Learning from Data

Raghavendra Selvan
Erik Dam
Data Science Lab
Faculty of SCIENCE
raghav@di.ku.dk
@SuperVoxel

UNIVERSITY OF COPENHAGEN

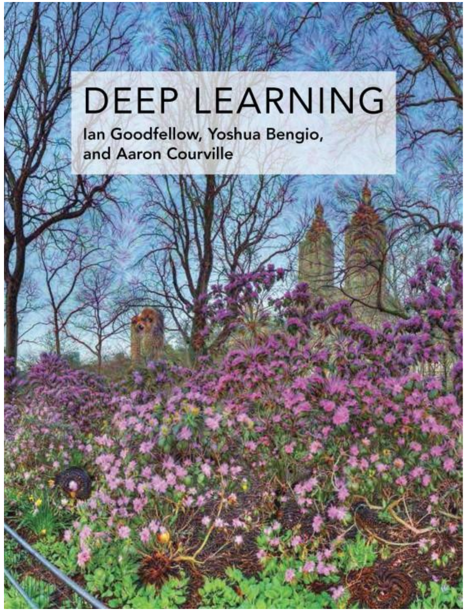
Overview of the course

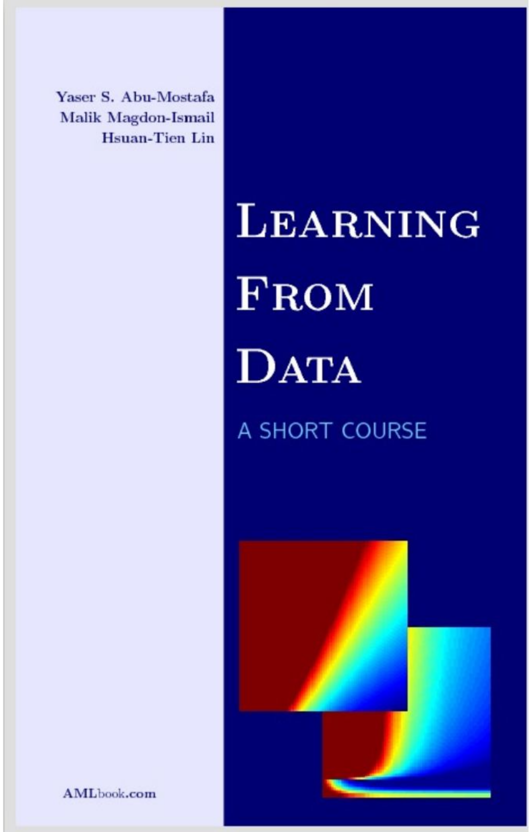
Lecture plan

	Monday	Tuesday	Wednesday	Thursday	Friday
09:00-10:30	Mon1: Overview and Denoising	Tue1: Learning from data	Wed1: Linear Models	Thu1: Knee MRI features	Fri1: Multi Layer Perceptrons
10:45-12:00	Mon2: Feature Detection	Tue2: Knowing your data	Wed2: Optimization	Thu2: Supervised Learning 1	Fri2: Convolutional Neural Networks
Lunch					
12:45-14.15	Mon3: Ridge/Root Detection	Tue3: Machine Learning Basics-1	Wed3: Unsupervised Learning 1	Thu3: Supervised Learning 2	Fri3: CNN root segmentation
14:30-16.00	Mon4: Postprocessing	Tue4: Machine Learning Basics-2	Wed4: Unsupervised Learning 2	Thu4: What is the classifier doing?	Fri4: Graph Neural Networks

Literature

1. Learning from Data by Mostafa et al.
2. Deep Learning by Goodfellow et al.

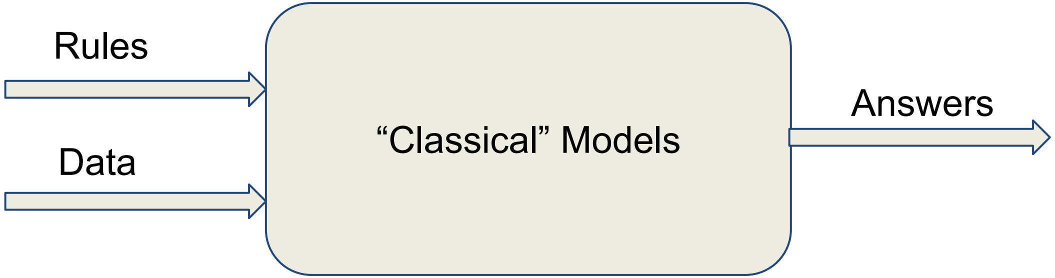




Overview

- Design-based methods
- Learning from data
- Underlying data distributions
- Perceptron
- Generalization error

Design-based methods



Design-based methods

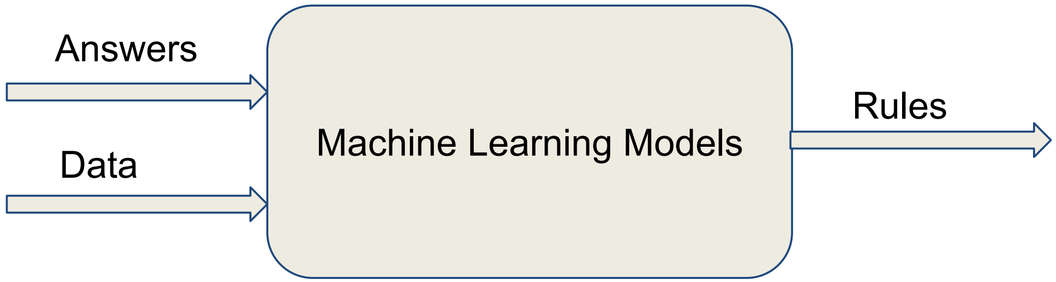
* Convert to Gray scale
* Gaussian Smoothing
* Compute Hessians
* Eigen values
* Tune thresholds

