Another way to classify data using a smoothing function:

$x_1, x_2, \dots x_k$ → Types ① ② ③ ④ ⋯

(input variables)

y_1 y_2 y_3 y_4 ⋯

↑ 0 or 1

Predicted y-values: $\hat{y}_1$ $\hat{y}_2$ $\hat{y}_3$ $\hat{y}_4$

want to be between 0 & 1.

Parameters θ for model for each type

$\theta^{(j)}$ for type y_j.

Do regression: answware.

$$\hat{y}_j = \frac{e^{\theta^{(j)T} x}}{\sum_j e^{\theta^{(j)T} x}}$$

← soft max function.

See that $\sum_j \hat{y}_j = 1$.

Think that $\hat{y}_j$ = probability that the data point is of type (j) (i.e. $y_j = 1$).

Your real prediction:

$$\max_j \text{ of } \frac{e^{\theta^{(j)T} x}}{\sum_k e^{\theta^{(k)T} x}} = \text{prob. that your answer is correct.}$$

See examples of SVM classifiers using polynomial regression inputs:

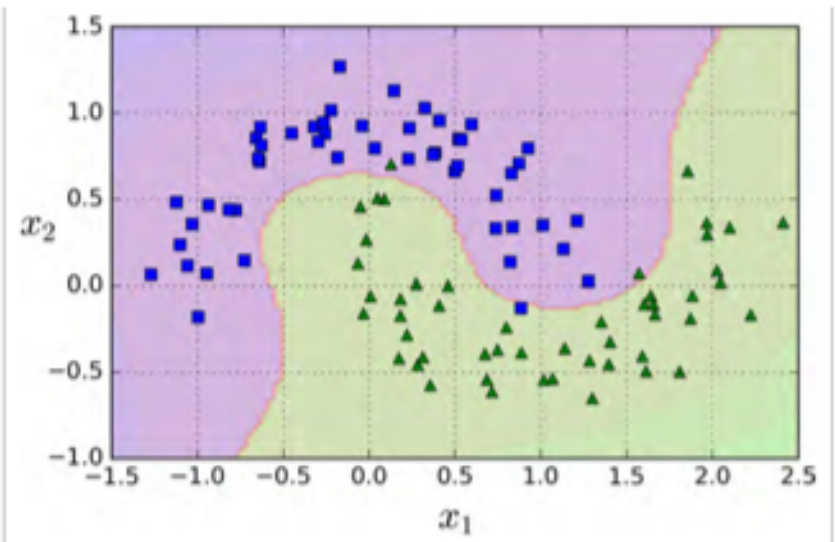


Figure 5-6. Linear SVM classifier using polynomial features

Another Classification Algorithm: Decision Tree.

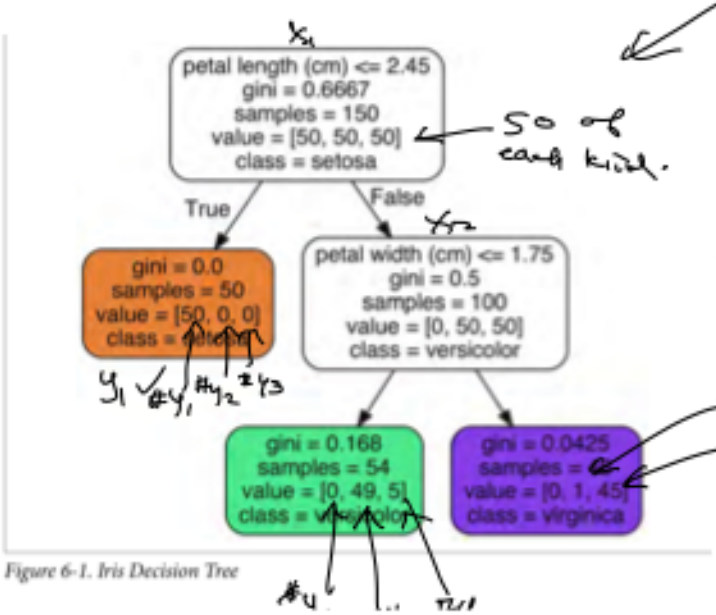


Figure 6-1. Iris Decision Tree

example from the book trying to classify the type of iris (setose, versicolor, virginica) from petal length, petal width, sepal length, ...

