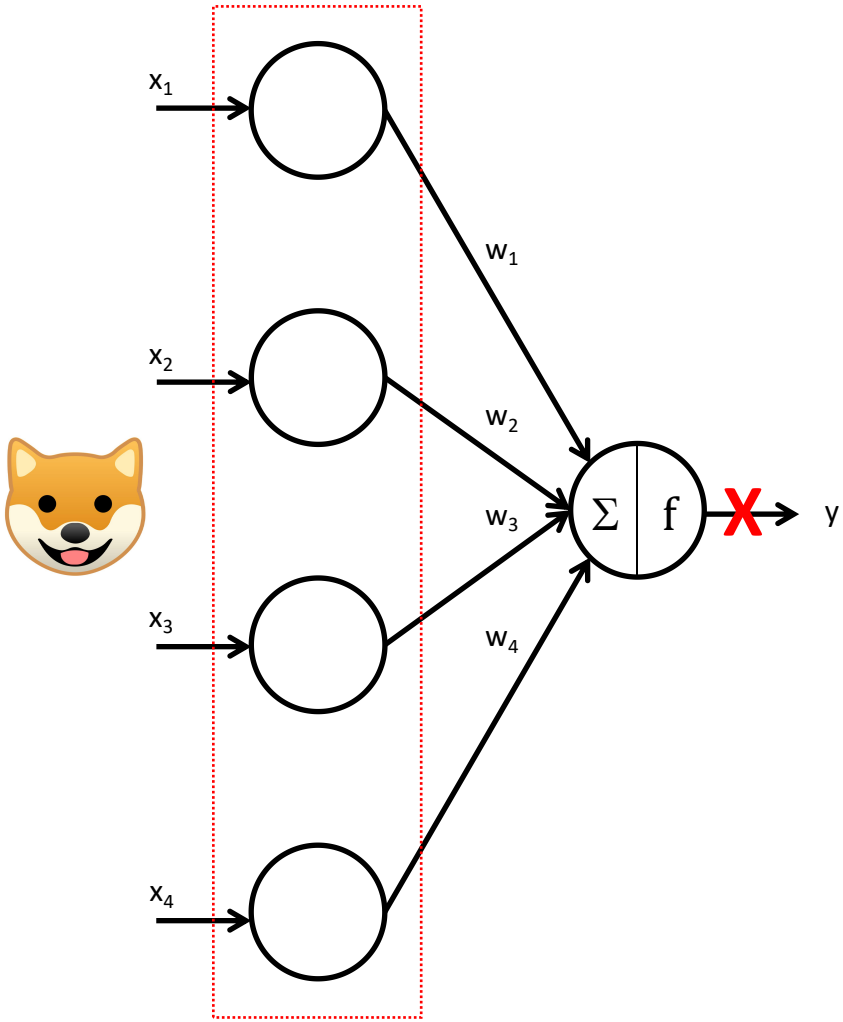
Artificial Neuron (Perceptron)

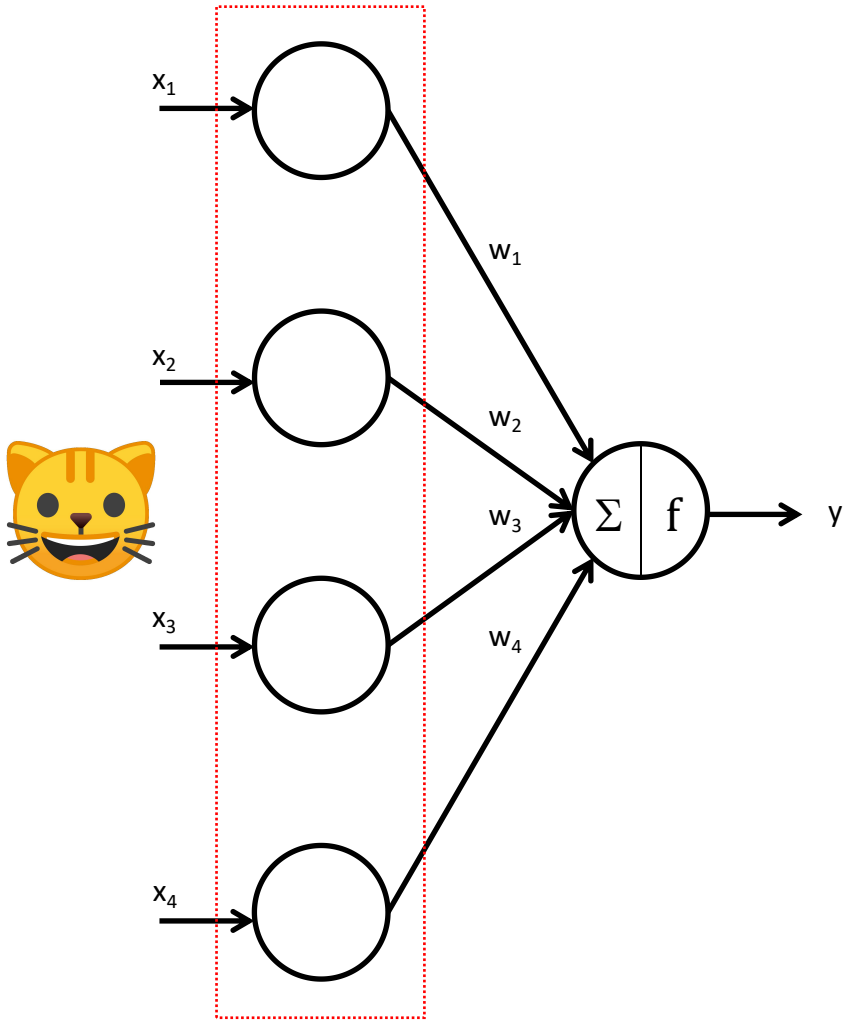
A (**single-layer**) **perceptron** is a model of a biological neuron. It is made of the following components:

- inputs x_i - numerical values representing information
- weights w_i - numerical values representing how "important" corresponding input is
- weighted sum: $\Sigma\ w_i * x_i$
- activation function f that decides if the neuron "fires"



