maj vote

When all fails what should be the answer.

Termination condition

Decision Tree Learning Algorithm

function DECISION-TREE-LEARNING(*examples*, *attributes*, *default*) **returns** a decision tree
inputs: *examples*, set of examples
attributes, set of attributes
default, default value for the goal predicate

if *examples* is empty **then return** *default*
else if all *examples* have the same classification **then return** the classification
else if *attributes* is empty **then return** MAJORITY-VALUE(*examples*)
else
best ← CHOOSE-ATTRIBUTE(*attributes*, *examples*)
tree ← a new decision tree with root test *best*
for each value v_i of *best* **do**
$examples_i$ ← {elements of *examples* with *best* = v_i}
subtree ← DECISION-TREE-LEARNING($examples_i$, *attributes* − *best*,
MAJORITY-VALUE(*examples*))
add a branch to *tree* with label v_i and subtree *subtree*
end
return *tree*

DTR(, {a1, a2}, +)

Then this will return negative

maj vote

When all fails
what should be
the answer.

Termination
condition

Decision Tree Learning Algorithm

```
function DECISION-TREE-LEARNING(examples, attributes, default) returns a decision tree
  inputs: examples, set of examples
          attributes, set of attributes
          default, default value for the goal predicate

  if examples is empty then return default
  else if all examples have the same classification then return the classification
  else if attributes is empty then return MAJORITY-VALUE(examples)
  else
     best ← CHOOSE-ATTRIBUTE(attributes, examples)
     tree ← a new decision tree with root test best
     for each value v_i of best do
        examples_i ← {elements of examples with best = v_i}
        subtree ← DECISION-TREE-LEARNING(examples_i, attributes − best,
                                          MAJORITY-VALUE(examples))
        add a branch to tree with label v_i and subtree subtree
     end
     return tree
```

DTR({a1, a2}, + }

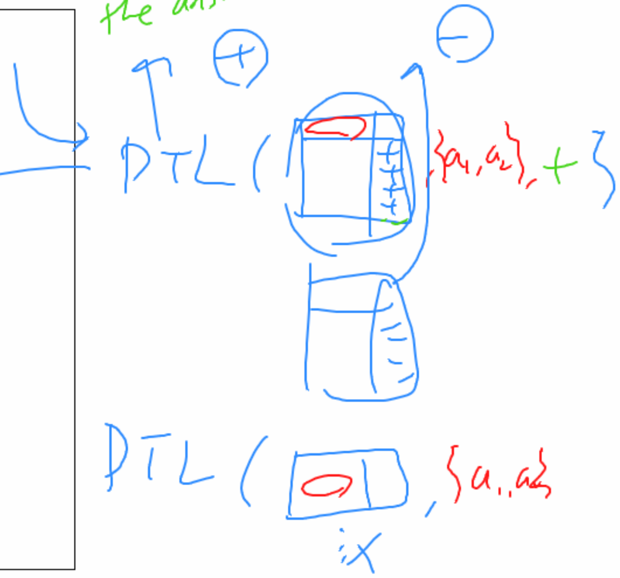
Then this will return negative

when all fails
what should be
the answer.

Termination
condition

Decision Tree Learning Algorithm

```
function DECISION-TREE-LEARNING(examples, attributes, default) returns a decision tree
  inputs: examples, set of examples
          attributes, set of attributes
          default, default value for the goal predicate

  if examples is empty then return default
  else if all examples have the same classification then return the classification
  else if attributes is empty then return MAJORITY-VALUE(examples)
  else
      best ← CHOOSE-ATTRIBUTE(attributes, examples)
      tree ← a new decision tree with root test best
      for each value v_i of best do
          examples_i ← {elements of examples with best = v_i}
          subtree ← DECISION-TREE-LEARNING(examples_i, attributes − best,
                                           MAJORITY-VALUE(examples))
          add a branch to tree with label v_i and subtree subtree
      end
      return tree
```



And then you have well samples. Okay, then, just return the majority value. right? So

DTL(, {a1, a2, .. an} , ϕ)

{a1 a2 . ai .. an}

best

{a1 a2 . . ai-1, ai+1, . . an}

best attr.

