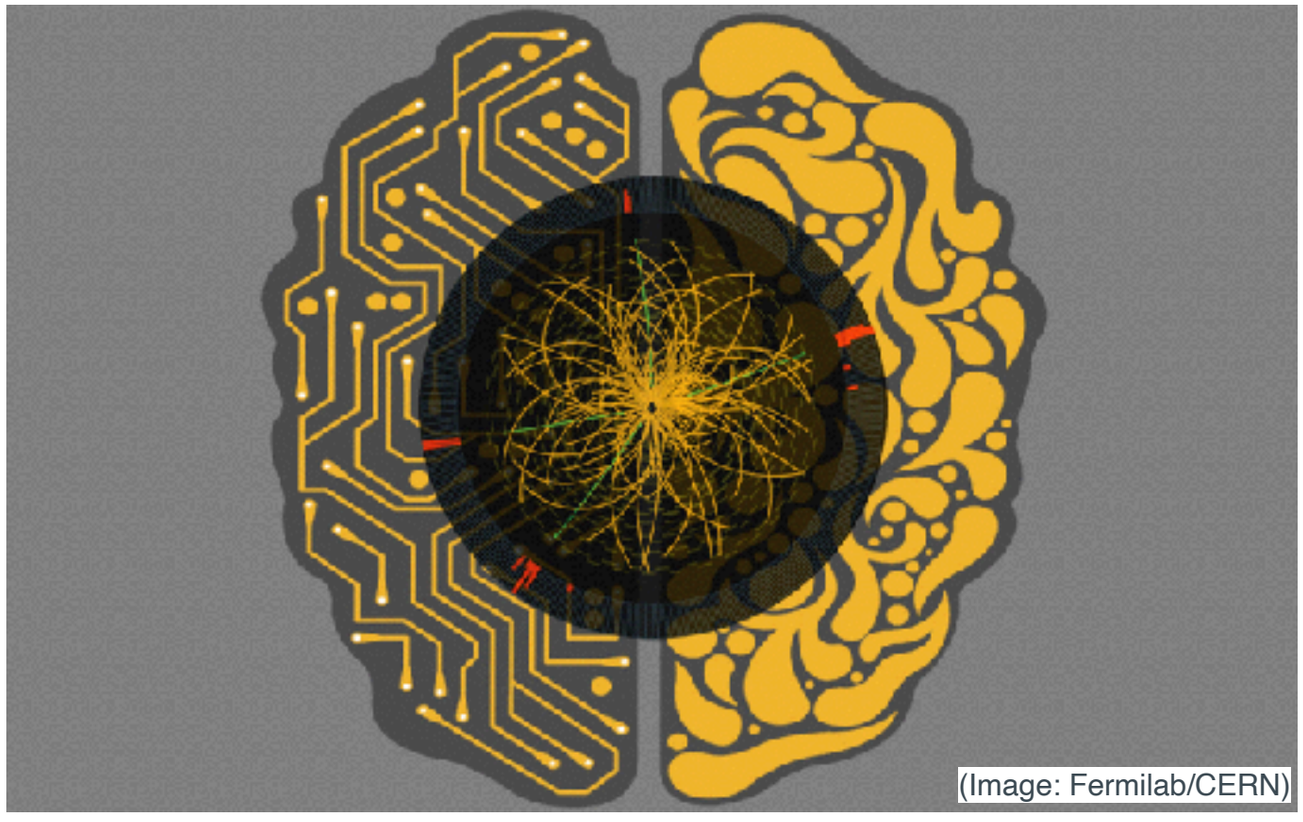
# PHY 835: Collider Physics Phenomenology

*Machine Learning in Fundamental Physics*

Gary Shiu, UW-Madison


(Image: Fermilab/CERN)

## Lecture 7: Perceptron

# Recap of Lecture 6

- Multi-class classification, e.g., handwritten digits MNIST.
- Information Content
- Shannon entropy (relative entropy, KL divergence, KL distance)
- Gibbs' inequality

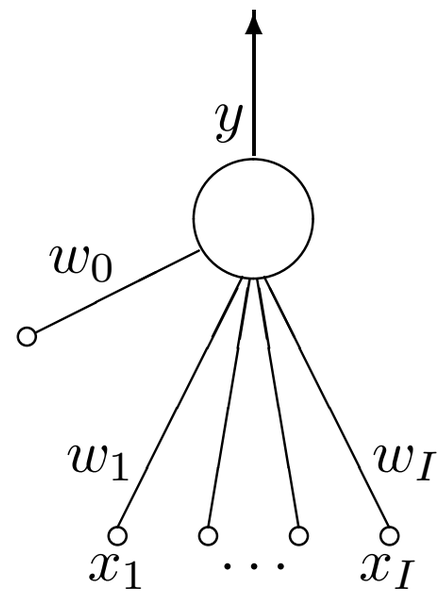
# Outline for today

- Capacity of Perceptron
- Survey of classifiers
- Decision Trees
- Reference: MacKay's book, Chapter 40

# Perceptron

- Binary classifier = single neuron (perceptron)

$$y = \sigma(\mathbf{x}^T \cdot \mathbf{w}) = \sigma(\sum_{i=0}^{I} x_i w_i)$$




- What is the capacity of this system, i.e. how much information can be stored by training a neuron? (eventually a neural network).

- Capacity = infinite (as each weight is real number)? No, as receiver is not able to examine the weights directly.

- K inputs for perceptron, N data points. Possible number of binary labels $2^N$. What is the probability that all N bits are correctly reproduced?

- How large can N be, for a given K, while keeping this probability close to 1?