Single-layer Perceptron as a Classifier

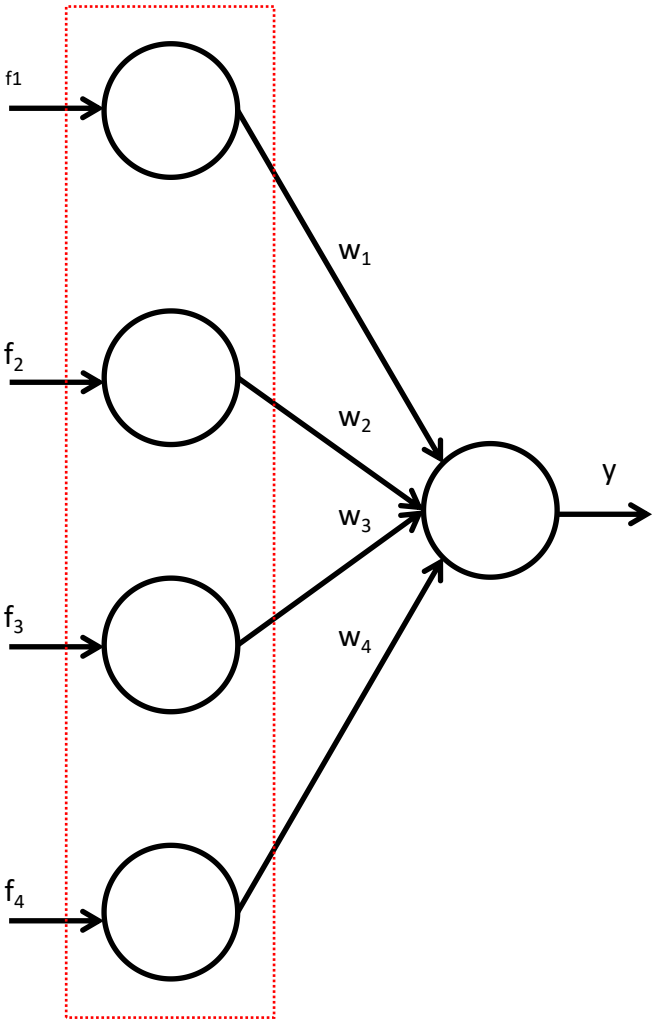


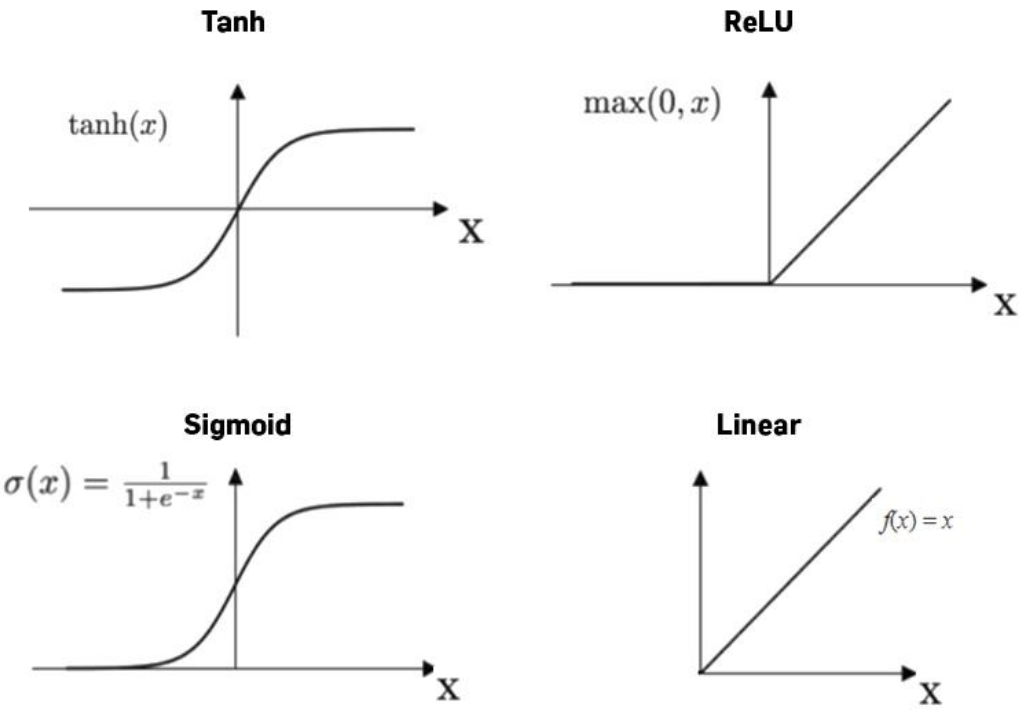


$\Sigma\, w_i * x_i < 0 \rightarrow f = 0$ → **NOT CAT**

$\Sigma\, w_i * x_i \geq 0 \rightarrow f = 1$ → **CAT**

Selected Activation Functions

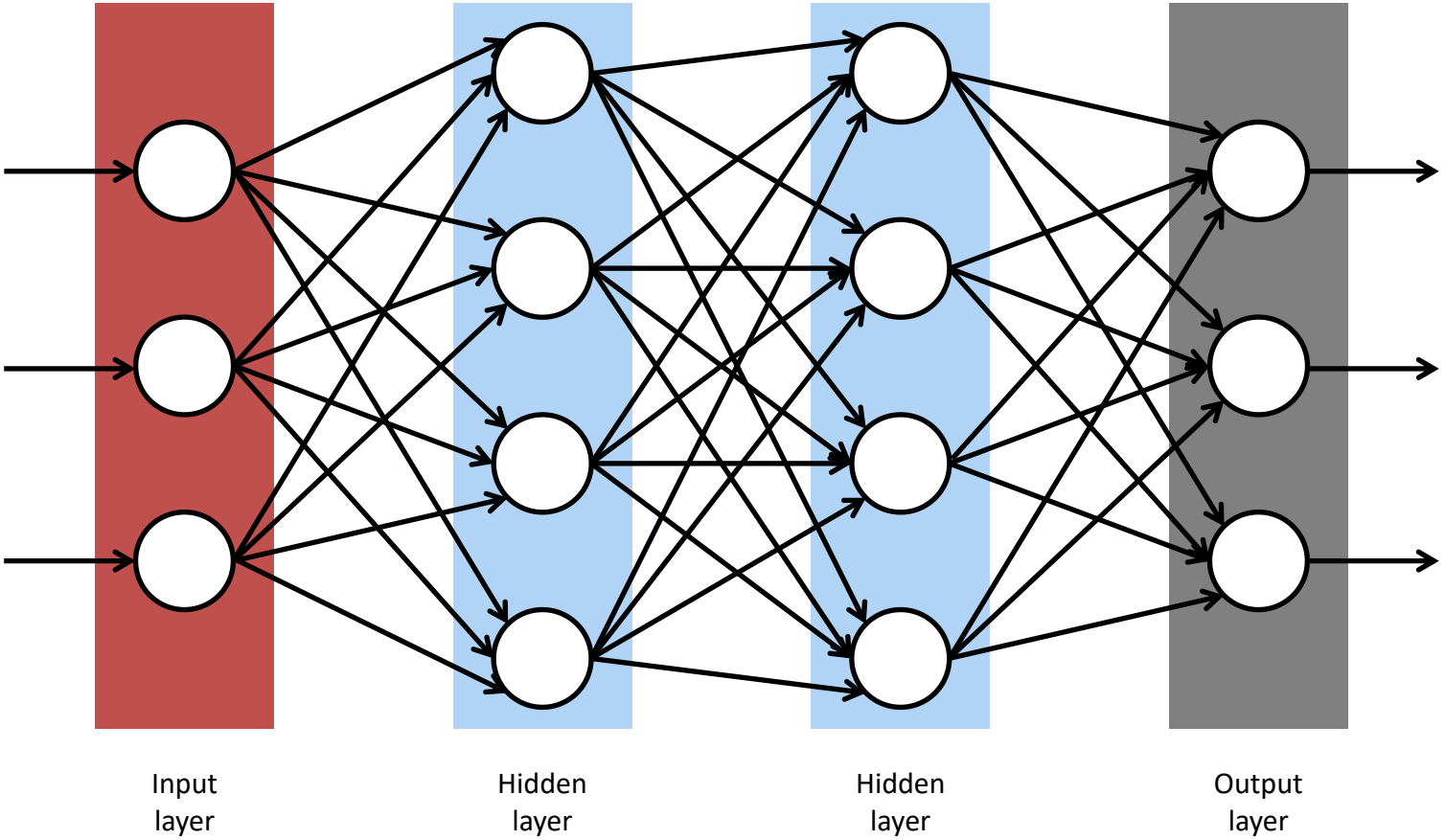