best

v1, v2, v3

Decision Tree Learning Algorithm

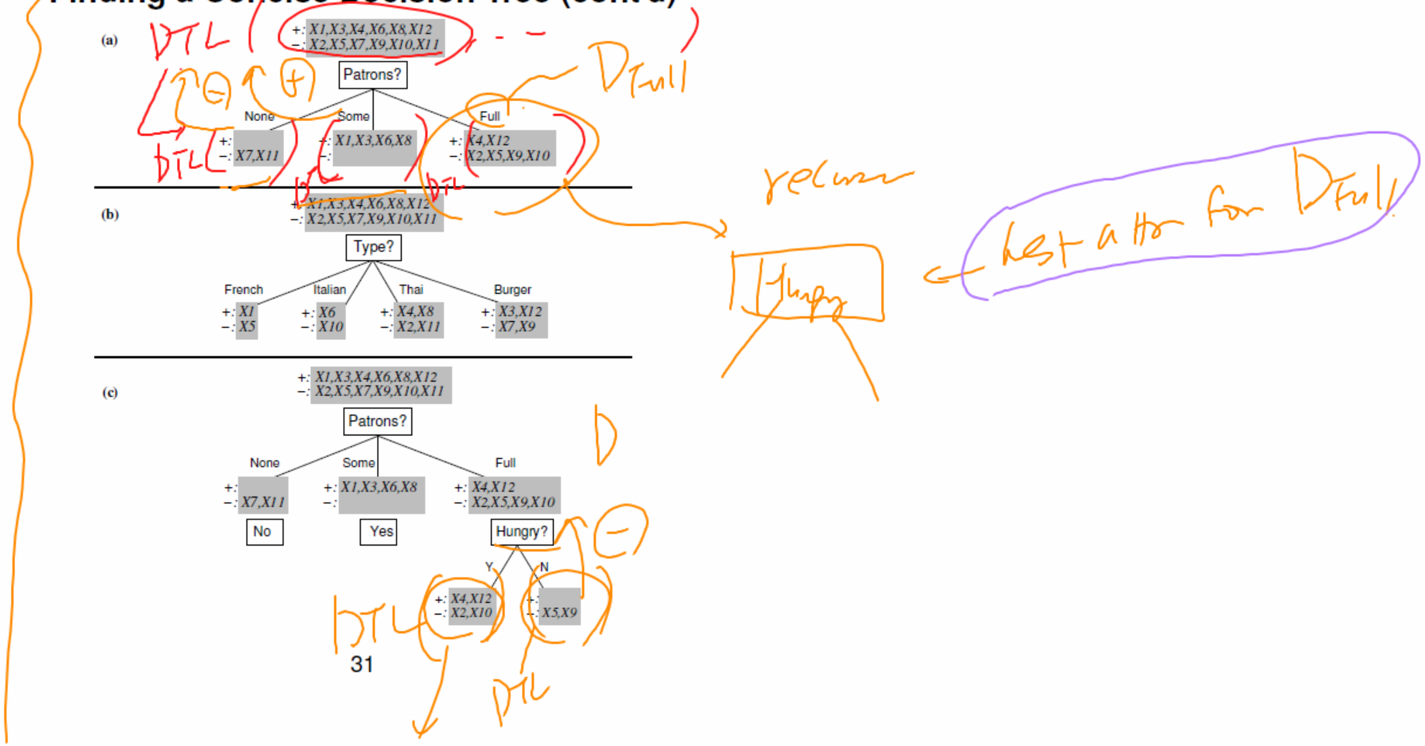
```
function DECISION-TREE-LEARNING(examples, attributes, default) returns a decision tree
   inputs: examples, set of examples
           attributes, set of attributes
           default, default value for the goal predicate

   if examples is empty then return default
   else if all examples have the same classification then return the classification
   else if attributes is empty then return MAJORITY-VALUE(examples)
   else
      best ← CHOOSE-ATTRIBUTE(attributes, examples)
      tree ← a new decision tree with root test best
      for each value v_i of best do
         examples_i ← {elements of examples with best = v_i}
         subtree ← DECISION-TREE-LEARNING(examples_i, attributes − best,
                                           MAJORITY-VALUE(examples))
         add a branch to tree with label v_i and subtree subtree
      end
      return tree
```

find the subsets based on attr value.

DTL( ) DTL( ) DTL(,)

subtree subtree subtree

Finding a Concise Decision Tree (cont'd)



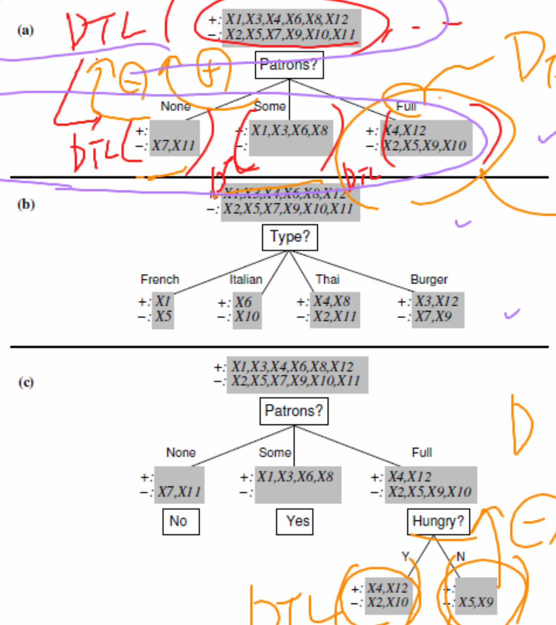
So what the only thing that we remains to be extended is, how, this best attribute is selected.