# General Position

- **Assumption:** Data points are in **generic position**:
    - *Any subset of size ≤ K is linear independent, and no K+1 of them lie in a (K-1) dimensional plane.*

e.g., for K=3, no 3 points are collinear and no 4 points are coplanar.

- Goal: count the number of threshold functions T(N,K) using the linear threshold function:

$$y = f\left(\sum_{k=1}^{K} w_k x_k\right) \qquad\qquad f(a) = \begin{cases} 1 & a > 0 \\ 0 & a \leq 0. \end{cases}$$

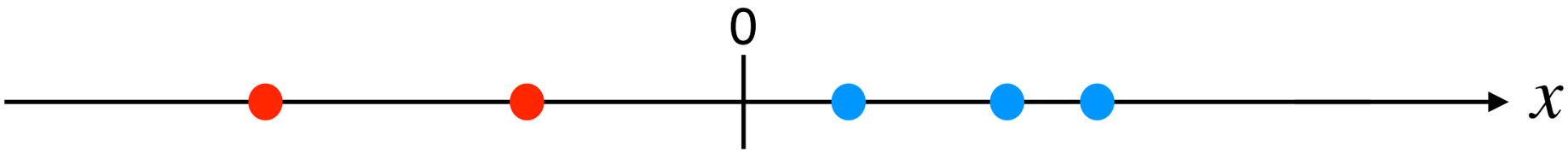
- Assume there is no bias $w_0$. The capacity of a neuron with a bias can be obtained by replacing K by K+1 in final result (even ≠ general position).

# Counting T(N,K)

- **Goal:** count T(N,K) = # distinct threshold functions on N points in general positions in K dimensions.

- Start with some special cases and derive recurrence relation.

- K=1, any N:

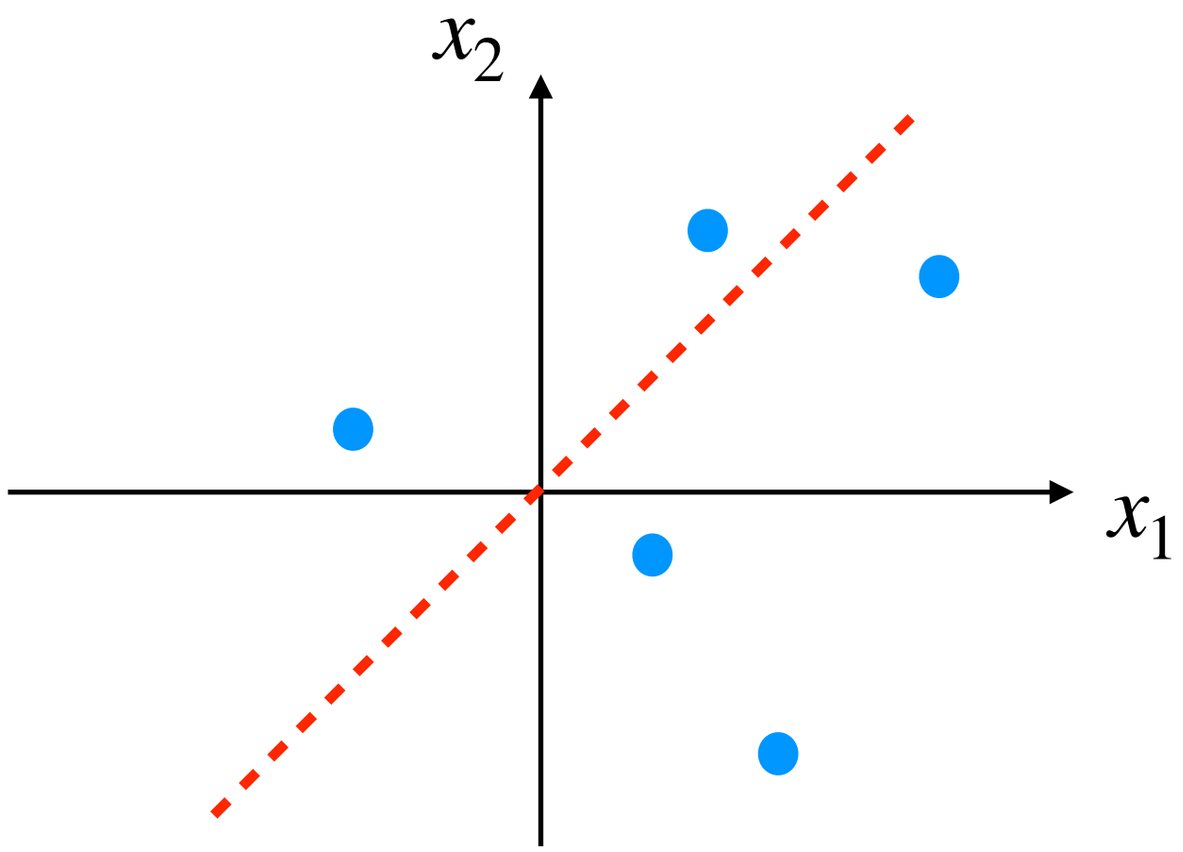



Only one weight $w_1$; by changing its sign, we can realize {red=1, blue=0} or {red=0, blue=1}, hence T(N,1)=2.

- N=1, any K: only one point $\mathbf{x}^{(1)}$ so the two possible labelings can be realized by $\mathbf{w} = \pm\, \mathbf{x}^{(1)}$, hence, T(1,K)=2.