ReLU: Rectified Linear Unit

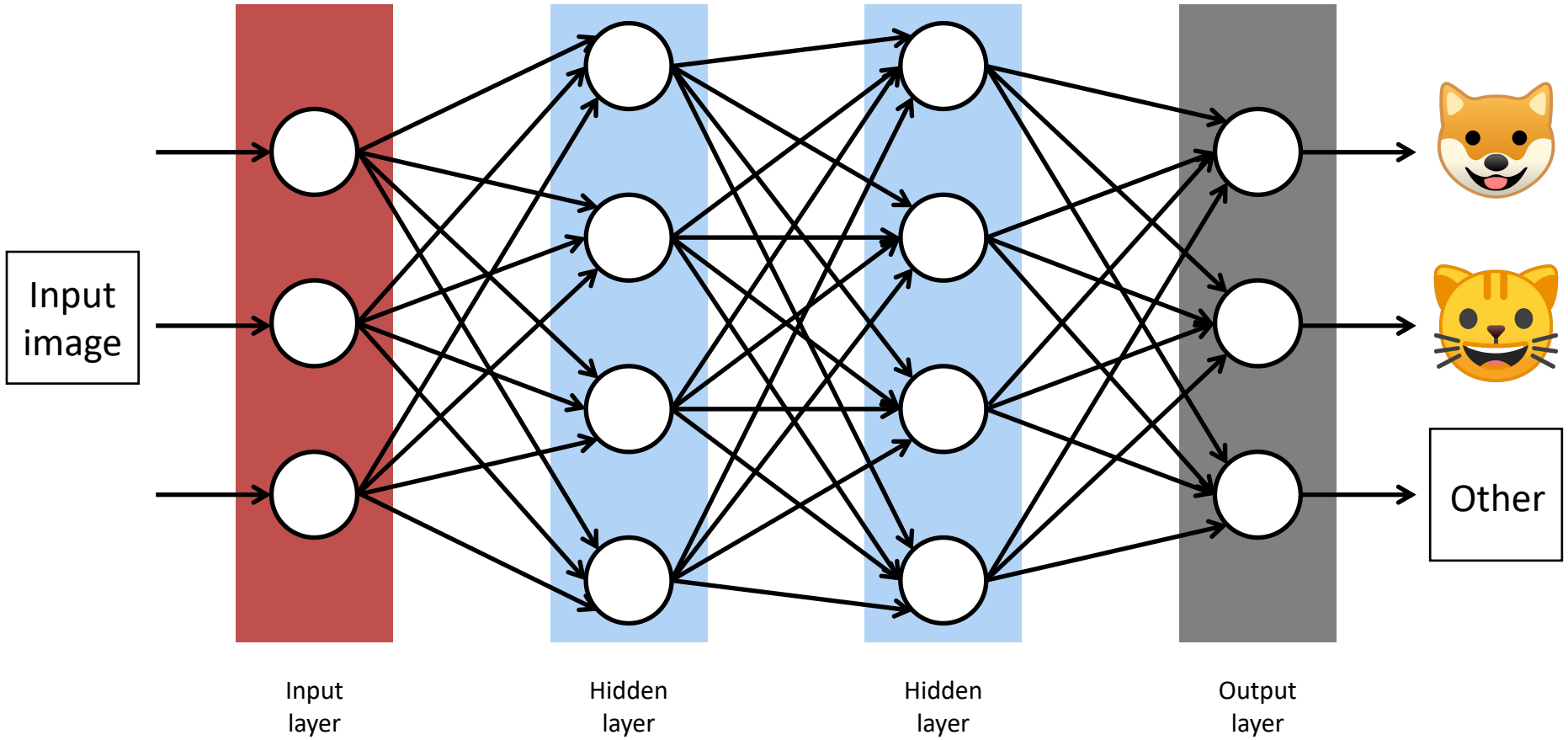
Artificial Neural Network (ANN)

An artificial neural network is made of **multiple artificial neuron layers**.

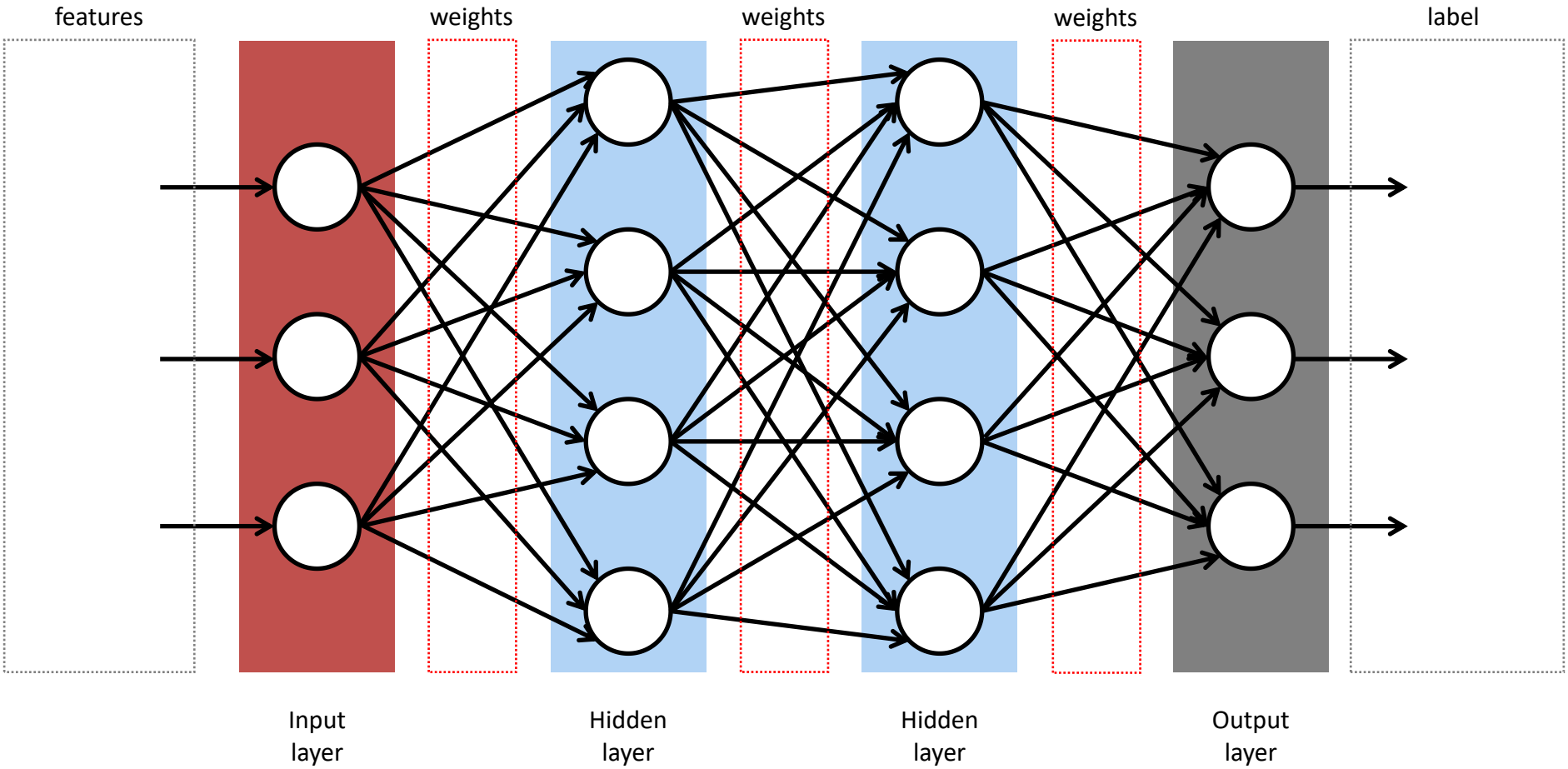


ANN as an Image Classifier

An artificial neural network can be used as a **classifier** as well.