Data

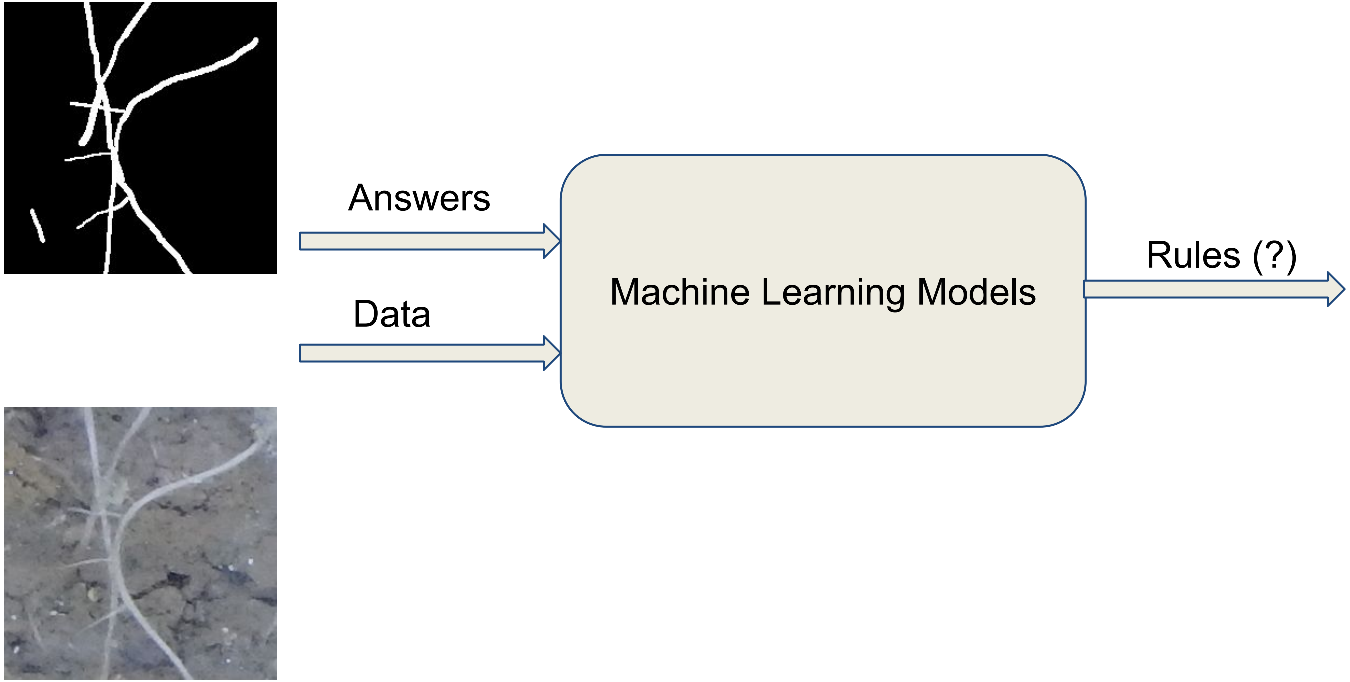
“Classical” Models:
Frangi Vesselness Filter

Answers

Machine Learning = Learning from Data



Machine Learning = Learning from Data

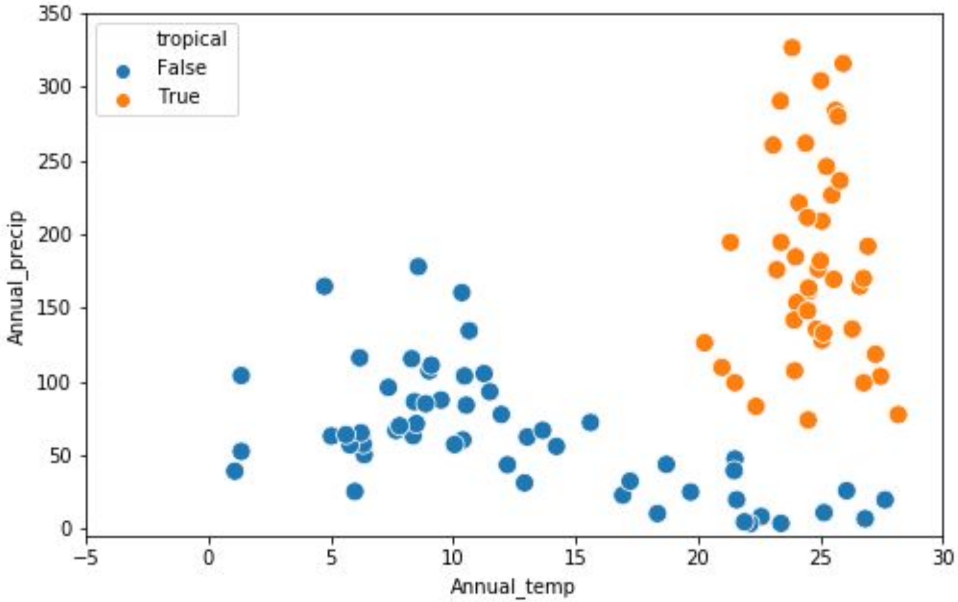


Machine Learning = Learning from Data

- ML is a lot about discovering patterns
  - Big data
  - Big computers
  - More complex patterns, than before
- Learning from examples
  - Natural to humans
  - Temptation to call it AI
- What you have is what you get (mostly)
  - Large & diverse datasets
  - Features and flaws are learned