# Counting T(N,K)

- K=2, any N:

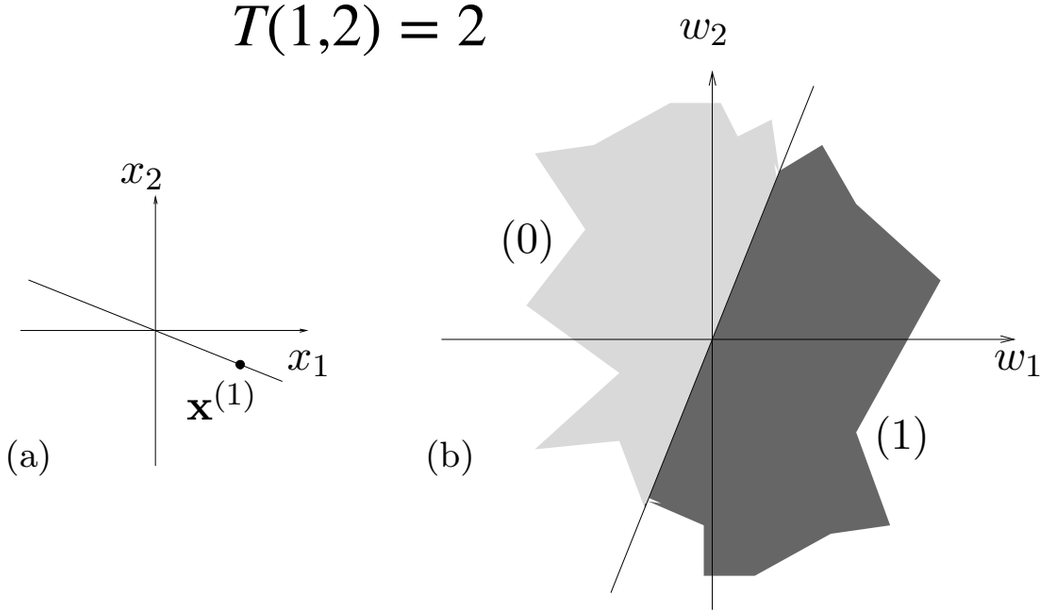



- Spin the separating line around the origin:
  - Cross one point at a time (general positions);
  - In one revolution, every point is passed over twice.

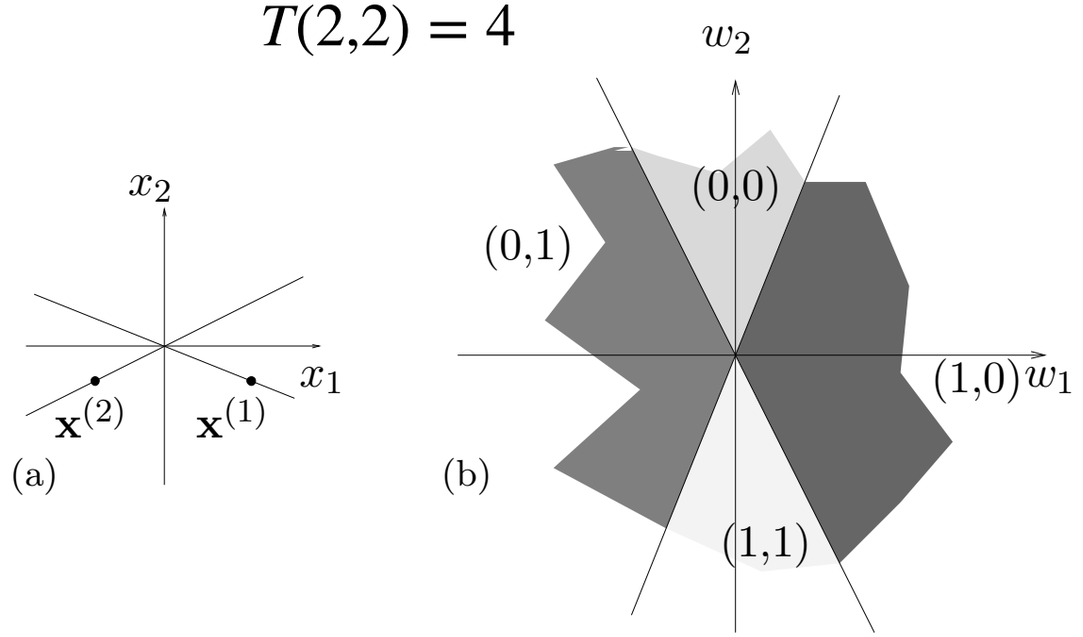
$\Rightarrow T(N,2) = 2N$ which is $< 2^N$ for $N \geq 3$ (not all binary functions can be realized by a linear threshold function).