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save

Who can see what you share here? Recording On

Initial Uncertainty

Finding a Concise Decision Tree (cont'd)



Find. Uncertainty

return

Hungry

best attr for Dfull

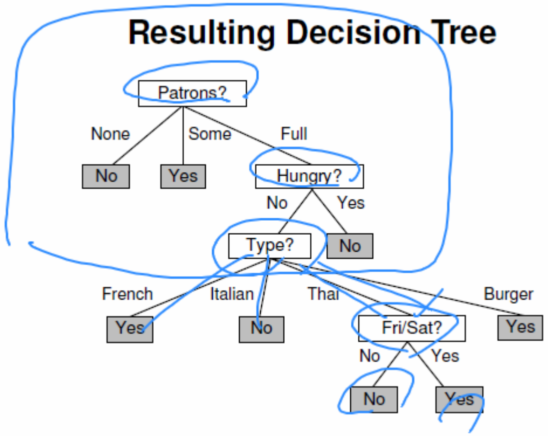
31

Here the uncertainty remains. so how can you quantify that

You are screen sharing

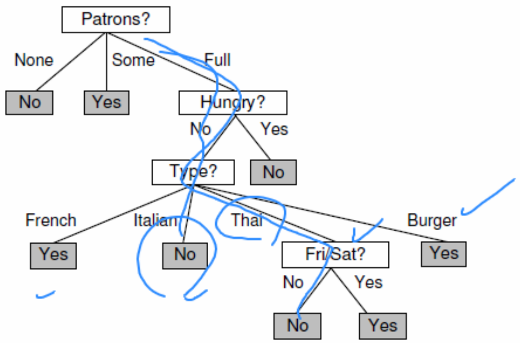
Stop Share

Resulting Decision Tree



- Some attributes are not tested at all.
- Odd paths can be generated (Thai food branch).
- Sometimes the tree can be incorrect for new examples (exceptional cases).

Resulting Decision Tree



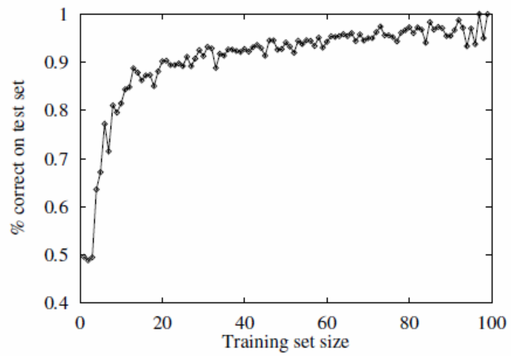
- Some attributes are not tested at all.
- Odd paths can be generated (Thai food branch).
- Sometimes the tree can be incorrect for new examples (exceptional cases).

So if get a new sample then following through this let's say you came here then, maybe the correct answer is Yes.

training set → used for training

test set

Accuracy of Decision Trees

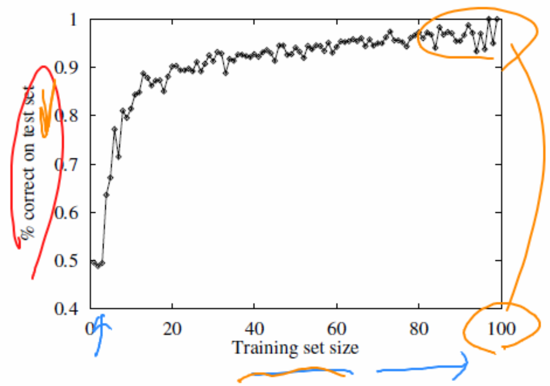


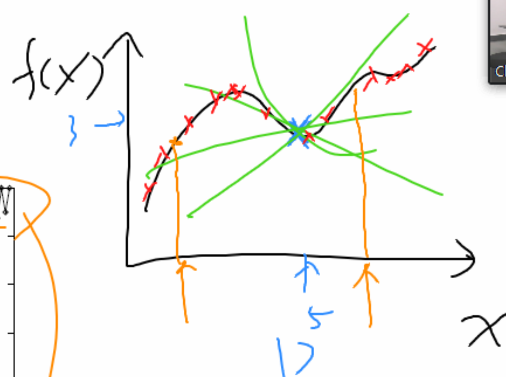
- Divide examples into training and test sets.
- Train using the training set.
- Measure accuracy of resulting decision tree on the test set.