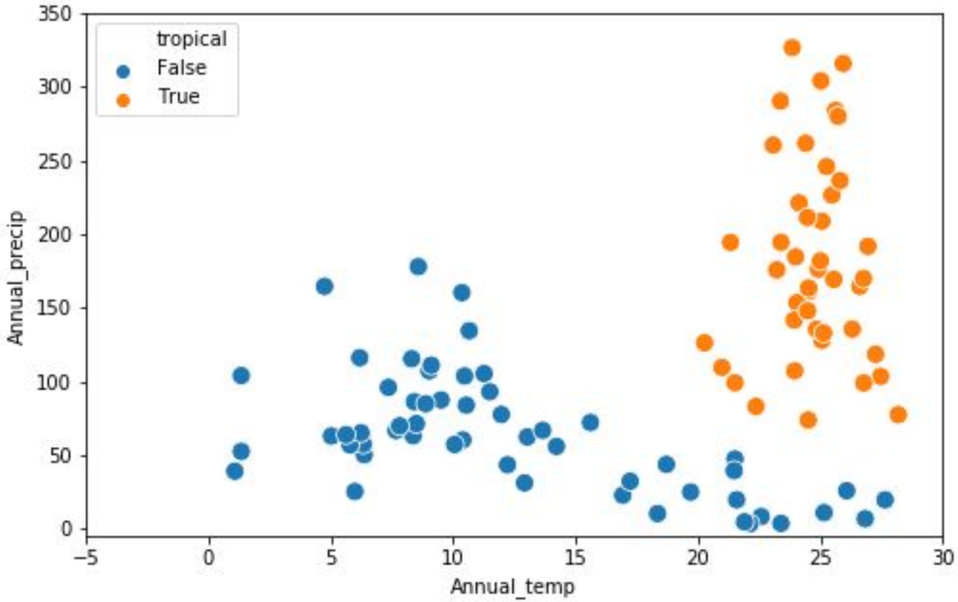
Linear Classification

- Given a training set, with binary labels
- Model a linear classifier based on the training data
- Predict classification on new data



https://datacatalog.worldbank.org/dataset/climate-change-knowledge-portal-historical-data

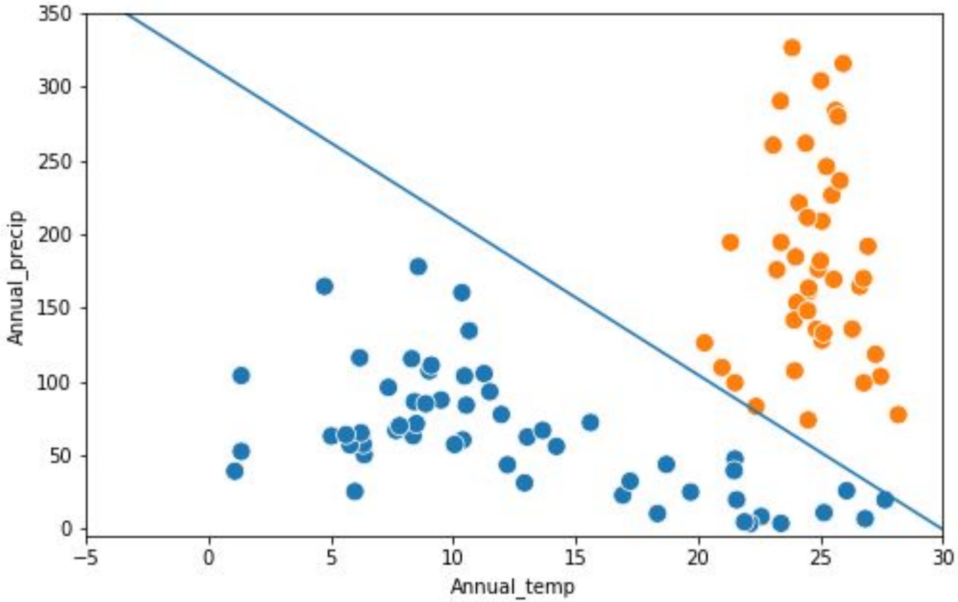
Linear Separability



https://datacatalog.worldbank.org/dataset/climate-change-knowledge-portal-historical-data

Linear Separability

If the d-dimensional data is linearly separable, then there exists at least one (d-1) dimensional hyperplane that is a classifier.



https://datacatalog.worldbank.org/dataset/climate-change-knowledge-portal-historical-data

Linear Separability

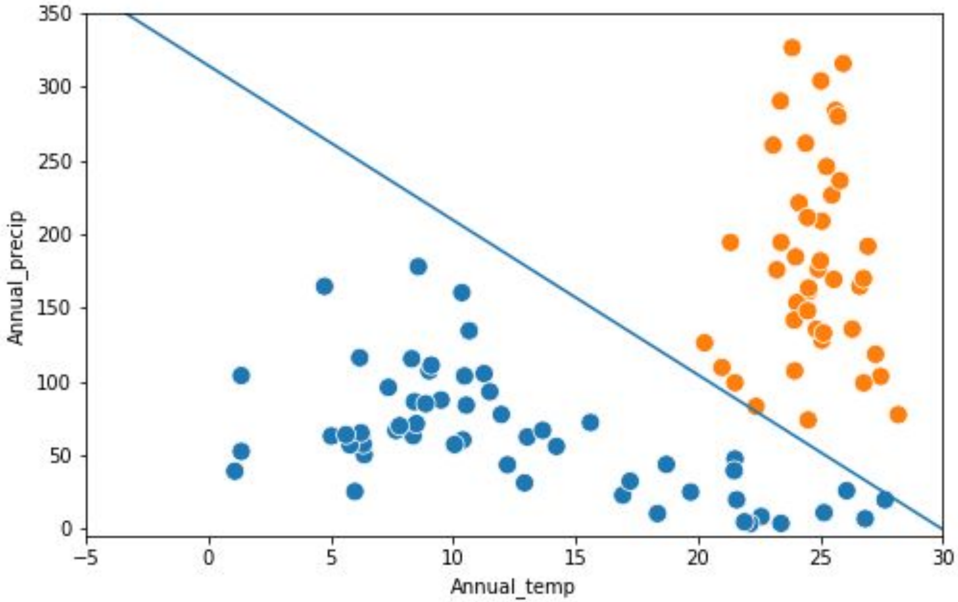
If the d-dimensional data is linearly separable, then there exists at least one (d-1) dimensional hyperplane that is a classifier.