# Counting T(N,2) in Weight Space

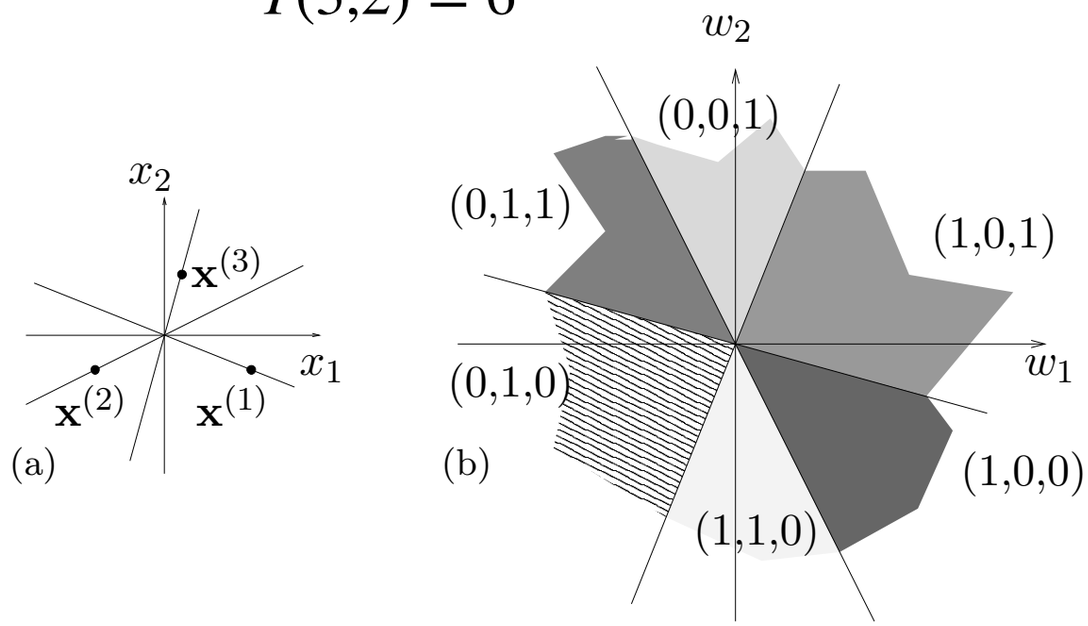
$T(1,2) = 2$




$T(2,2) = 4$




$T(3,2) = 6$

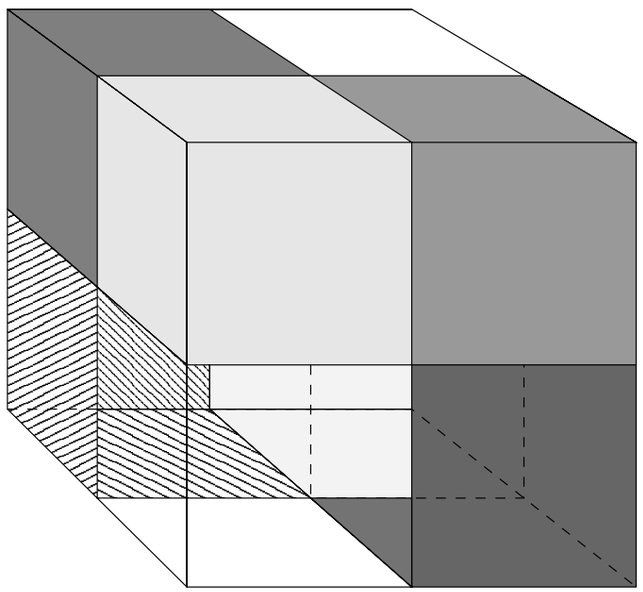



Each point defines a hyperplane in the weight space that produce 2 labelings:

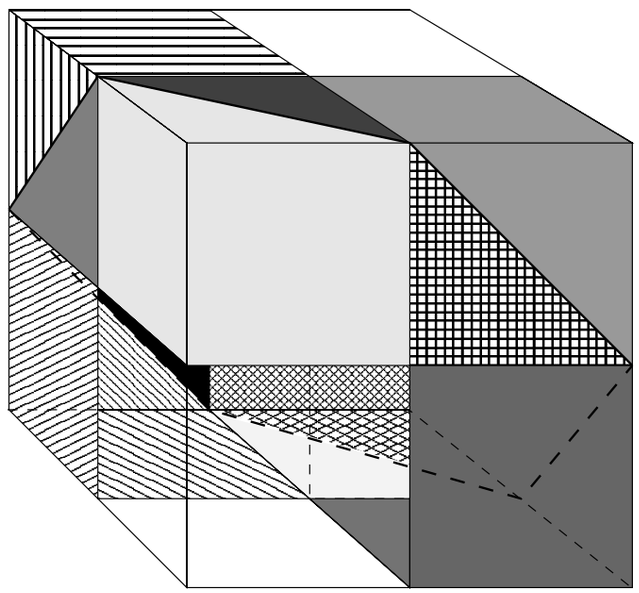
$$\mathbf{x}^{(n)} \cdot \mathbf{w} = 0$$

# Counting T(N,3) in Weight Space

$T(3,3) = 8$



$T(4,3) = 14$



| | $K$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $N$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 4 | 4 | | | | | |
| 3 | 2 | 6 | 8 | | | | | |
| 4 | 2 | 8 | 14 | | | | | |
| 5 | 2 | 10 | | | | | | |
| 6 | 2 | 12 | | | | | | |

# Recurrence Relation

- Adding an N-th hyperplane in K dimensions bisects T(N-1,K-1) of the T(N-1,K) regions that were created by the previous N-1 hyperplanes.

$$T(N,K) = 2T(N-1,K-1) + \left[T(N-1,K) - T(N-1,K-1)\right] = T(N-1,K) + T(N-1,K-1)$$

- subject to the boundary conditions: T(N,1)=2 and T(1,K)=2.

- The recurrence relation is satisfied by the Pascal's triangle:

$$C(N,K) \equiv \binom{N}{K} \equiv \frac{N!}{(N-K)!K!}.$$

| $N$ | $K$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 0 | 1 | | | | | | | |
| 1 | 1 | 1 | | | | | | |
| 2 | 1 | 2 | 1 | | | | | |
| 3 | 1 | 3 | 3 | 1 | | | | |
| 4 | 1 | 4 | 6 | 4 | 1 | | | |
| 5 | 1 | 5 | 10 | 10 | 5 | 1 | | |

- **Convention:** $C(N,K) \equiv 0$ for $K > N$ or $K < 0$.