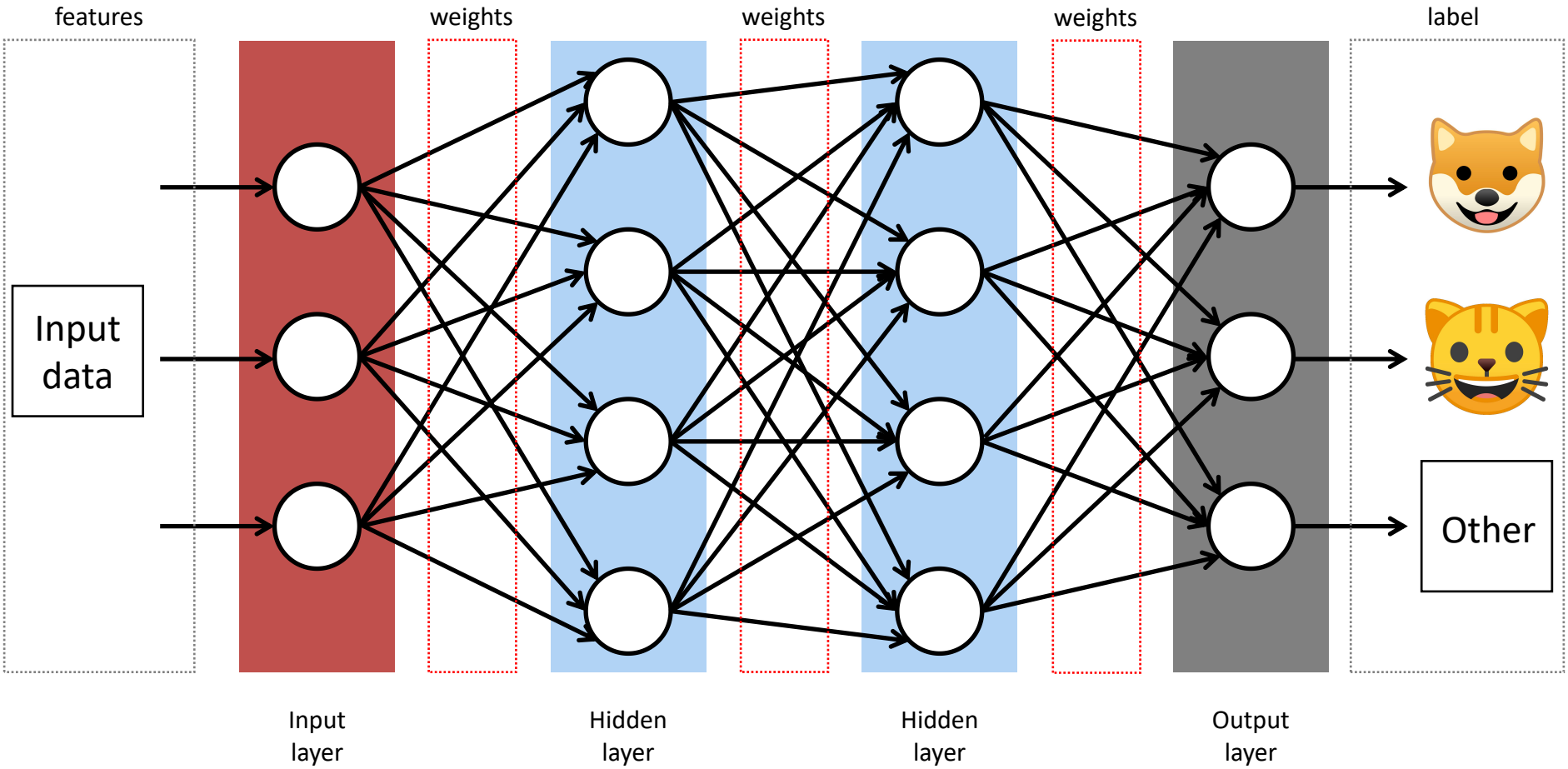


ANN as a Classifier



ANN: Supervised Learning

In order to work properly a classifier **needs to be trained** first with **labeled data**.



Training will **adjust all the weights** within this artificial neural network.

Training Data: Features + Labels

Typically input data will be represented by a **limited set of features**.

Features:
Wheels: 4
Weight: 8 tons
Passengers: 1

Label:
Truck

Features:
Wheels: 6
Weight: 8 tons
Passengers: 1

Label:
Truck

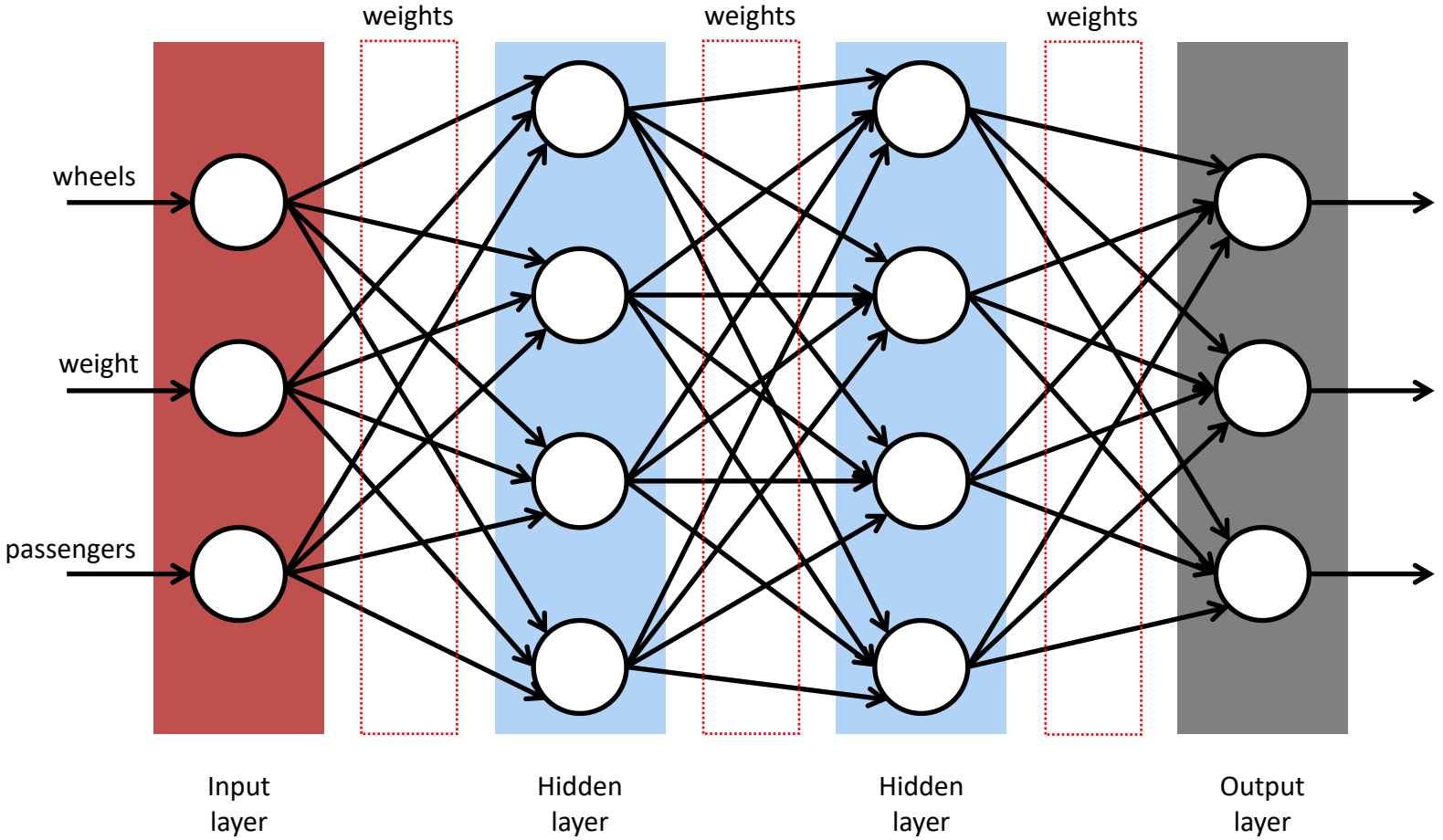
Features:
Wheels: 4
Weight: 1 ton
Passengers: 4

Label:
Car

Features:
Wheels: 4
Weight: 2 tons
Passengers: 4

Label:
Car

ANN: Supervised Learning



Training Data: Images + Labels

A classifier **needs to be "shown" thousands of labeled examples to learn**.