training set → used for training

test set → generalization performance

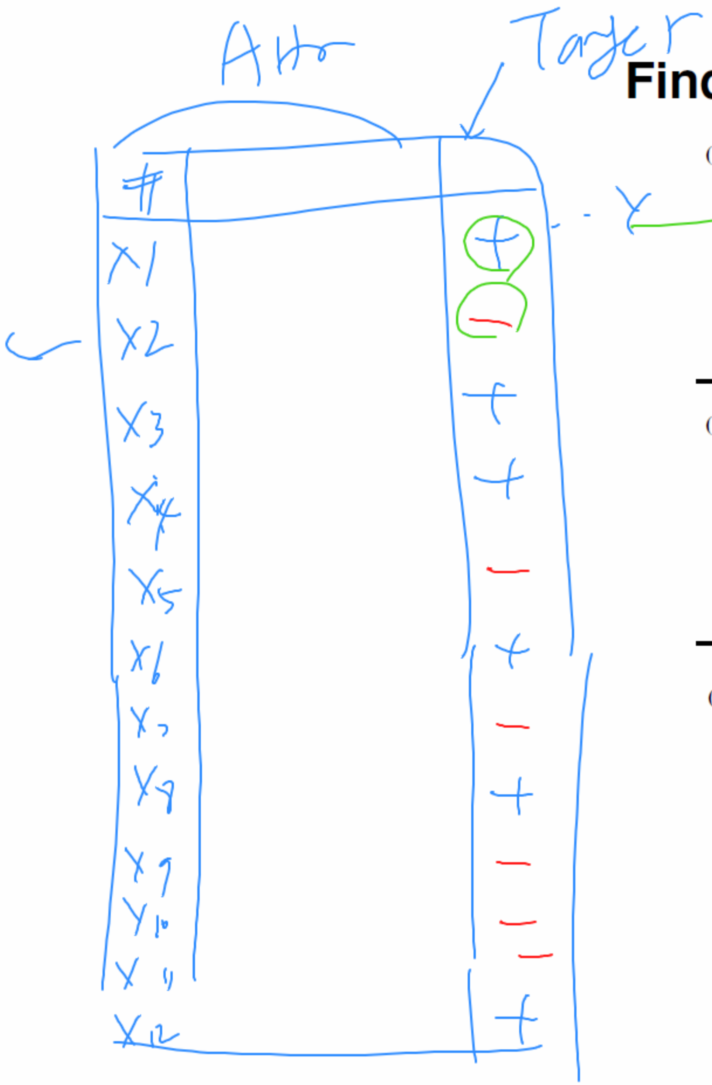
Accuracy of Decision Trees



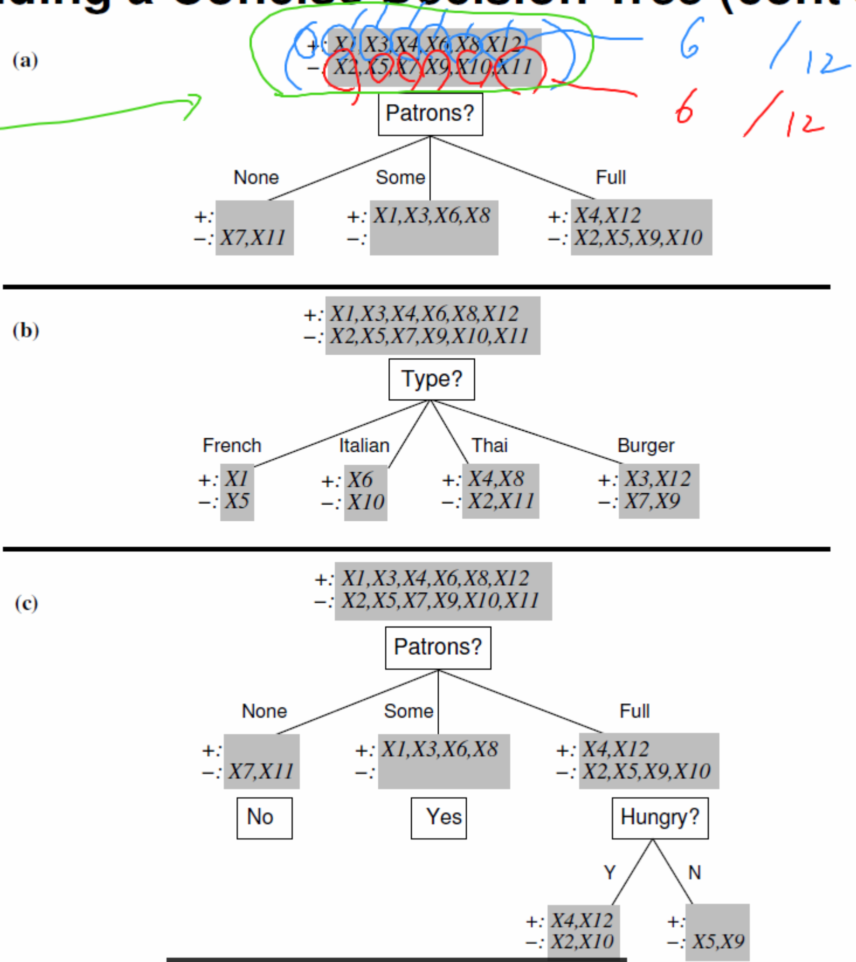


x	f
5	3

- Divide examples into training and test sets.
- Train using the training set.
- Measure accuracy of resulting decision tree on the test set.



Finding a Concise Decision Tree (cont'd)



So that's how we got this representation even just the plus minus positive method class

Choosing the Best Attribute to Test First

Use Shannon's information theory to choose the attribute that give the maximum **information gain**.

- Pick an attribute such that the information gain (or entropy reduction) is maximized. average uncertainty .
- Entropy measures the **average surprisal** of events. Less probable events are more surprising.

probability .

$X \in \{T/F\}$

maximum uncertainty

random variable.

$P(X=T) = 0.5$
$P(X=F) = 0.5$

Entropy and Information Gain

degree of uncertainty.

$$Entropy(E) = \sum_{i \in C} -P_i log_2(P_i)$$

$$Gain(E, A) = Entropy(E) - \sum_{v \in Values(A)} \frac{|E_v|}{|E|} Entropy(E_v)$$

T F F T T T F T F T T F F

- E: set of examples: $\langle x, f(x) \rangle$ pairs
- C: set of output classes
- A: a single attribute
- E_v: set of examples where attribute $A = v$.
- $|S|$: cardinality of set S.

$P(X=T) = 1.0$
$P(X=F) = 0.0$

T

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save

Who can see what you share here? Recording On

$X \in \{T/F\}$

maximum uncertainty

$P(X=T) = 0.5$

$P(X=F) = 0.5$