# Solving the Recurrence Relation

- Even C(N,K) satisfies the same recurrence relation as T(N,K), it does not mean T(N,K)=C(N,K).

- Solution: T(N,K)=linear combination of C(N+$\alpha$, K + $\beta$) subject to boundary conditions: T(N,1)=2 and T(1,K)=2.

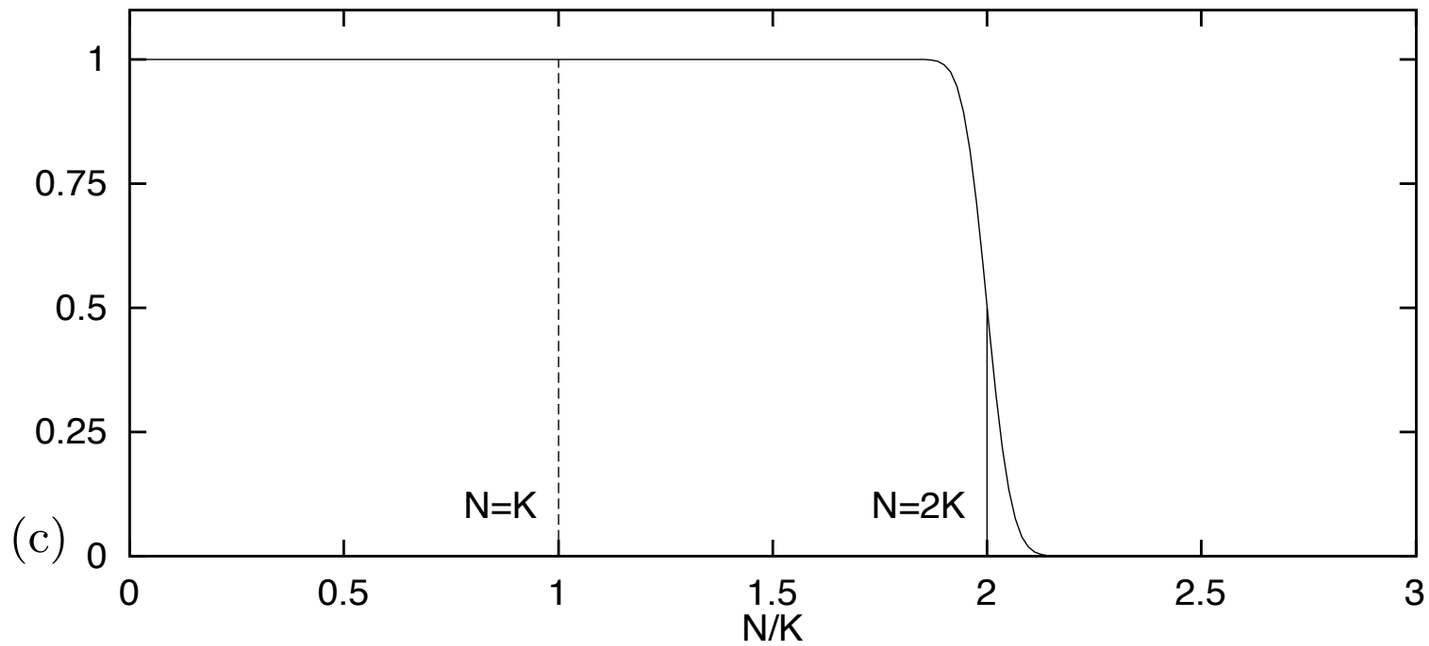
$$T(N,K) = 2\sum_{k=0}^{K-1}\binom{N-1}{k} = \begin{cases} 2^N & K \geq N \\ 2\sum_{k=0}^{K-1}\binom{N-1}{k} & K < N. \end{cases}$$

- equal to # of binary labelings $2^N$ for all N ≤ K

- **VC( Vapnik–Chervonenkis) dimension** of a class of functions = maximum # of points on which any arbitrary labelling can be realized.
- VC dimension of binary threshold function on K dimension = K.

# Interpretation

- $T(N,K)/2^N$ tells us the probability that an arbitrary labelling can be memorized by our neuron.




**The capacity of a linear threshold neuron, for large K, is 2 bits/weight.**

- A single neuron can almost certainly memorize up to N = 2K random binary labels perfectly, but will almost certainly fail to memorize more.