Label: BUS

Label: CAR

Label: BRIDGE

Label: PALM

Label: TRAFFIC LIGHT

Label: TAXI

Label: CROSSWALK

Label: CHIMNEY

Label: MOTORCYCLE

Label: STREET SIGN

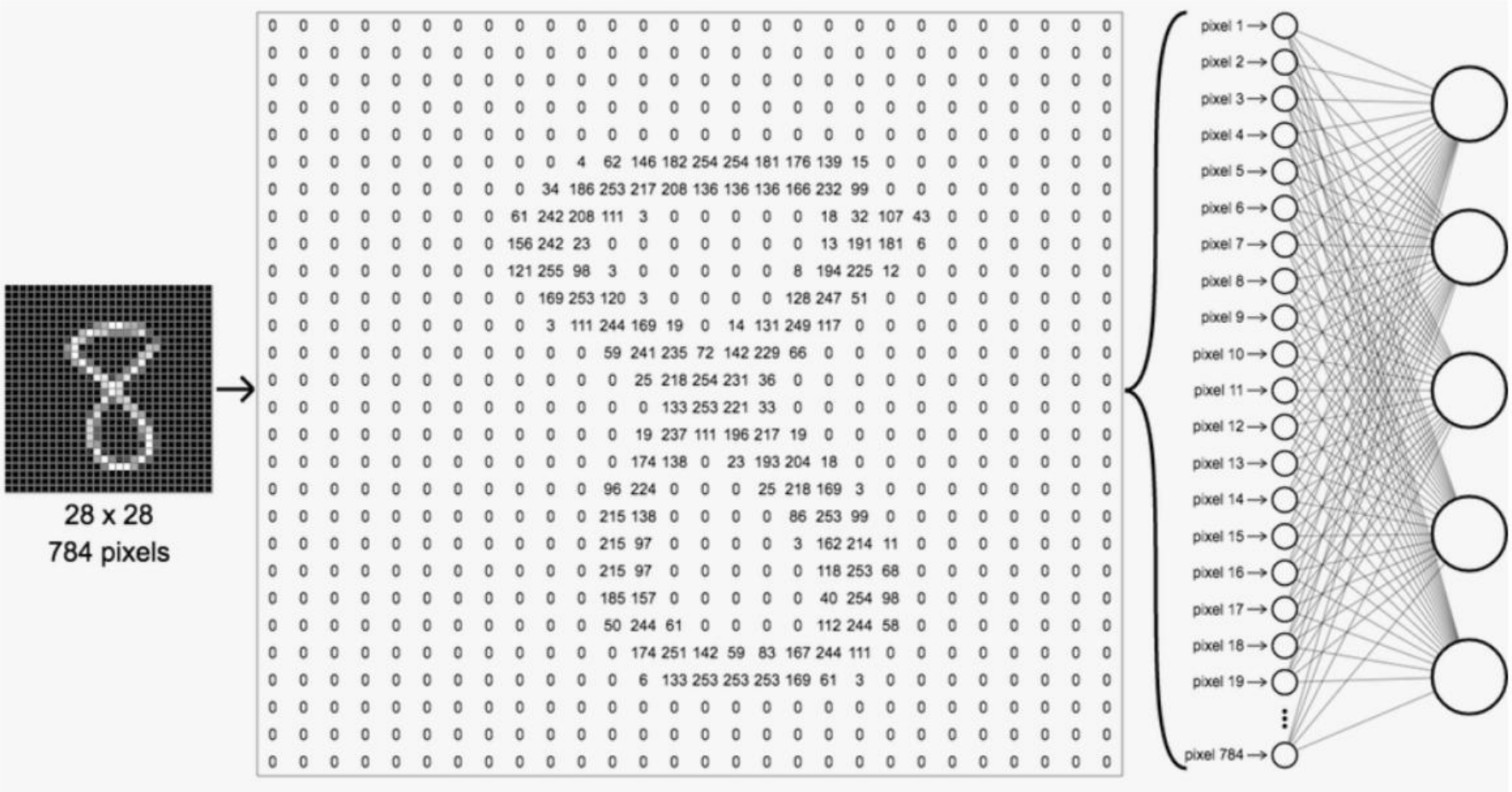
Label: HYDRANT

Label: BICYCLE

Note how some images are "incomplete" and "flawed".

Digit Image as ANN Feature Set

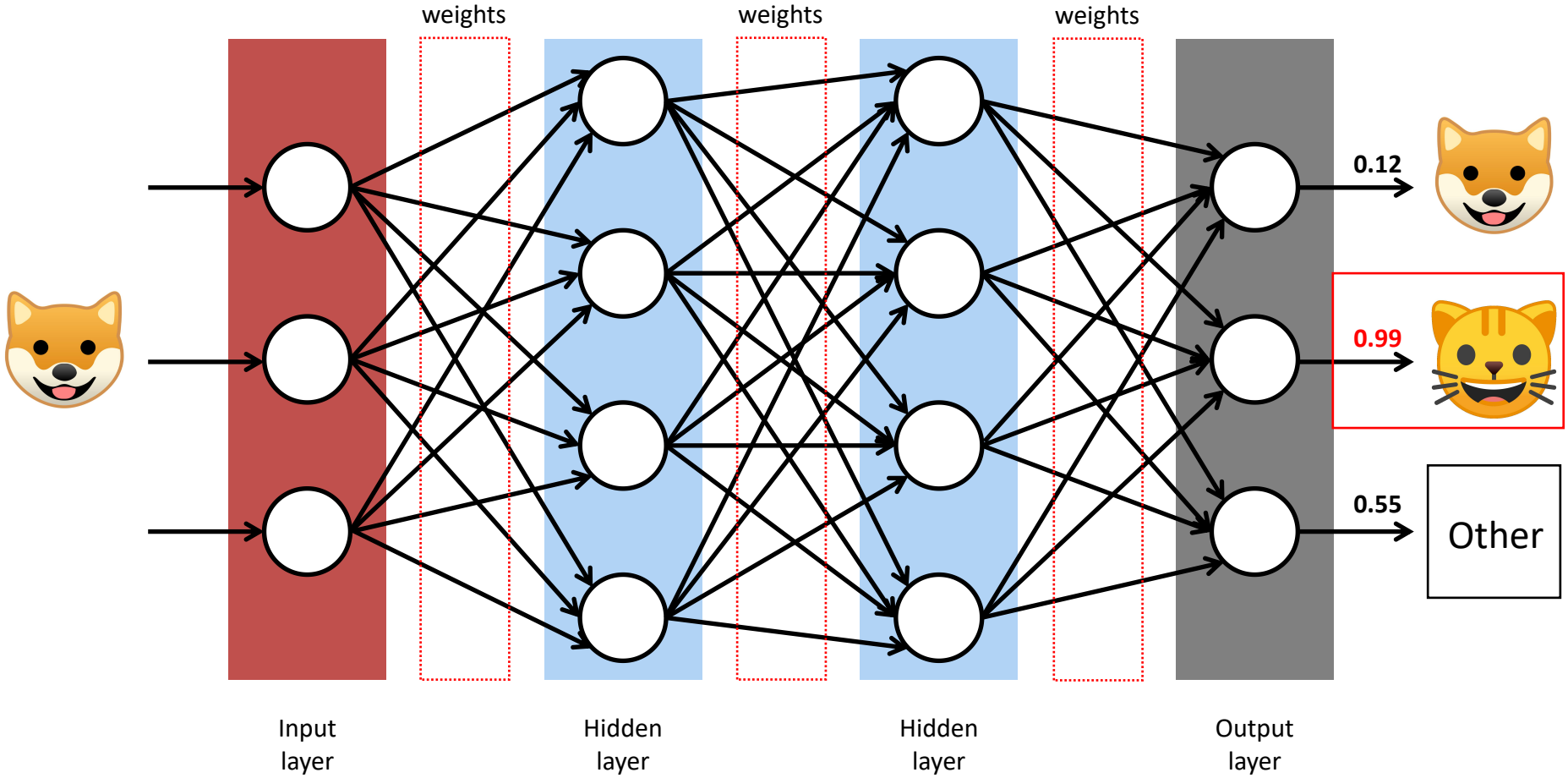
Individual features need to be “extracted” from an image. An image is numbers.



Source: https://nikolanews.com/not-just-introduction-to-convolutional-neural-networks-part-1/