random variable.

~~Uncertainty~~ $(P(+), P(-))$

Entropy and Information Gain

degree of uncertainty.

$$Entropy(E) = \sum_{i \in C} -P_i log_2(P_i)$$

$$Gain(E, A) = Entropy(E) - \sum_{v \in Values(A)} \frac{|E_v|}{|E|} Entropy(E_v)$$

- E: set of examples: $\langle x, f(x) \rangle$ pairs
- C: set of output classes
- A: a single attribute
- E_v: set of examples where attribute $A = v$.
- $|S|$: cardinality of set S.

T F F T T T F T F T T F F −

$P(X=T) = 1.0$

$P(X=F) = 0.0$

T T T T T T T T T T T

→ no uncertainty

$P(+)$

$P(+) + P(-) = 1.$

$P(+)$

$P(-) = 1 - P(+)$

$P(-)$

$P(X=T) = 0.0$

$P(X=F) = 1.0$

F F F ... F F

?

Then you can represent it as a function of p plus, because you can just call this

You are screening sharing Stop Share

Entropy and Information Gain

$$Entropy(E) = \sum_{i \in C} -P_i log_2(P_i)$$

$$Gain(E, A) = Entropy(E) - \sum_{v \in Values(A)} \frac{|E_v|}{|E|} Entropy(E_v)$$

- E: set of examples: $\langle x, f(x) \rangle$ pairs
- C: set of output classes
- A: a single attribute
- E_v: set of examples where attribute $A = v$.
- $|S|$: cardinality of set S.

$X \in \{T/F\}$

$P(X=T) = 0.5$

$P(X=F) = 0.5$

maximum uncertainty

random variable.

degree of uncertainty

max

~~Uncertainty(P(+), P(-))~~

Uncertainty(P(+), 1-P(+))

T F F T T T F T F T T F F F

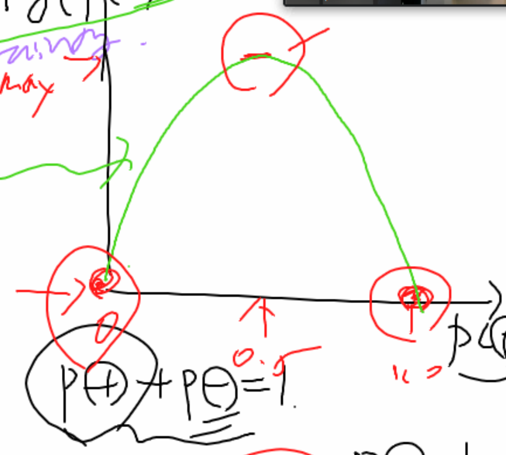
$P(+) = 1.0$

$P(X=T) = 1.0$

$P(X=F) = 0.0$

T T T T T T T T T T T

→ no uncertainty



$P(+)$

$P(+) + P(-) = 1$

"0.5"

$P(+)$

$P(-) = 1 - P(+)$

$P(-)$

$P(X=T) = 0.0$

$P(X=F) = 1.0$

?