# Outline of the Course

**Where do we stand?**

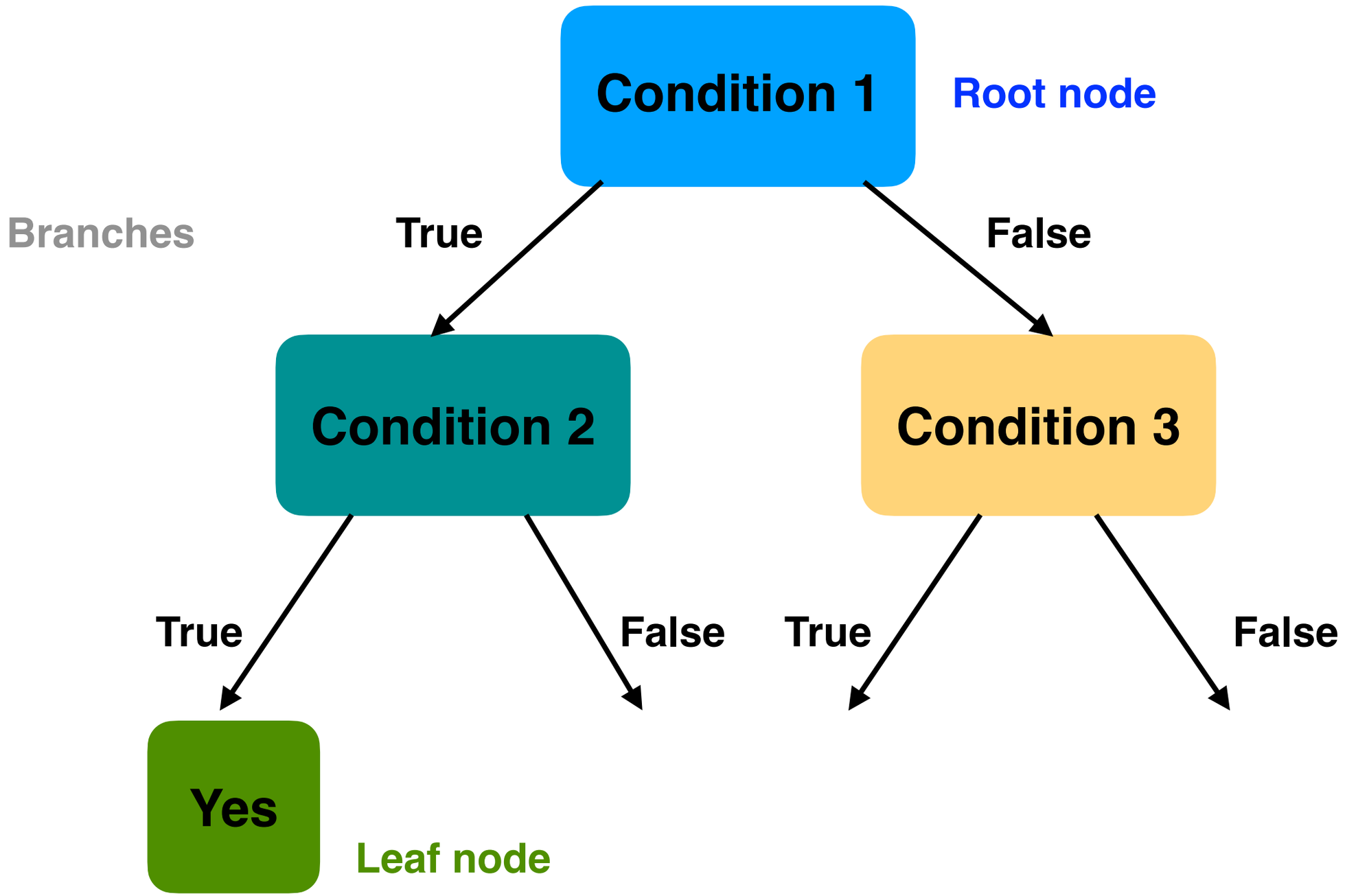
- Basic of Machine Learning
- Optimizers
- Regression
- Logistic/Multi-class classification
- A survey of classifiers
- Neural Networks
- Unsupervised learning
- Variational Methods
- Generative Adversarial Networks
- Normalizing Flows
- Reinforcement Learning
- Applications in Physics

# Survey of Classifiers

# Overview

- There are several alternatives to neural network classifiers, I will give a brief overview on how they work:
    - Decision Trees
    - Support Vector Machines
    - Bagging
    - Boosting
    - Random Forests
- Why? Because they work very well in several situations and are sometimes easier to interpret.

# Decision Trees



# Decision Trees

- Depth of tree = maximum number of splitting conditions.
- Stop growing the tree when 1) all items on a branch have the same features (values) or 2) other stopping criterion is met.
- Usually have maximum criterion to avoid overfitting.

- At each splitting node, look for features which provide the best splitting condition. How do we quantify best?
- **Maximize "information gain":**

$$\text{Gain}\ (S, \mathscr{A}) = \text{Entropy}\ (S) - \sum_{\nu \in \mathscr{A}} \frac{|S^{\nu}|}{|S|} \text{Entropy}\ (S^{\nu}); \qquad \text{Entropy}\ (S) = -\sum_{i} p_i \log_2 p_i$$

**set** **attributes** **size of set**

# Decision Trees: Example

| Day | Outlook | Temp. | Humidity | Wind | Go hiking? |
|---|---|---|---|---|---|
| 1 | Sunny | Hot | High | Weak | No |
| 2 | Sunny | Hot | High | Strong | No |
| 3 | Overcast | Hot | High | Weak | Yes |
| 4 | Rain | Mild | High | Weak | Yes |
| 5 | Rain | Cool | Normal | Weak | Yes |
| 6 | Rain | Cool | Normal | Strong | No |
| 7 | Overcast | Cool | Normal | Strong | Yes |
| 8 | Sunny | Mild | High | Weak | No |
| 9 | Sunny | Cool | Normal | Weak | Yes |
| 10 | Rain | Mild | Normal | Weak | Yes |
| 11 | Sunny | Mild | Normal | Strong | Yes |
| 12 | Overcast | Mild | High | Strong | Yes |
| 13 | Overcast | Hot | Normal | Weak | Yes |
| 14 | Rain | Mild | High | Strong | No |