Mathematically, the hyperplane (in this case) a line is given as:

$$w_0 + w_1 x_1 + w_2 x_2 = 0$$

Or in vector form,

$$\mathbf{w}^T \mathbf{x} = 0$$

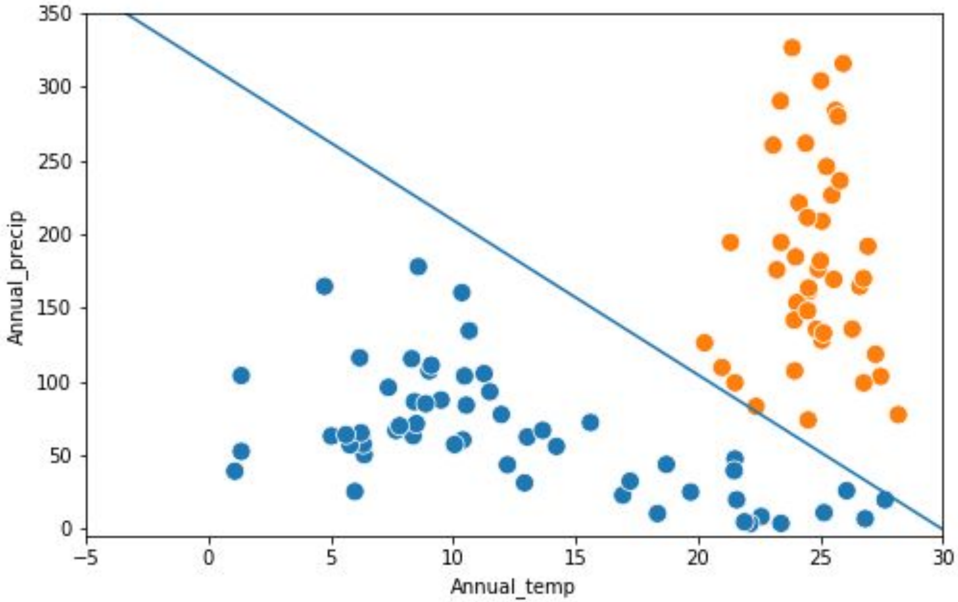


https://datacatalog.worldbank.org/dataset/climate-change-knowledge-portal-historical-data

A first ML Algorithm

Given the training data,

$$\mathbf{X} = \{\mathbf{x_i}\} : \mathbf{x_i} = [x_0, \ldots, x_d]^T$$
$$\mathbf{Y} = \{y_i\} : y_i \in \{+1, -1\}$$



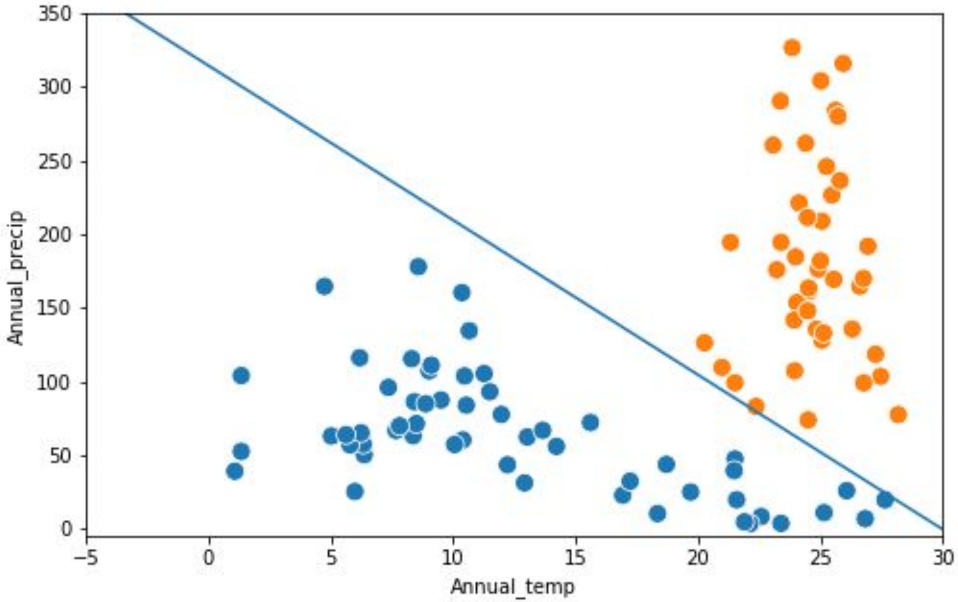
A first ML Algorithm

Given the training data,

$$\mathbf{X} = \{\mathbf{x_i}\} : \mathbf{x_i} = [x_0, \ldots, x_d]^T$$
$$\mathbf{Y} = \{y_i\} : y_i \in \{+1, -1\}$$

The model should be of the form:

$$h(\mathbf{x}) = \begin{cases} +1 & \text{if } \mathbf{w}^T\mathbf{x} \geq 0 \\ -1 & \text{if } \mathbf{w}^T\mathbf{x} < 0 \end{cases}$$



A first ML Algorithm

Given the training data,

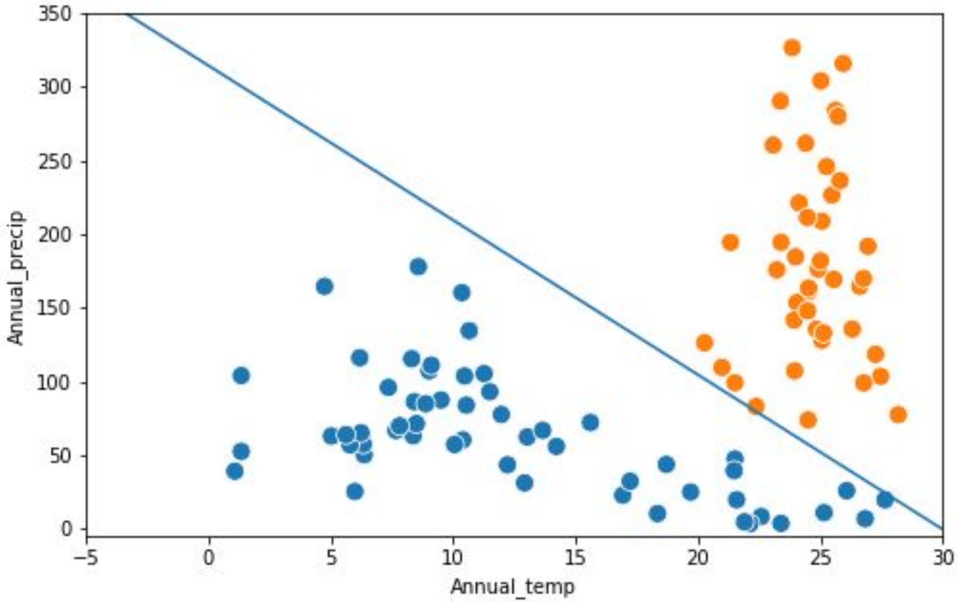
$$\mathbf{X} = \{\mathbf{x_i}\} : \mathbf{x_i} = [x_0, \ldots, x_d]^T$$
$$\mathbf{Y} = \{y_i\} : y_i \in \{+1, -1\}$$