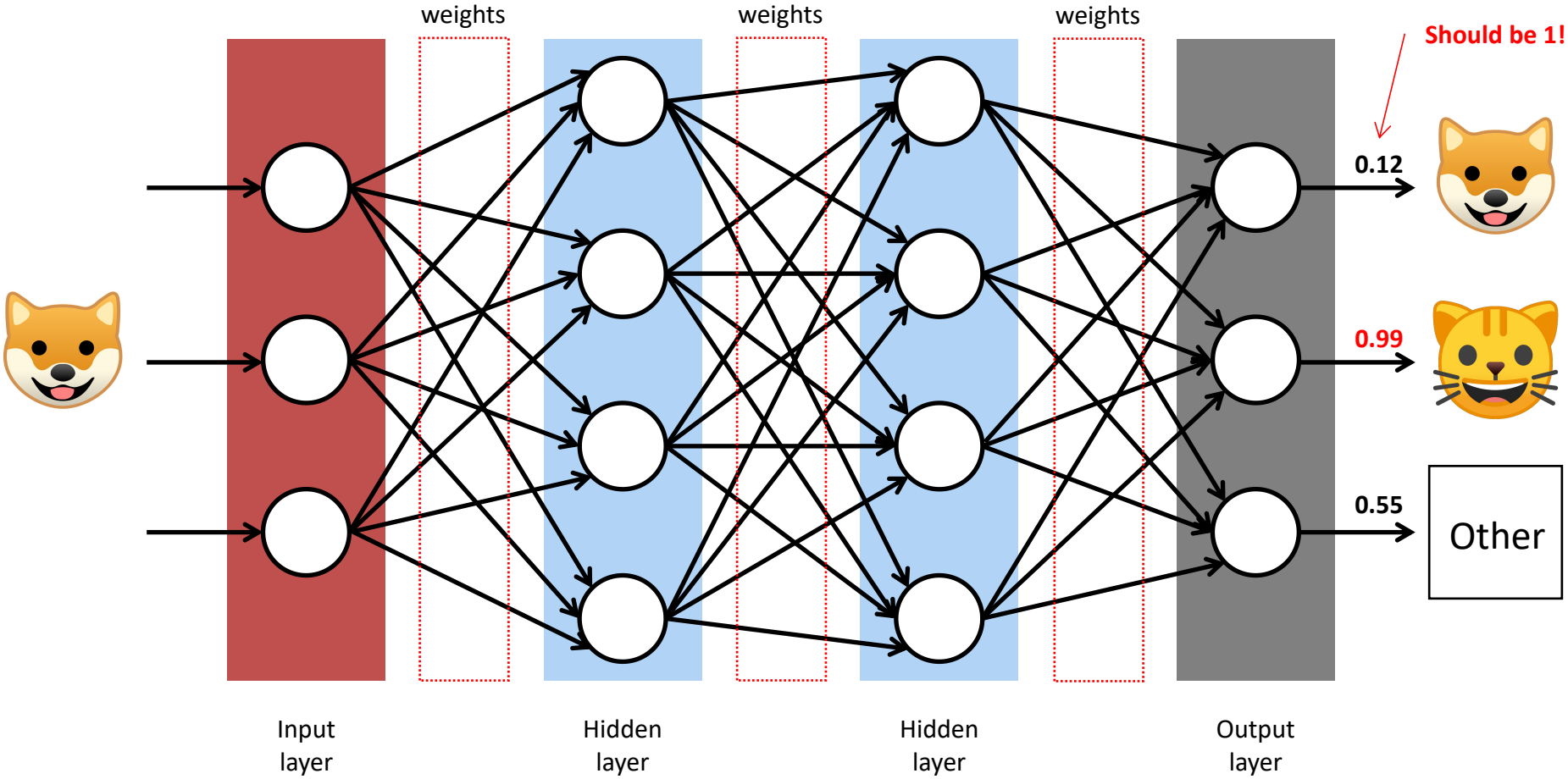
ANN: Supervised Learning

An **untrained classifier** will **NOT label input data correctly**.



ANN: Training

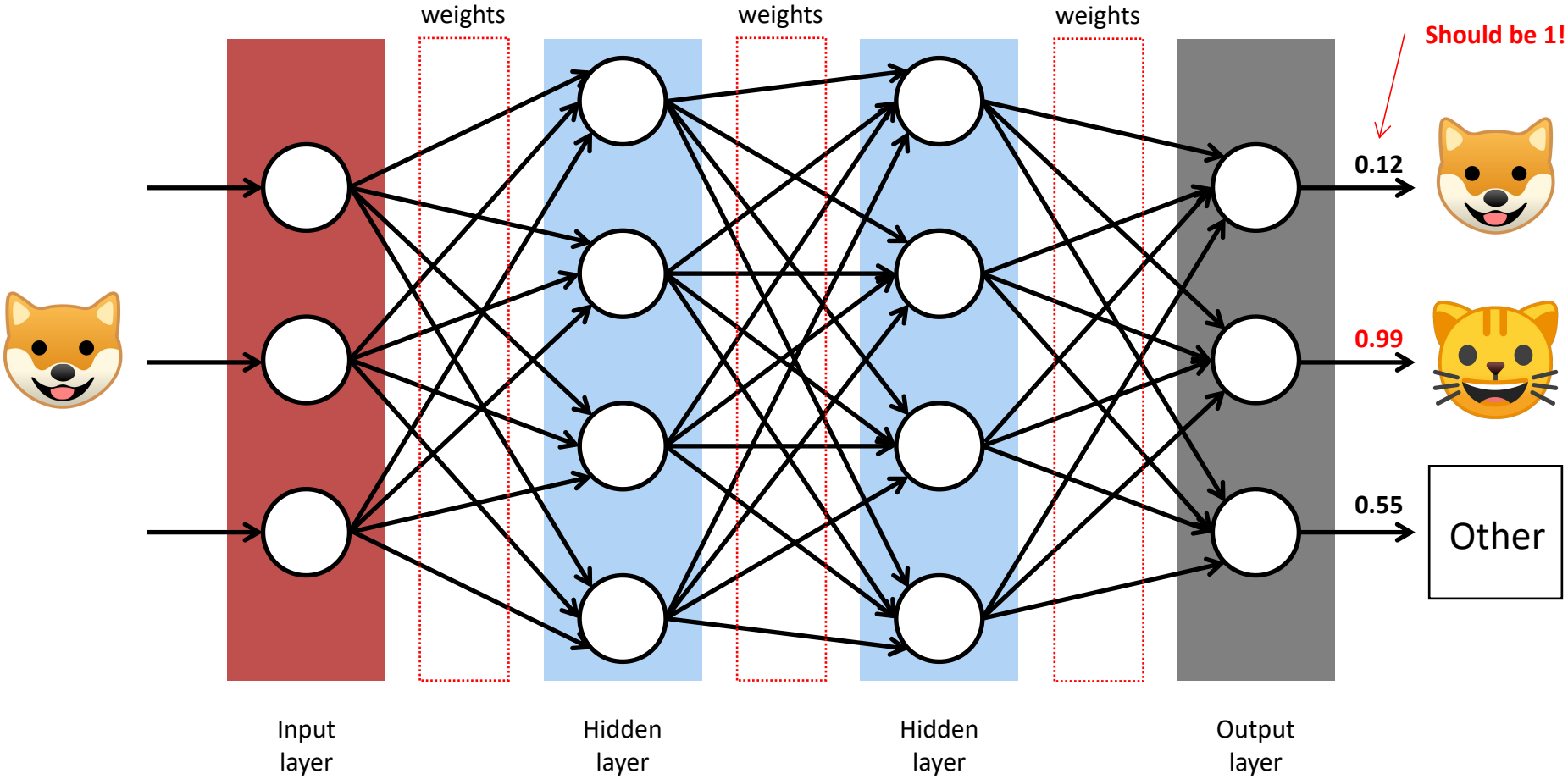
Given: input data  and it's corresponding **expected** label: DOG calculate "error".



"Error" = 0.88. Go back and **adjust all the weights** to ensure it is lower next time.

ANN: Training

Show data / label pair:  / DOG.



Correct all the weights. Repeat many times.