# Decision Trees: Example

## What conditions should we pick?

| Day | Outlook | Temp. | Humidity | Wind | Go hiking? |
|---|---|---|---|---|---|
| 1 | Sunny | Hot | High | Weak | No |
| 2 | Sunny | Hot | High | Strong | No |
| 3 | Overcast | Hot | High | Weak | Yes |
| 4 | Rain | Mild | High | Weak | Yes |
| 5 | Rain | Cool | Normal | Weak | Yes |
| 6 | Rain | Cool | Normal | Strong | No |
| 7 | Overcast | Cool | Normal | Strong | Yes |
| 8 | Sunny | Mild | High | Weak | No |
| 9 | Sunny | Cool | Normal | Weak | Yes |
| 10 | Rain | Mild | Normal | Weak | Yes |
| 11 | Sunny | Mild | Normal | Strong | Yes |
| 12 | Overcast | Mild | High | Strong | Yes |
| 13 | Overcast | Hot | Normal | Weak | Yes |
| 14 | Rain | Mild | High | Strong | No |

$$\text{Gain}\ (S, \text{outlook})$$
$$= \text{Entropy}\ (S)$$
$$-\frac{5}{14}\ \text{Entropy}\ (S, \text{outlook} = \text{sunny})$$
$$-\frac{4}{14}\ \text{Entropy}\ (S, \text{outlook} = \text{outcast})$$
$$-\frac{5}{14}\ \text{Entropy}\ (S, \text{outlook} = \text{rain})$$
$$= 0.246$$

$$\text{Entropy}\ (S) = -\frac{5}{14}\log\frac{5}{14} - \frac{9}{14}\log\frac{9}{14} = 1.245$$

irrespective of outlook

$$\text{Entropy}\ (S, \text{outlook} = \text{sunny}) = -\frac{2}{5}\log\frac{2}{5} - \frac{3}{5}\log\frac{3}{5} = 0.971$$

2 yes and 3 no

$$\text{Entropy}\ (S, \text{outlook} = \text{overcast}) = 0$$

all yes

$$\text{Entropy}\ (S, \text{outlook} = \text{rain}) = -\frac{3}{5}\log\frac{3}{5} - \frac{2}{5}\log\frac{2}{5} = 0.971$$

3 yes and 2 no

# Decision Trees: Example

What conditions should we pick?

| Day | Outlook | Temp. | Humidity | Wind | Go hiking? |
|---|---|---|---|---|---|
| 1 | Sunny | Hot | High | Weak | No |
| 2 | Sunny | Hot | High | Strong | No |
| 3 | Overcast | Hot | High | Weak | Yes |
| 4 | Rain | Mild | High | Weak | Yes |
| 5 | Rain | Cool | Normal | Weak | Yes |
| 6 | Rain | Cool | Normal | Strong | No |
| 7 | Overcast | Cool | Normal | Strong | Yes |
| 8 | Sunny | Mild | High | Weak | No |
| 9 | Sunny | Cool | Normal | Weak | Yes |
| 10 | Rain | Mild | Normal | Weak | Yes |
| 11 | Sunny | Mild | Normal | Strong | Yes |
| 12 | Overcast | Mild | High | Strong | Yes |
| 13 | Overcast | Hot | Normal | Weak | Yes |
| 14 | Rain | Mild | High | Strong | No |

$$\text{Gain}\ (S, \text{Outlook}) = 0.246$$

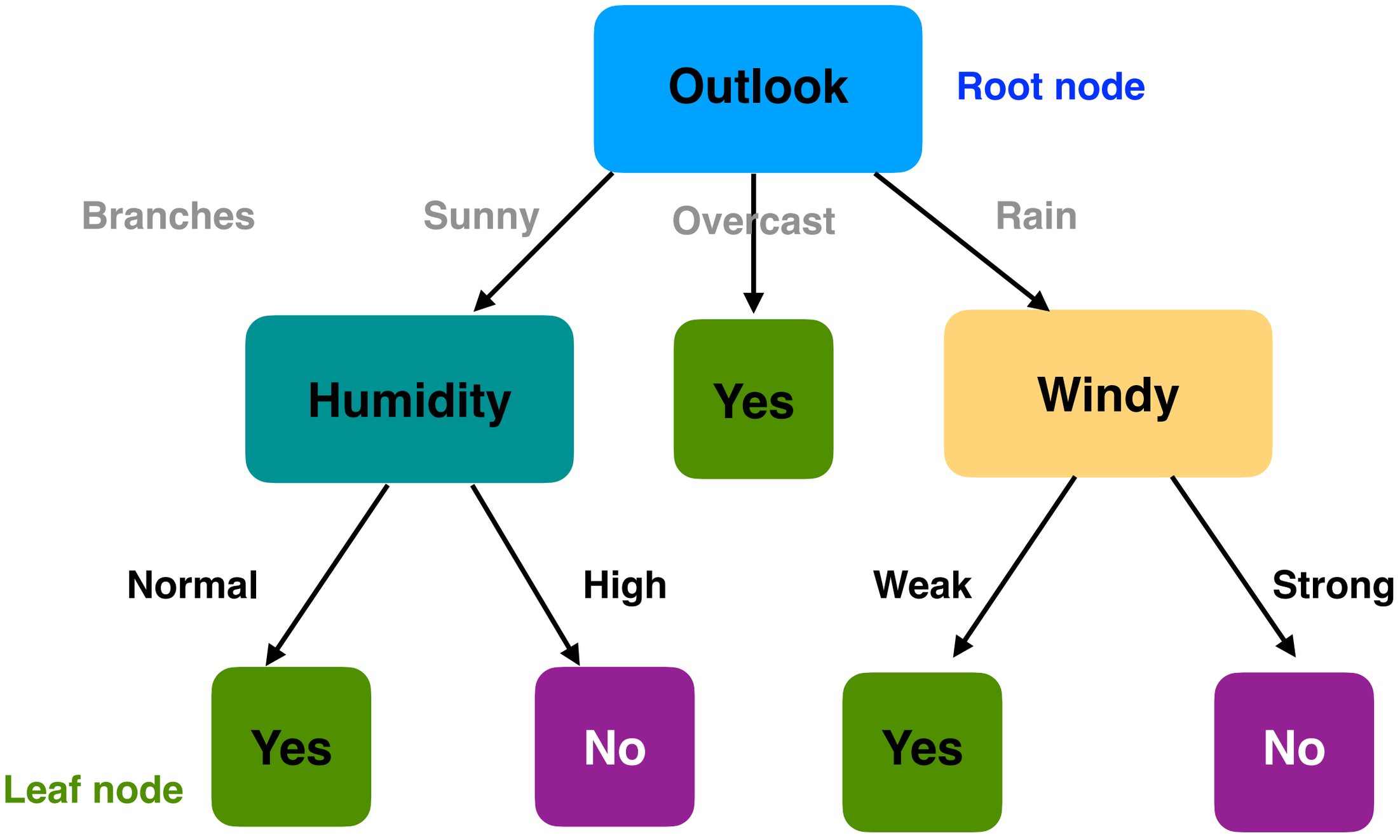
$$\text{Gain}\ (S, \text{Humidity}) = 0.151$$

