- Learning
  - Decision trees
    - Building them
    - Building good ones
  - Sub-symbolic learning
    - Neural networks

Decision trees: issues

- Constructing a decision tree is easy… really easy!
  - Just add examples in turn.


- Difficulty: how can we extract a ***simplified*** decision tree?
  - This implies (among other things) establishing a preference order (bias) among alternative decision trees.
  - Finding the smallest one proves to be VERY hard. Improving over the trivial one is okay.

Office size example

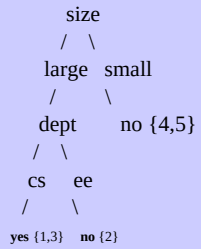
Training examples:

1. large ^ cs ^ faculty -> yes
2. large ^ ee ^ faculty -> no
3. large ^ cs ^ student -> yes
4. small ^ cs ^ faculty -> no
5. small ^ cs ^ student -> no

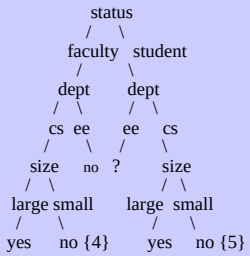
The questions about office size, department and status tells use something about the mystery attribute.

Let's encode all this as a decision tree.

Decision tree #1



Decision tree #2



Making a tree

How can we build a decision tree (that might be good)?

Objective: an algorithm that builds a decision tree from the root down.
Each node in the decision tree is associated with a set of training examples that are split among its children.

- Input: a node in a decision tree with no children and associated with a set of training examples
- Output: a decision tree that classifies all of the examples i.e., all of the training examples stored in each leaf of the decision tree are in the same class

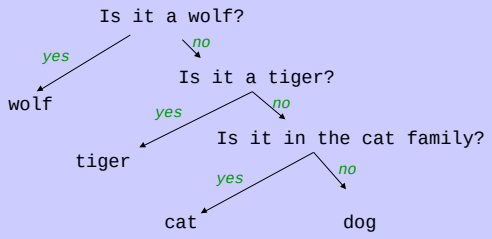
Procedure: Buildtree

If all of the training examples are in the same class,
then quit,
else 1. Choose an attribute to split the examples.
2. Create a new child node for each value of the attribute.
3. Redistribute the examples among the children according to the attribute values.
4. Apply buildtree to each child node.

Is this a good decision tree? Maybe? How do we decide?

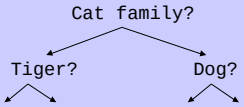
A "Bad" tree

- To identify an animal (goat,dog,housecat,tiger)



- Max depth 3.
- To get to fish or goat, it takes three questions.ions.
- In general, a bad tree for N categories can take N questions.

- Can't we do better? A good tree?



- Max depth 2 questions.
  - More generally, $\log_2(N)$ questions.

Best Property

- Need to select property / feature / attribute
- Goal: find short tree (Occam's razor)
  1. Base this on MAXIMUM depth
  2. Base this on the AVERAGE depth
     A) over all leaves
     B) over all queries
- select most informative feature
  - One that best splits (classifies) the examples
- Use measure from **information theory**
  - Claude Shannon (1949)

Optimizing the tree

All based on *buildtree.*

To minimize maximum depth, we want to build a balanced tree.

- Put the training set (TS) into any order.
- For each question Q
  - Construct a K-tuple of 0s and 1s
    - The jth entry in the tuple is
      - 1 if the jth instance in the TS has answer YES to Q
      - 0 if it has answer NO
    - Discard al questions of only one answer (all 0 or 1's).

Min Max Depth

- Minimize max depth:
- At each query, come as close as possible to cutting the number of samples in the subtree in half.
- This suggests the number of questions per subtree is given by the $\log_2$ of the number of sample categories to be subdivided.
  - Why $\log_2$??

Entropy

Measures the (im) purity in collection S of examples

Entropy(S) =

$$- [\ p_+ \log_2 (p_+) + p_- \log_2 (p_-)\ ]$$

- p_+ is the proportion of positive examples.
- p_- is the proportion of negative examples.

N.B. This is not a fully general definition of entropy.

Example

- S, 14 examples, 9 positive, 5 negative

Entropy([9+,5-]) =

$$-(9/14) \log_2(9/14) - (5/14)\log_2(5/14) = 0.940$$

Intuition / Extremes

- Entropy in collection is zero if all examples in same class.
- Entropy is 1 if equal number of positive and negative examples.

Intuition:

```
If you pick random example, how
  many bits do you need to
  specify what class the example
  belongs too?
```


Entropy: definition

- Often referred to a “randomness”.
- How useful is a question:
  - How much guessing does knowing an answer save?
- How much “surprise” value is there in a question.

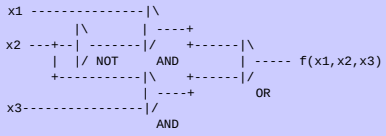
Information Gain

General definition

- Entropy(S) =

$$\sum_{1}^{c} p_i \log_2 (p_i)$$

- In this lecture we consider some alternative hypothesis spaces based on continuous functions. Consider the following boolean circuit.



The topology is fixed and logic elements are fixed so there is a single Boolean function.

Is there a fixed topology that can be used to represent a family of functions?

Yes! Neural-like networks (aka artificial neural networks) allow us this flexibility and more; we can represent arbitrary families of continuous functions using fixed topology networks.

The idealized neuron

- Artificial neural networks come in several “flavors”.
  - Most of based on a simplified model of a neuron.
- A set of (many) inputs.
- One output.
- Output is a function of the sum on the inputs.
  - Typical functions:
    - Weighted sum
    - Threshold
    - Gaussian