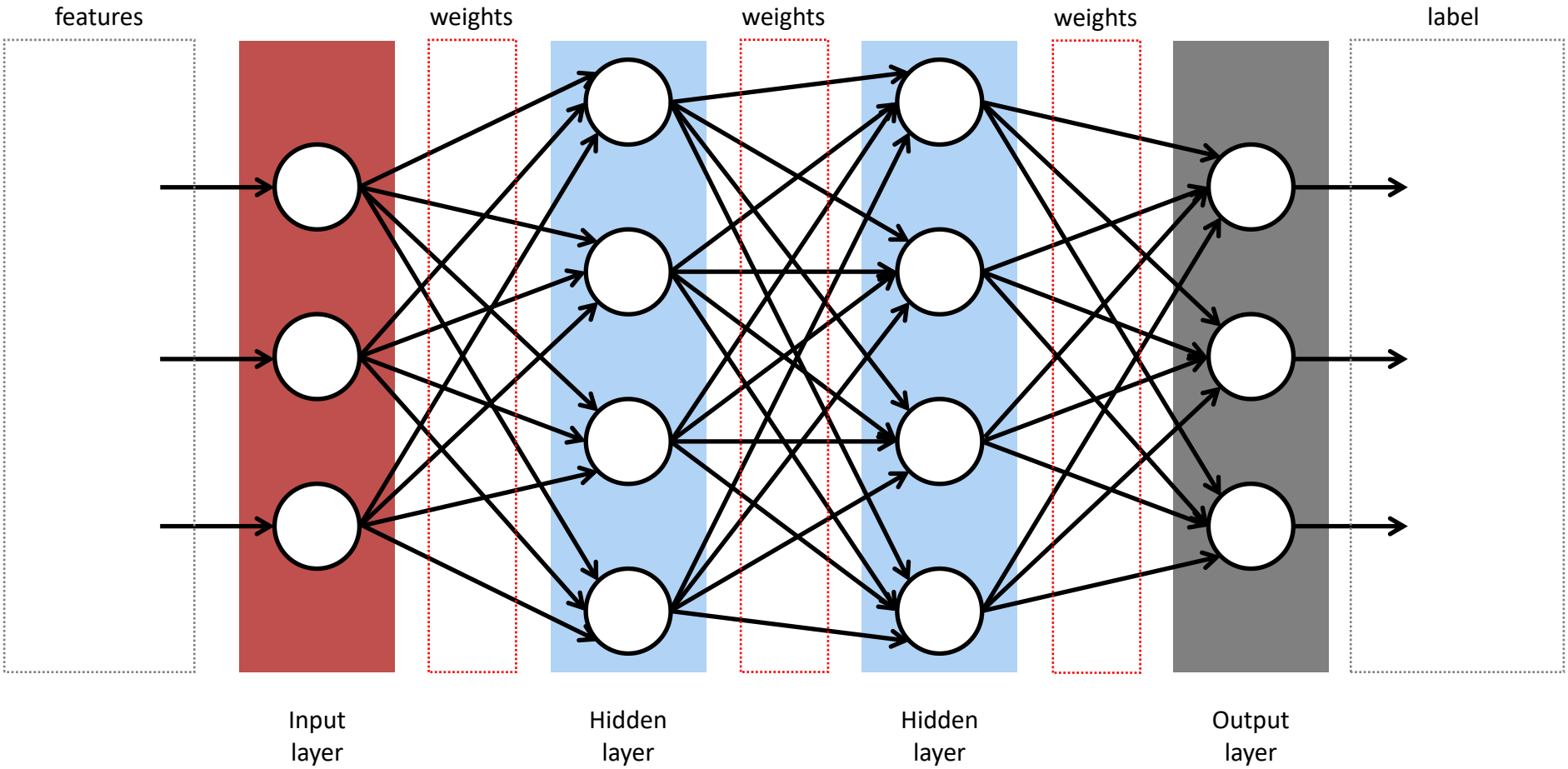
Exercise: ANN Demo

http://playground.tensorflow.org/

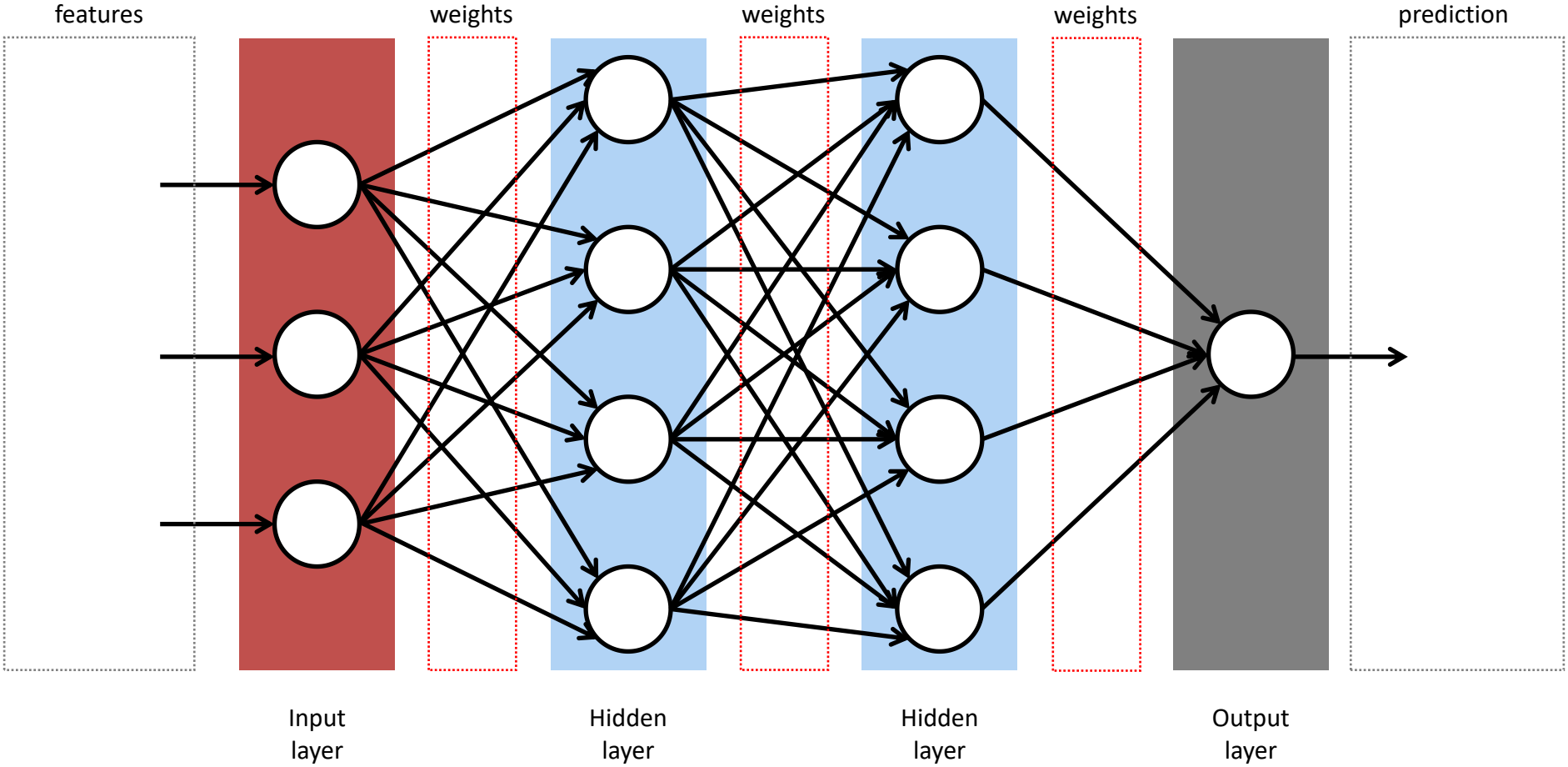
Exercise: Train a Classifier!