The model should be of the form:

$$h(\mathbf{x}) = \begin{cases} +1 & \text{if } \mathbf{w}^T\mathbf{x} \geq 0 \\ -1 & \text{if } \mathbf{w}^T\mathbf{x} < 0 \end{cases}$$

Or, more compactly:

$$h(\mathbf{x}) = \text{sign}\left(\mathbf{w}^T\mathbf{x}\right)$$



A first ML Algorithm

Given the training data,

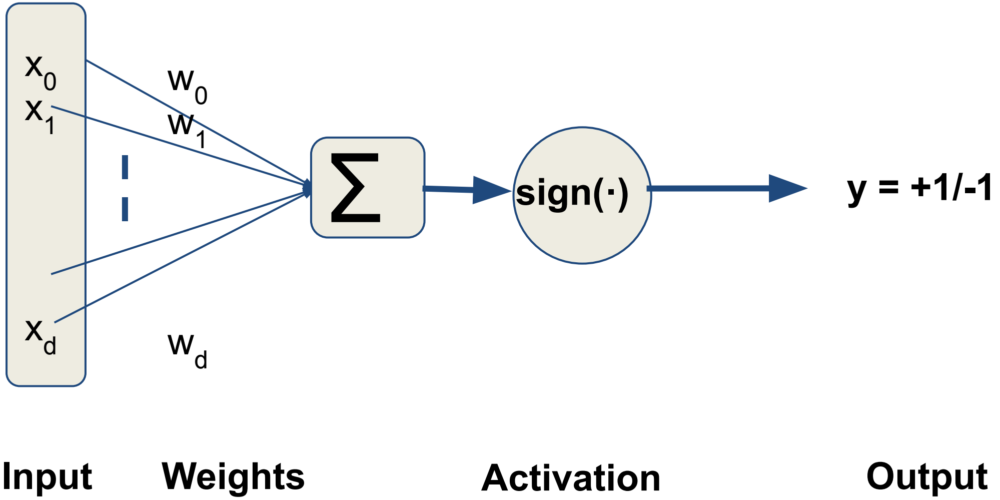
$$\mathbf{X} = \{\mathbf{x_i}\} : \mathbf{x_i} = [x_0, \ldots, x_d]^T$$
$$\mathbf{Y} = \{y_i\} : y_i \in \{+1, -1\}$$

The model should be of the form:

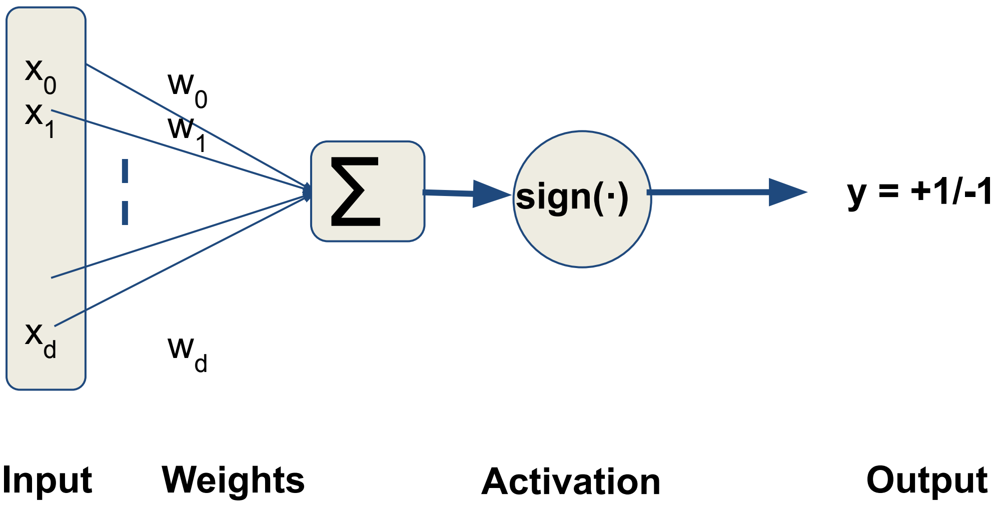
$$h(\mathbf{x}) = \begin{cases} +1 & \text{if } \mathbf{w}^T\mathbf{x} \geq 0 \\ -1 & \text{if } \mathbf{w}^T\mathbf{x} < 0 \end{cases}$$

Or, more compactly:

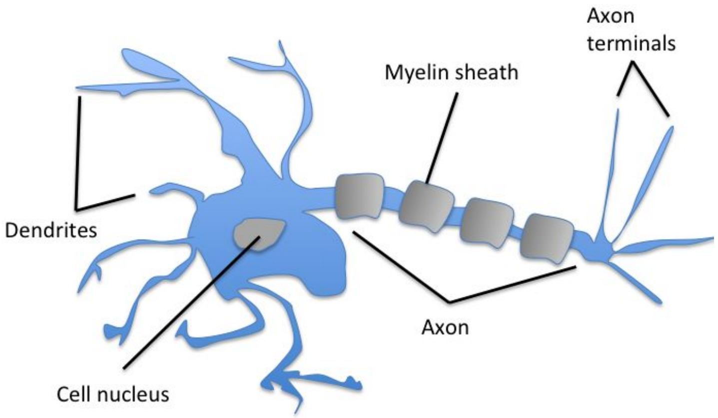
$$h(\mathbf{x}) = \text{sign}\left(\mathbf{w}^T\mathbf{x}\right)$$