" #Y_2 #Y_3
↑ wrong

gini = shows how good the classification in a given box is.

$$= 1 - \sum_{j=1}^{k} P_j^2$$ ← percentage of j^{th} type in the box. ← want to be close to 0.

1st box gini $= 1 - \left[\left(\frac{50}{150}\right)^2 + \left(\frac{50}{150}\right)^2 + \left(\frac{50}{150}\right)^2\right]$

$\left(\frac{1}{3}\right)^2 + \left(\frac{1}{3}\right)^2 + \left(\frac{1}{3}\right)^2$

$= \frac{2}{3}.$

orange box

gini $= 1 - \left[\left(\frac{50}{50}\right)^2 + \left(\frac{0}{50}\right)^2 + \left(\frac{0}{50}\right)^2\right]$

1

$= 0.$

green box

gini $= 1 - \left[\left(\frac{0}{54}\right)^2 + \left(\frac{49}{54}\right)^2 + \left(\frac{5}{54}\right)^2\right]$

≈ 0.168

Question: How is the variable and breakpoint (eg $X_2 \leq 2.783$) chosen?

Answer: Consider all possible input columns, all possible break points → choose the input column (variable) and break pt so that it gives a minimum

Score for one of the resulting boxes.

Then: keep going through the decision tree.

How does this look like with the classification of the data?

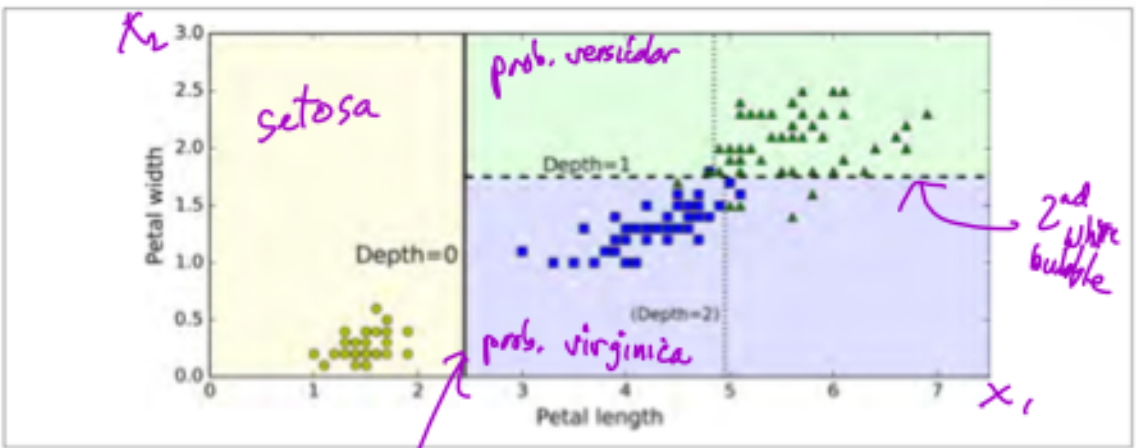


Figure 6-2. Decision Tree decision boundaries

Ensemble Learning –

several different algorithms at the same time.

- Use all the different algorithms to make a prediction of the classification of a new data point $(-, -, -, \ldots, -)$

(Then → the most often chosen classification is the one you choose.)

* Could also choose the classification is by weighting votes with output probabilities.

When we use the same training data but chop into small subsets (randomly, with replacement) ⟶ use one algorithm on this.

⟶ get ensemble result (eg by voting)