F F F ... F F

So let's say

You are screen sharing

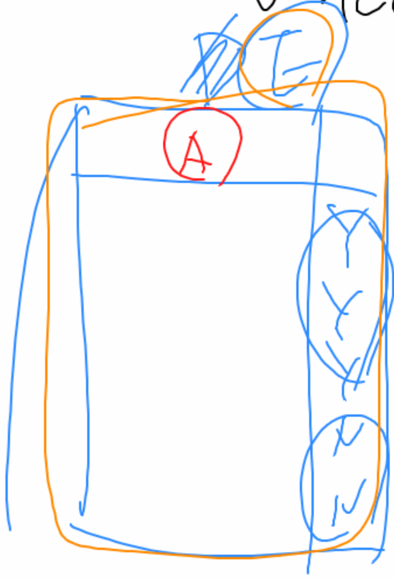
Stop Share

Who can see what you share here? Recording On

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save

Uncertainty $(P_\oplus, P_\ominus)$

$P_i =$



$P_\oplus = 3/5$
P_Y

$P_\ominus = 2/5$
P_N

$C = \{Y, N\}$

Entropy and Information Gain

dataset

$$Entropy(E) = \sum_{i \in C} -P_i log_2(P_i)$$

outcomes

$$Gain(E, A) = Entropy(E) - \sum_{v \in Values(A)} \frac{|E_v|}{|E|} Entropy(E_v)$$

- E: set of examples: $\langle x, f(x) \rangle$ pairs
- C: set of output classes
- A: a single attribute
- E_v: set of examples where attribute $A = v$.
- $|S|$: cardinality of set S.

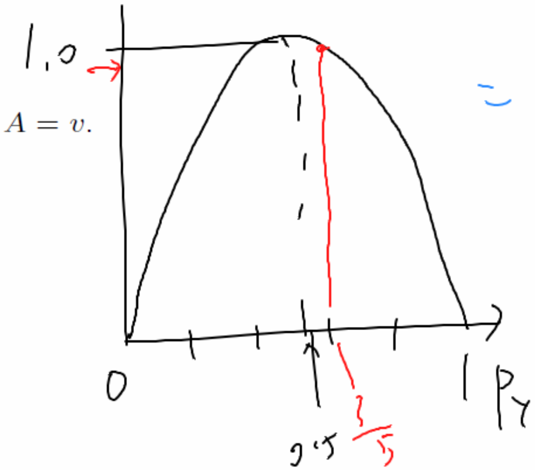
$$\sum_{i \in \{Y,N\}} -P_i \log_2(P_i)$$

$$= -P_Y \log_2 P_Y - P_N \log_2 P_N$$

$$= -\frac{3}{5} \cdot \log_2 \frac{3}{5} - \frac{2}{5} \cdot \log_2 \frac{2}{5} = 1.0$$

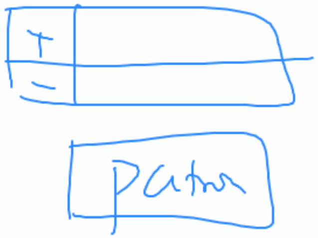
0.81?

when $P_Y = P_N = 0.5$



36

But anyway.



Data set

Prob of outcome class (+, −) (Y, N) ⋮

Entropy and Information Gain

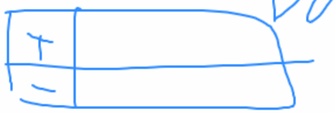
single attribute

$$Entropy(E) = \sum_{i \in C} -P_i log_2(P_i)$$

$$Gain(E, A) = Entropy(E) - \sum_{v \in Values(A)} \frac{|E_v|}{|E|} Entropy(E_v)$$

before after

- E: set of examples: $\langle x, f(x) \rangle$ pairs
- C: set of output classes
- A: a single attribute
- E_v: set of examples where attribute $A = v$.
- $|S|$: cardinality of set S.



Dataset E

single attribute

Dataset

Prob of outcome class (+, -) (Y, N)

Entropy and Information Gain

$$Entropy(E) = \sum_{i \in C} -P_i log_2(P_i)$$

$$Gain(E, A) = Entropy(E) - \sum_{v \in Values(A)} \frac{|E_v|}{|E|} Entropy(E_v)$$

before

after

{None, Some, Full}

- E: set of examples: $\langle x, f(x) \rangle$ pairs
- C: set of output classes
- A: a single attribute
- E_v: set of examples where attribute $A = v$.
- $|S|$: cardinality of set S.