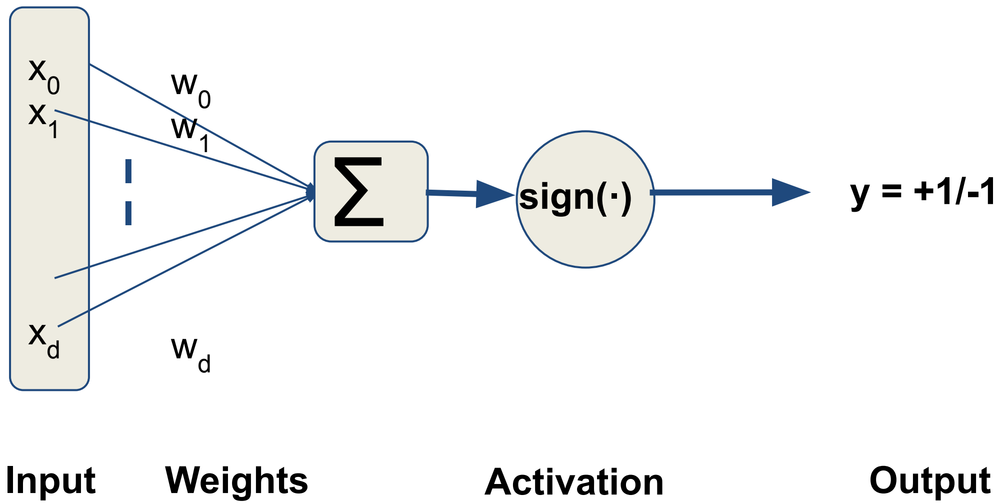
Perceptron Learning Algorithm



Perceptron Learning Algorithm



Schematic of a biological neuron.



https://sebastianraschka.com/Articles/2015_singlelayer_neurons.html

Perceptron Learning Algorithm

Algorithm: Perceptron Learning Algorithm

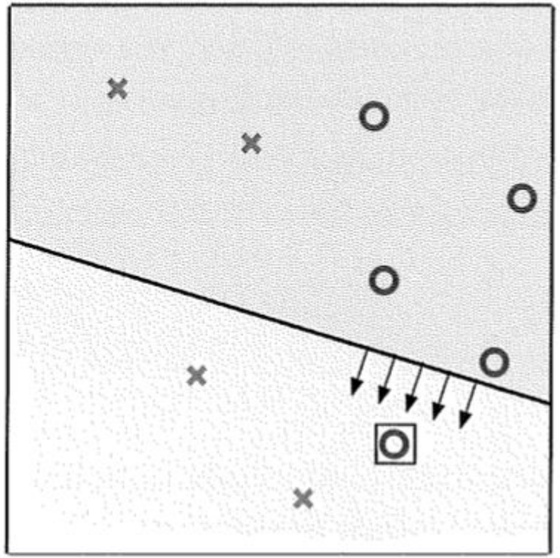
$P \leftarrow inputs \quad with \quad label \quad 1;$
$N \leftarrow inputs \quad with \quad label \quad 0;$
Initialize $\mathbf{w}$ randomly;
while $!convergence$ **do**
  Pick random $\mathbf{x} \in P \cup N$;
  if $\mathbf{x} \in P \quad and \quad \mathbf{w}.\mathbf{x} < 0$ **then**
    $\mathbf{w} = \mathbf{w} + \mathbf{x}$;
  end
  if $\mathbf{x} \in N \quad and \quad \mathbf{w}.\mathbf{x} \geq 0$ **then**
    $\mathbf{w} = \mathbf{w} - \mathbf{x}$;
  end
end
//the algorithm converges when all the inputs are classified correctly

Perceptron Learning Algorithm

Algorithm: Perceptron Learning Algorithm

$P \leftarrow$ *inputs with label* 1;
$N \leftarrow$ *inputs with label* 0;
Initialize $\mathbf{w}$ randomly;
while !*convergence* **do**
    Pick random $\mathbf{x} \in P \cup N$;
    if $\mathbf{x} \in P$ *and* $\mathbf{w}.\mathbf{x} < 0$ **then**
        $\mathbf{w} = \mathbf{w} + \mathbf{x}$;
    end
    if $\mathbf{x} \in N$ *and* $\mathbf{w}.\mathbf{x} \geq 0$ **then**
        $\mathbf{w} = \mathbf{w} - \mathbf{x}$;
    end
end
//the algorithm converges when all the inputs are classified correctly