⟶ This is called bagging

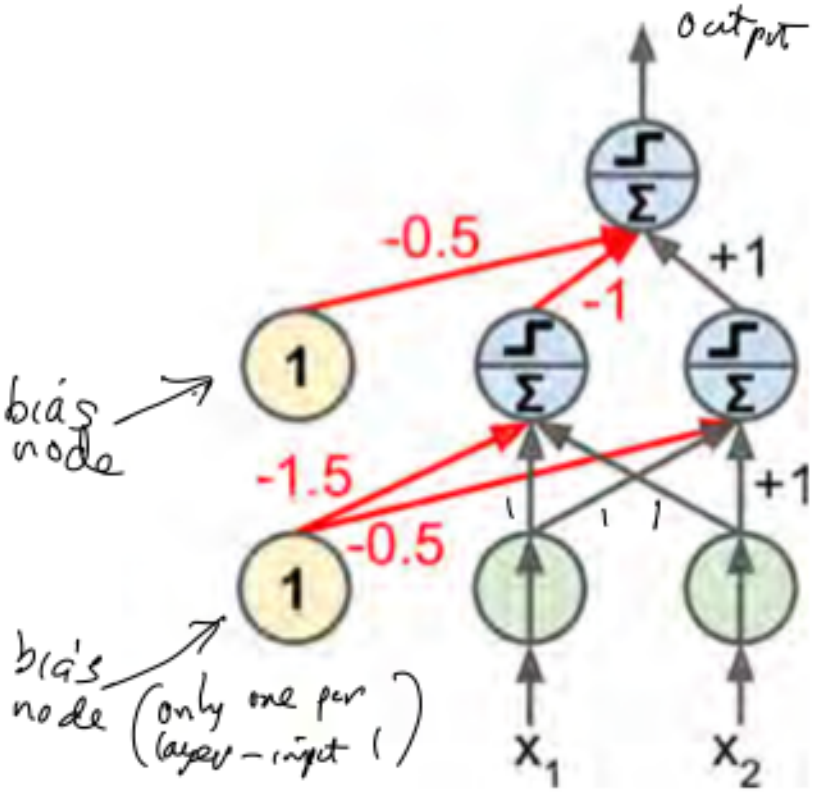
Amazing: it gives better outcome than just performing the algorithm on the whole training set.

Random Forest — uses bagging with the decision tree algorithm.

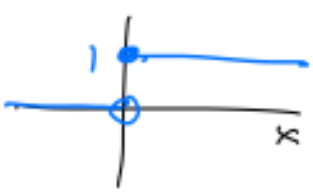
- Artificial Neural Networks (ANNs)
- Deep Learning Neural Networks —
  - same as regular neural networks but with hidden layers.

These are graph networks
nodes = vertices
input nodes & output nodes.

Example from book:



x_1 = input (0 or 1)
x_2 = input (0 or 1)