$$\text{Gain}\ (S, \text{Wind}) = 0.048$$

$$\text{Gain}\ (S, \text{Temp.}) = 0.029$$

**⇒ Choose Outlook maximizes the information gain**

# Decision Trees: Hiking Example

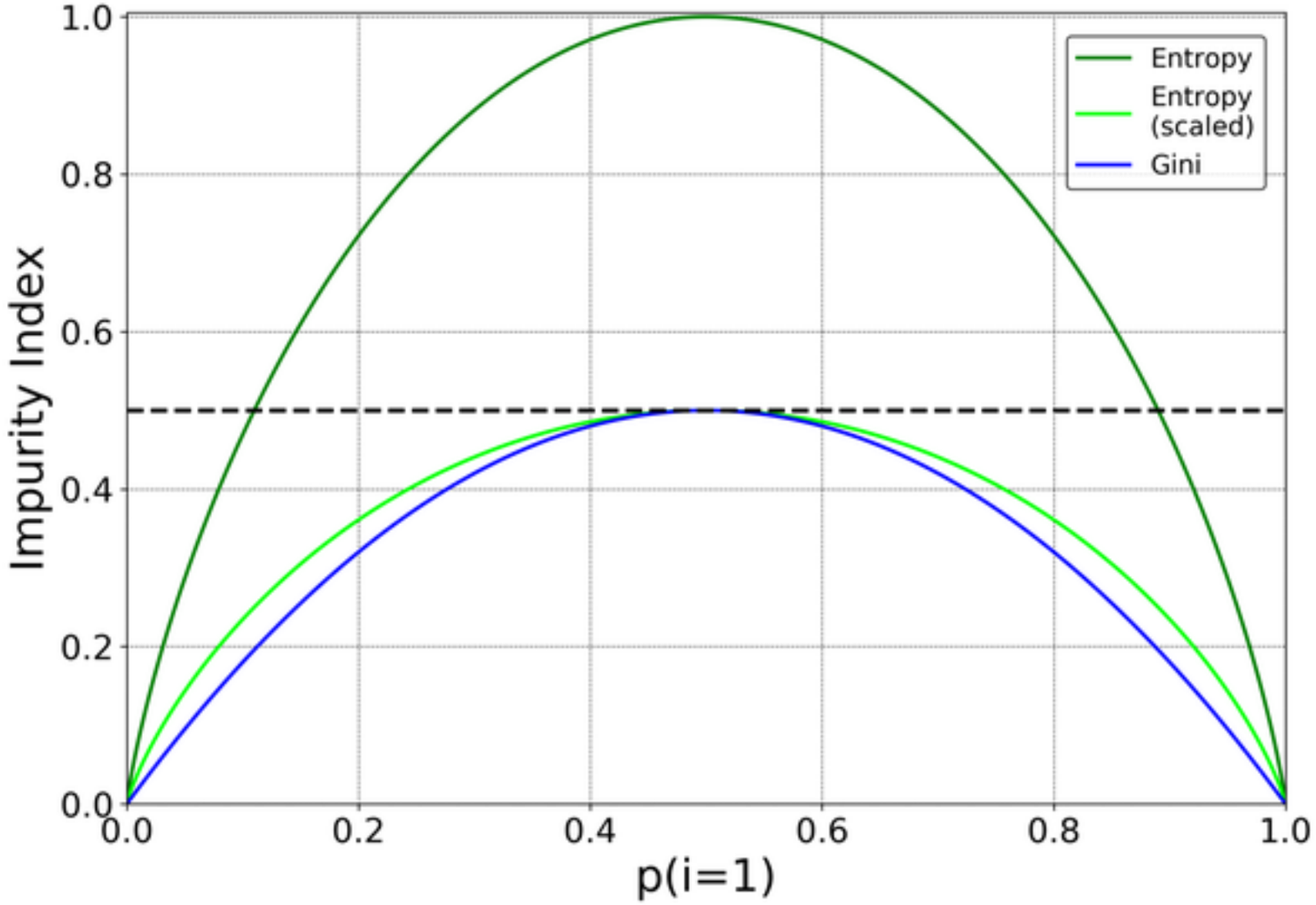


# Gini vs Entropy

$$\text{Entropy } (S) = -\sum_i p_i \log_2 p_i$$

$$\text{Gini } (S) = 1 - \sum_i p_i^2 = 1 - p^2 - (1-p)2 = -2p^2 + 2p$$

**two classes**




**options in sklearn implementation**

# Summary

- Capacity of Perceptron = 2 bits/weight
- Survey of classifiers
- Decision Trees
- How can use decision tree for regression?