Is none, some full

Entropy

+	n = 6
−	n = 6

Dataset E = 1.0 ← max value

|E| = 12

Patron

None Some Full

+	
−	n = 2

+	n = 4
−	

+	n = 2
−	n = 4

Ev = None Ev = Some Ev = Full

|Ev = None| = 2

|Ev = Some| = 4

|Ev = Full| = 6

Entropy and Information Gain

Dataset

Prob of outcome

class (+, −)

(Y, N)

single attribute

$$Entropy(E) = \sum_{i \in C} -P_i \log_2(P_i)$$

{2, 4, 6}

$$Gain(E, A) = Entropy(E) - \sum_{v \in Values(A)} \frac{|E_v|}{|E|} Entropy(E_v)$$

12

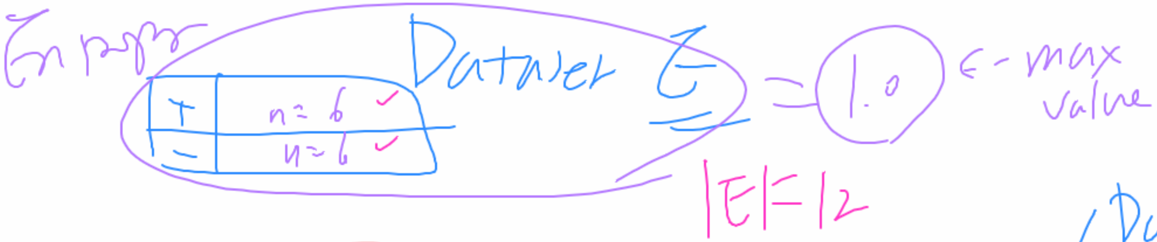
before

after

{None, Some, Full}

- E: set of examples: $\langle x, f(x) \rangle$ pairs
- C: set of output classes
- A: a single attribute
- E_v: set of examples where attribute $A = v$.
- $|S|$: cardinality of set S.

I said, this is 12, and then this could be either 2, 4, or 6, depending on the Dv.



Dataset

Prob of outcome

class (+, −)
(Y, N)

Entropy and Information Gain

single attribute

$$Entropy(E) = \sum_{i \in C} -P_i \log_2(P_i)$$

{2, 4, 6}

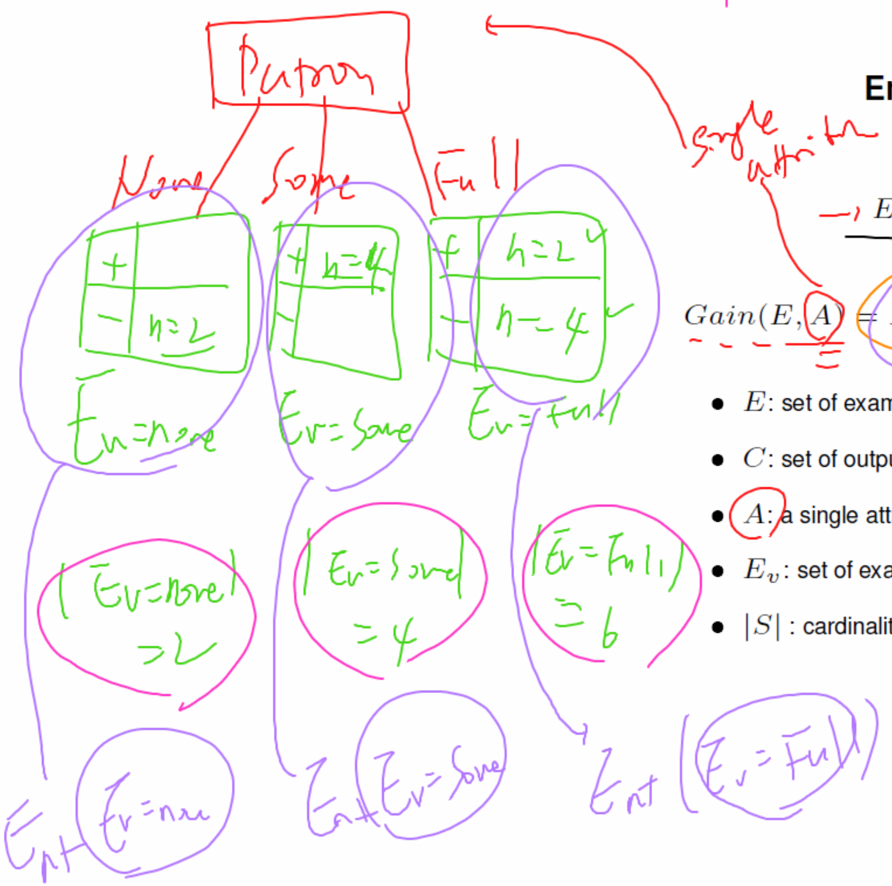
$$Gain(E, A) = Entropy(E) - \sum_{v \in Values(A)} \frac{|E_v|}{|E|} Entropy(E_v)$$

12

before

after

{None, Some, Full}



- E: set of examples: $\langle x, f(x) \rangle$ pairs
- C: set of output classes
- A: a single attribute
- E_v: set of examples where attribute $A = v$.
- $|S|$: cardinality of set S.