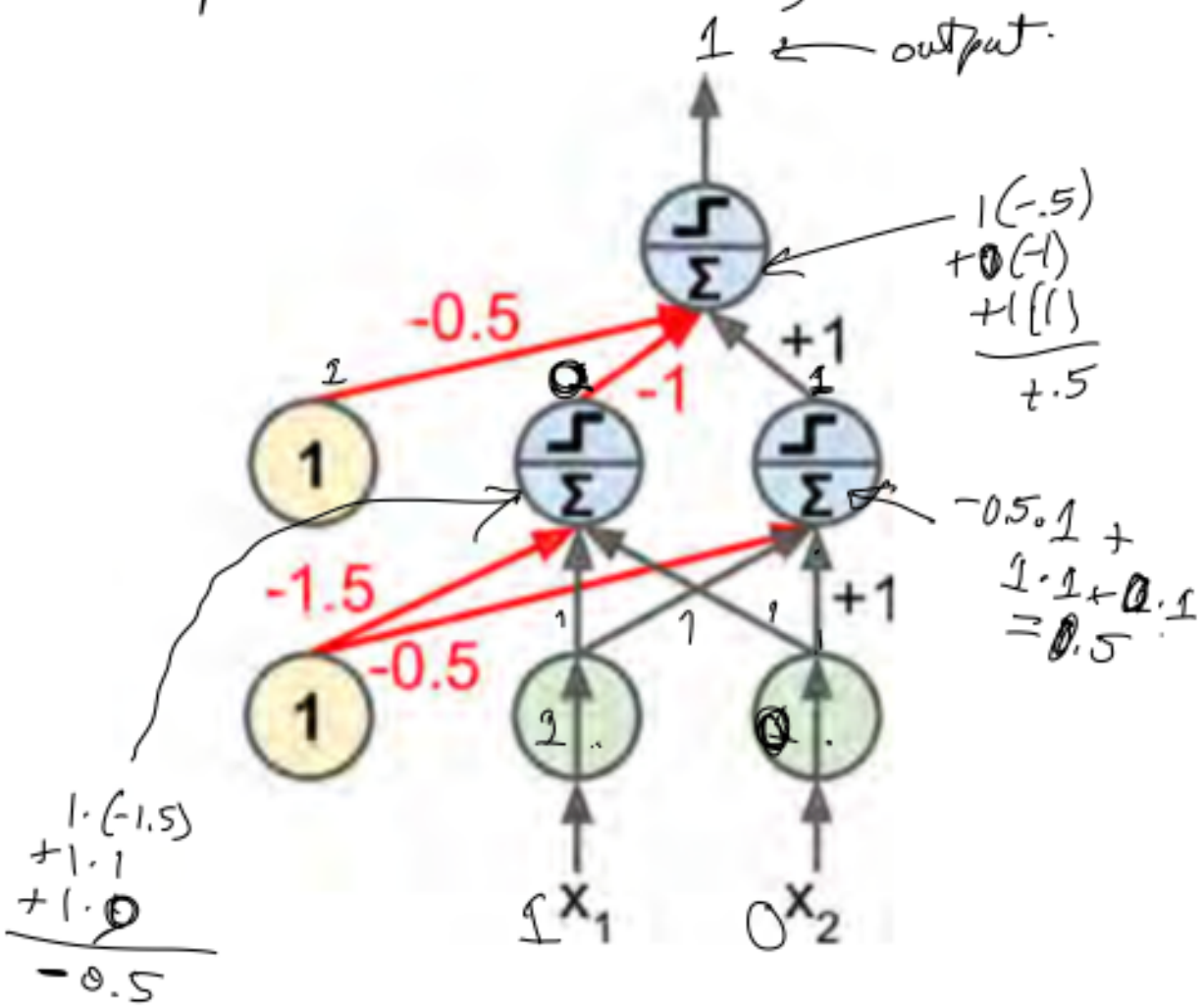


then apply $H(x)$ =

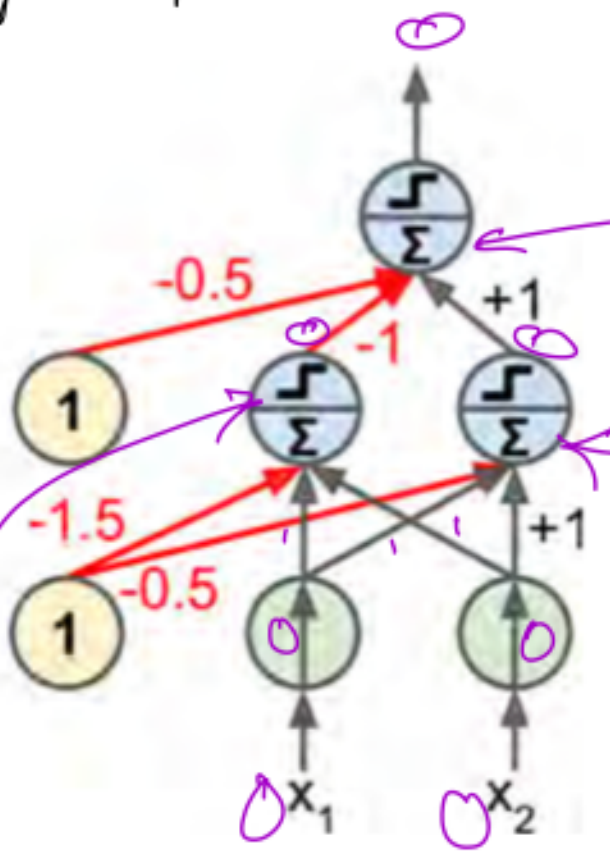
← sum up inputs

Numbers next to arrows are weights W_{ij} = weight from i^{th} node to j^{th} node. $\sum_i'$ means $\sum w_{ij} I_{ij}$

Sample input: $x_1 = 1, x_2 = 0$



Sample Input $x_1 = 0$ $x_2 = 0$



$\Sigma' = (-.5)(1) + 0 + 0 = -.5$

$\Sigma' = (-.5)(1) + (1)(0) + (1)(0) = -.5$

$\Sigma' = (-1.5)(1) + 1(0) + 1(0) = -1.5$

Set up Neural Net

- each layer has its own number of nodes —

parameters (mostly hidden)

→ weights w_{ij}.

→ Training →

reduce the weights contributing to wrong answers -

(follow backwards)

increase weights contr. to correct answers.