$$\mathbf{w}(t+1) = \mathbf{w}(t) + y(t)\mathbf{x}(t).$$

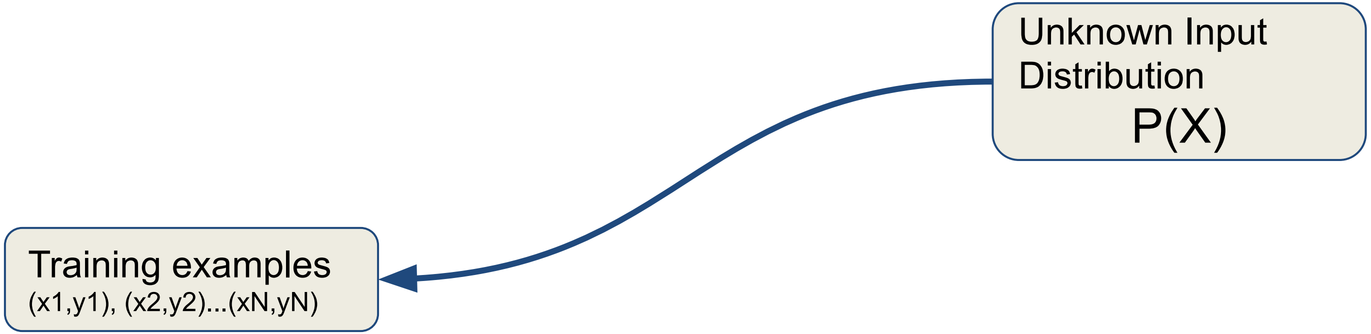


PLA Task in Notebook

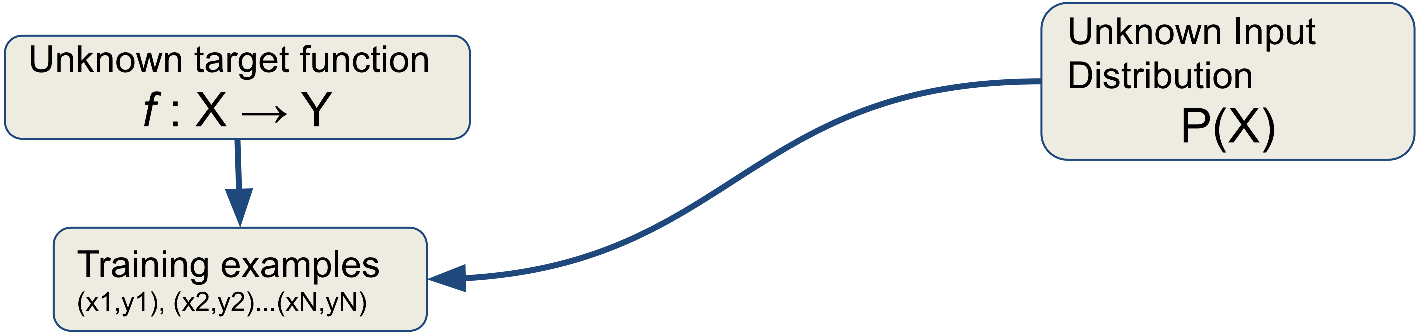
Formalising Machine Learning

Training examples
(x1,y1), (x2,y2)...(xN,yN)