https://teachablemachine.withgoogle.com/

ANN for Classification

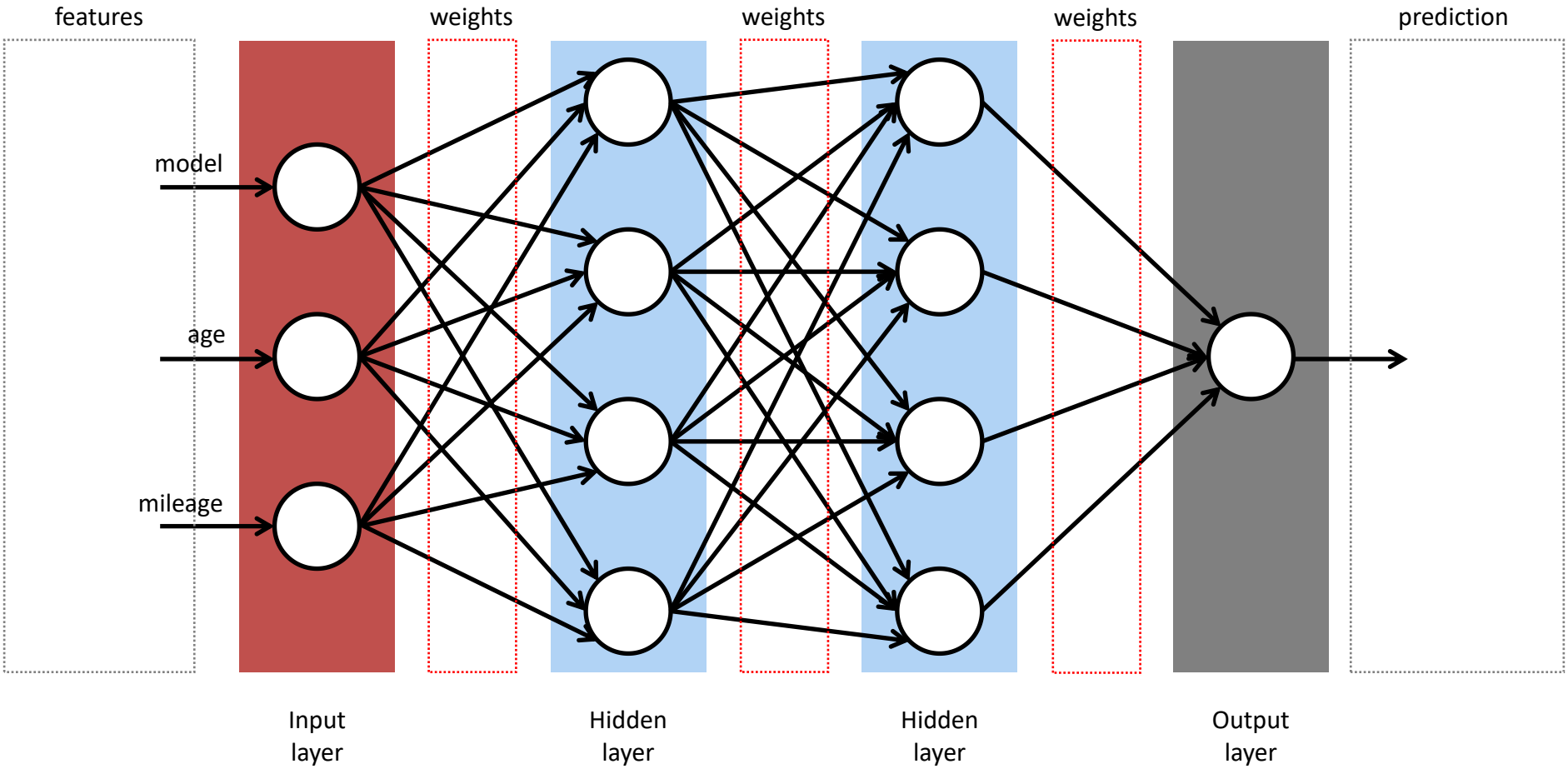


ANN for Regression



ANN for Regression: Used Car Price

Used car price predictor: train it first with used **car data - price** pairs.



Unsupervised Learning

What is Unsupervised Learning?

Idea:

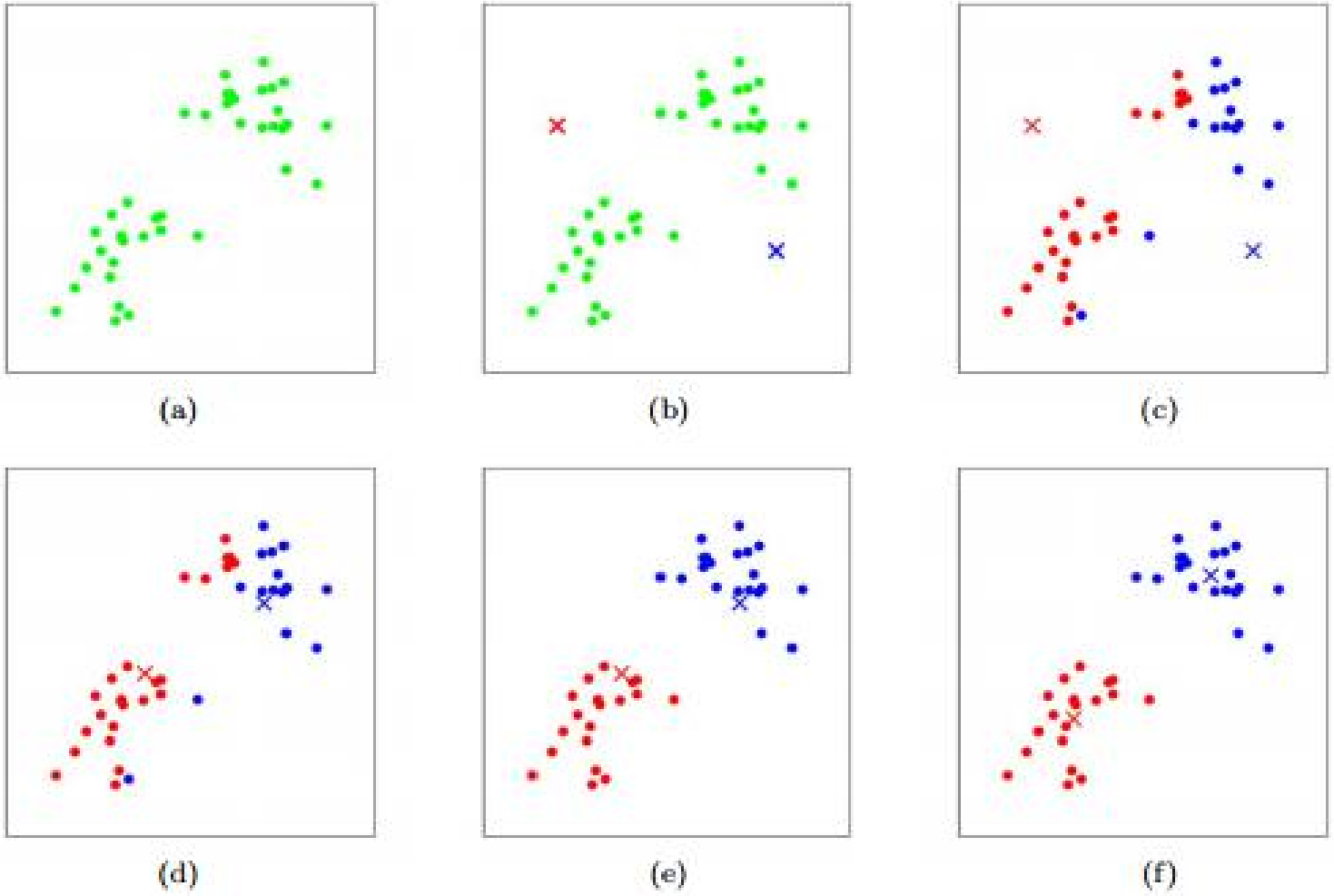
Unsupervised learning involves finding underlying patterns within data. Typically used in **clustering** data points (similar customers, etc.).

In other words:

- there is some structure (groups / clusters) in data (for example: customer information)
- we don't know what it is (**= no labels!**)
- unsupervised learning tries to discover it

Unsupervised Learning: K-Means Clustering

K-Means Clustering: The Idea

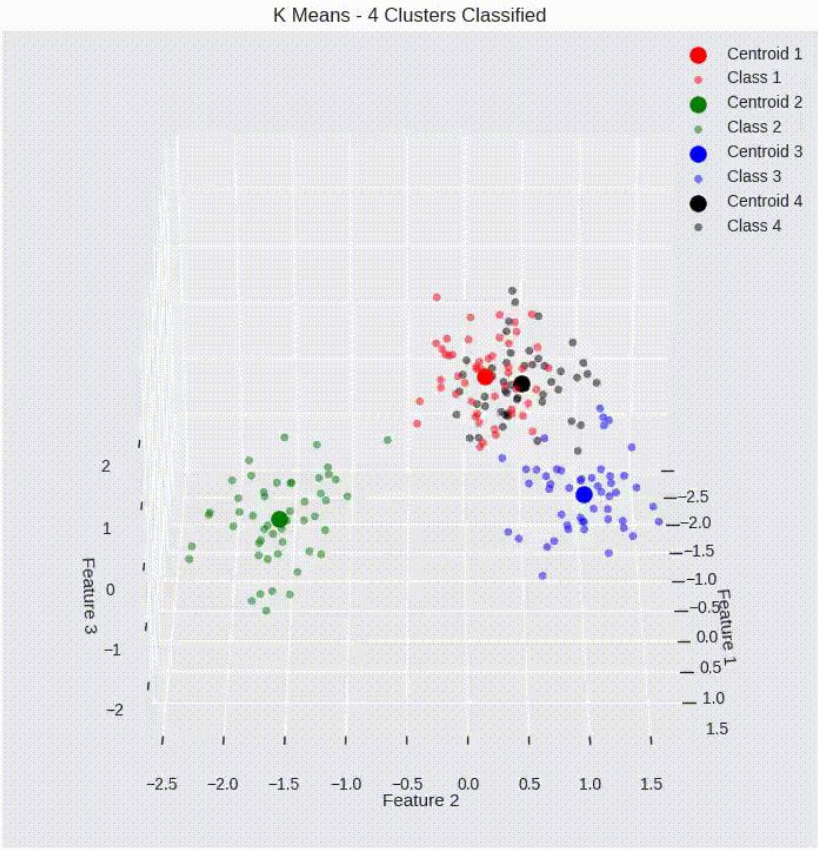


Source: https://stanford.edu/~cpiech/cs221/handouts/kmeans.html

Exercise: K-Means Clustering

https://lalejini.com/my_empirical_examples/KMeansClusteringExample/web/kmeans_clustering.html

3D K-Means Clustering Visualized



Source: https://github.com/Gautam-J/Machine-Learning

Where Would You Use Clustering?