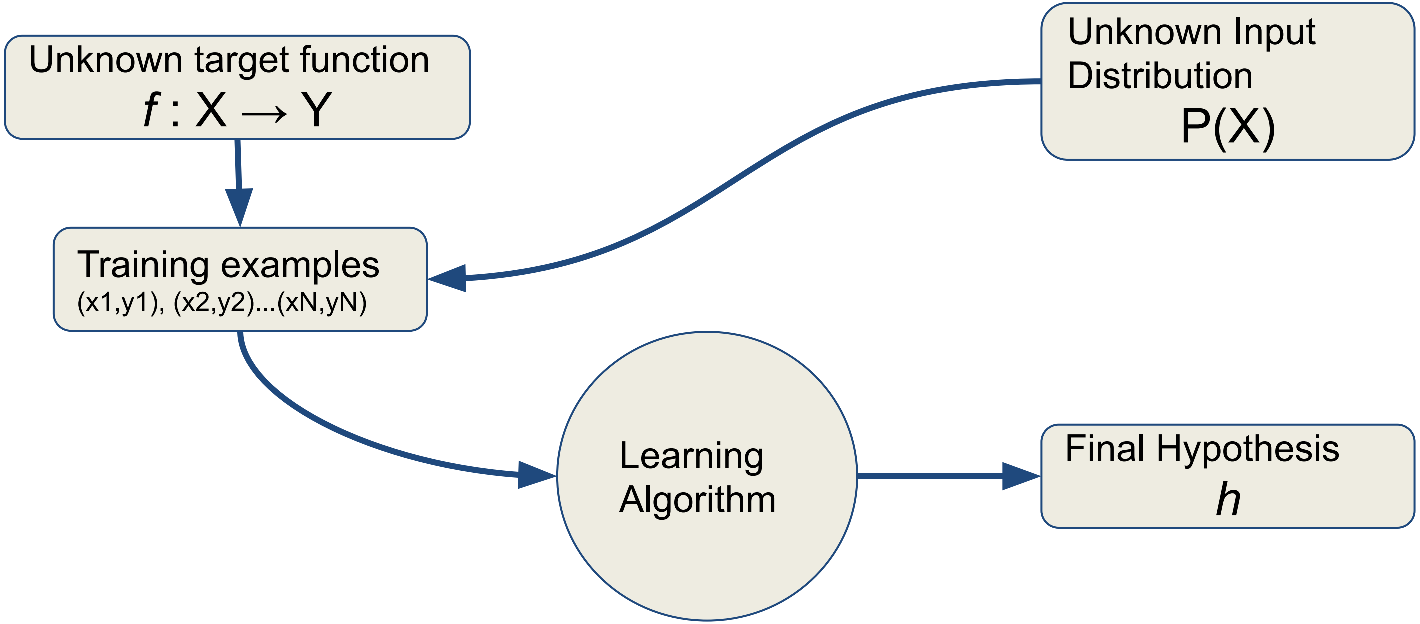
Formalising Machine Learning



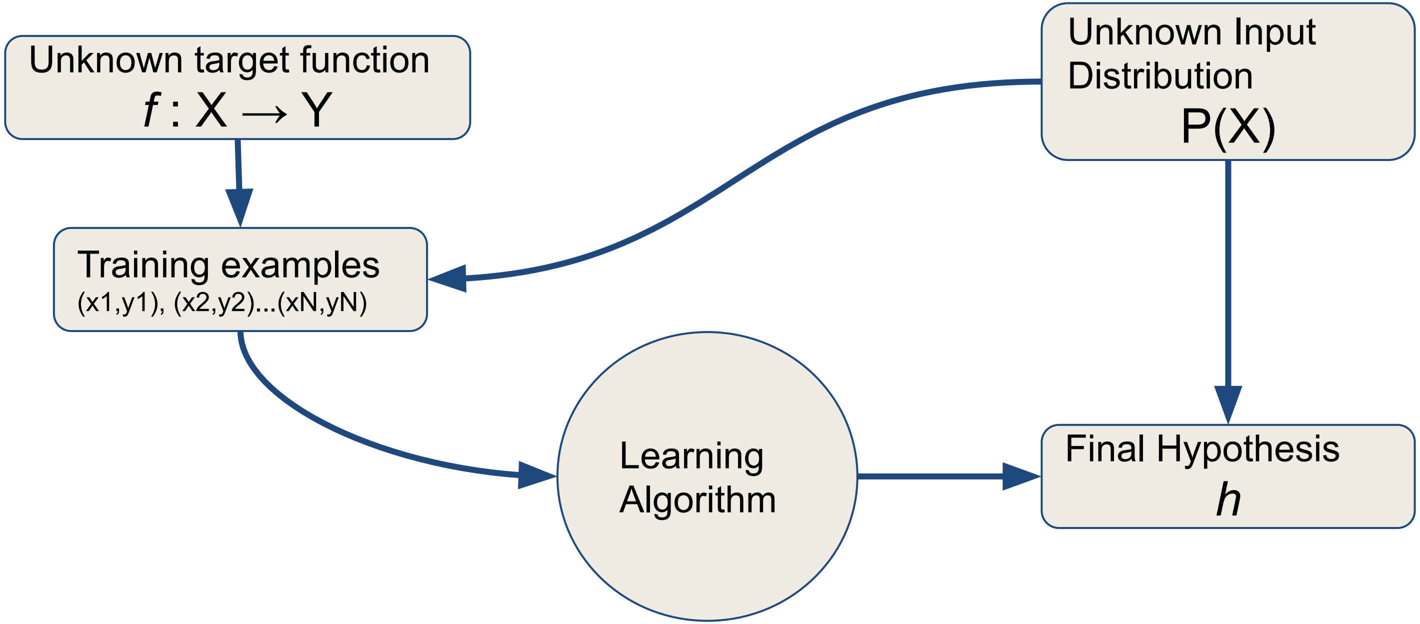
Formalising Machine Learning



Formalising Machine Learning



Formalising Machine Learning



Formalising Machine Learning

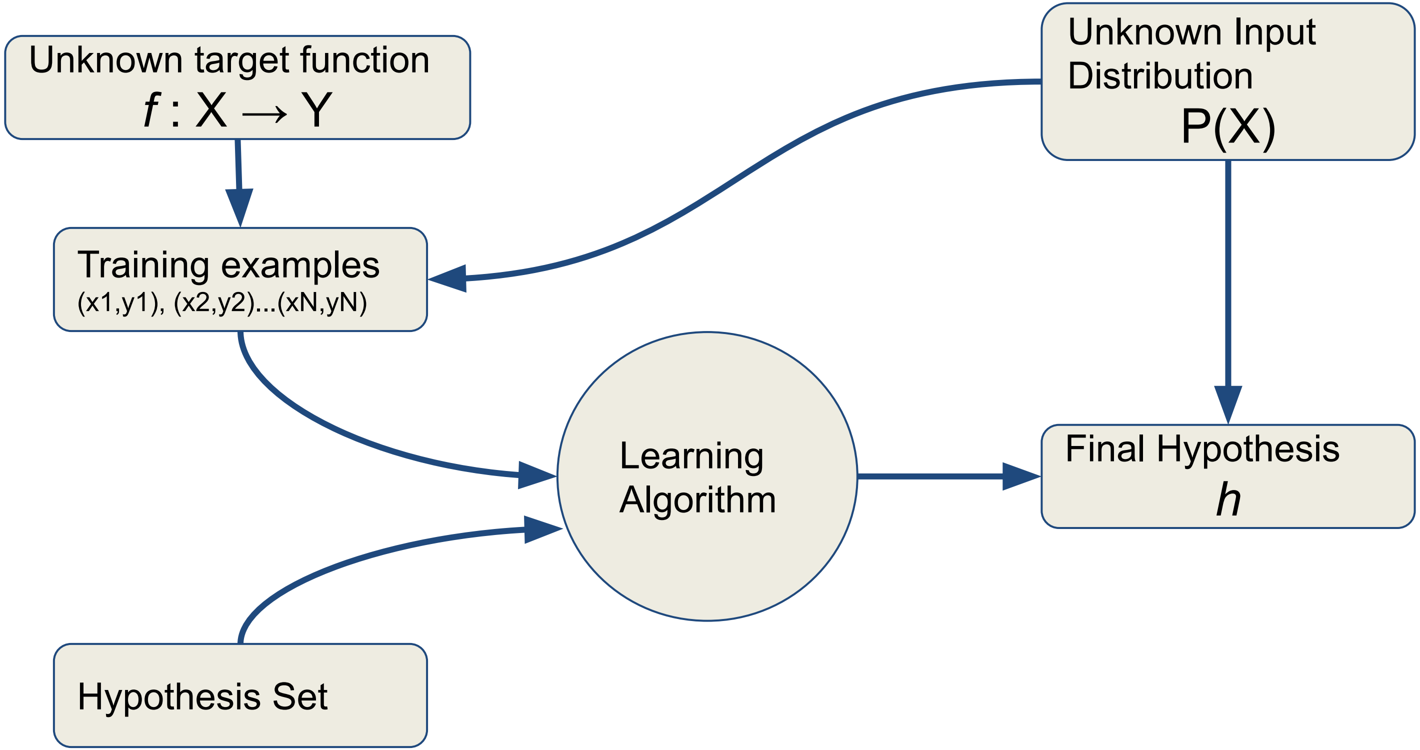


Figure based on Fig.1.9 Mostafa et al.

Generalization Error

Generalization Error

$$\mathbf{E}_{in}(h) = \frac{1}{n}\sum_{i=1}^{N} l(h(X_i), Y_i)$$

$$\mathbf{E}_{out}(h) = \mathbb{E}_{p(X,Y)}[l(h(X), Y)]$$

$$\mathcal{G}_{err} = \mathbf{E}_{out}(h) - \mathbf{E}_{in}(h)$$

Not obvious to minimize the generalization error.

Relation between data and generalization

- IID assumptions

Summary

- Can learn from data!
- Overcomes tedious model designs
- Perceptrons mimic biological neurons

Summary

- Can learn from data!
- Overcomes tedious model designs
- Perceptrons mimic biological neurons

However,

- Depends on the data
  - Strong assumptions
  - Distribution
  - Number of samples
  - High quality labels
- Many (equally better/worse) models to choose from
- Hard to generalize