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save

Who can see what you share here? Recording On

Choe, Yoonsuck

Entropy Dataset E = 1.0 ← max value

$|E| = 12$

Patron

None Some Full

E_v=none E_v=Some E_v=full

$|E_v = \text{none}| = 2$ $|E_v = \text{Some}| = 4$ $|E_v = \text{Full}| = 6$

Ent(E_v=none) Ent(E_v=Some) Ent(E_v=Full)

2/12 4/12 6/12 ← 2

single attribute

Dataset

Prob of outcome

class (+, −)

(Y, N)

Entropy and Information Gain

$$Entropy(E) = \sum_{i \in C} -P_i \log_2(P_i)$$

$$Gain(E, A) = Entropy(E) - \sum_{v \in Values(A)} \frac{|E_v|}{|E|} Entropy(E_v)$$

before after

{None, Some, Full}

- E: set of examples: $\langle x, f(x) \rangle$ pairs
- C: set of output classes
- A: a single attribute
- E_v: set of examples where attribute $A = v$.
- $|S|$: cardinality of set S.

So this ratio would be 6 over 12, so this would be number 2, which would be this particular zinc. Now

You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save

Who can see what you share here? Recording On

Attr Target

Constructing Decision Trees from Examples

Example	Attributes										Goal
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0–10*	*Yes*
X_2	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30–60*	*No*
X_3	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0–10*	*Yes*
X_4	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*10–30*	*Yes*
X_5	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	*No*
X_6	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0–10*	*Yes*
X_7	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0–10*	*No*
X_8	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0–10*	*Yes*
X_9	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	*No*
X_{10}	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10–30*	*No*
X_{11}	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0–10*	*No*
X_{12}	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30–60*	*Yes*

- Given a set of examples (**training set**), both **positive** and **negative**, the task is to construct a decision tree that describes a concise decision path.
- Using the resulting decision tree, we want to **classify** new instances of examples (either as **yes** or **no**).

category

You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save

Who can see what you share here? Recording On

Entropy

Dataset E = 1.0 ← max value

$|E| = 12$

Patron

None Some Full

E_v=none E_v=Some E_v=full

$|E_v$=none$| = 2$ $|E_v$=Some$| = 4$ $|E_v$=Full$| = 6$

Ent(E_v=none) Ent(E_v=Some) Ent(E_v=Full)

2/12 4/12 6/12

single attribute

Dataset

Prob of outcome

class (+, −) (Y, N)

Entropy and Information Gain

$$Entropy(E) = \sum_{i \in C} -P_i log_2(P_i)$$

$$Gain(E, A) = Entropy(E) - \sum_{v \in Values(A)} \frac{|E_v|}{|E|} Entropy(E_v)$$

{2, 4, 6}

before

after

- E: set of examples: $\langle x, f(x) \rangle$ pairs
- C: set of output classes
- A: a single attribute
- E_v: set of examples where attribute $A = v$.
- $|S|$: cardinality of set S.

{None, Some, Full}

Resulting Entropy

Σ ratio × subset's Entropy

resulting entropy.

You are screen sharing Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save

Who can see what you share here? Recording On

Entropy

Dataset E = 1.0 ← max value

|E| = 12

Patron

None Some Full

E_v = None E_v = Some E_v = Full

$|E_v = None| = 2$ $|E_v = Some| = 4$ $|E_v = Full| = 6$

Ent(E_v = None) Ent(E_v = Some) Ent(E_v = Full)

2/12 4/12 6/12

single attribute

Dataset

Prob of outcome

class (+, −)

(Y, N)

Entropy and Information Gain

$$Entropy(E) = \sum_{i \in C} -P_i \log_2(P_i)$$

$$Gain(E, A) = Entropy(E) - \sum_{v \in Values(A)} \frac{|E_v|}{|E|} Entropy(E_v)$$

before after

- E: set of examples: $\langle x, f(x) \rangle$ pairs
- C: set of output classes
- A: a single attribute
- E_v: set of examples where attribute $A = v$.
- $|S|$: cardinality of set S.

{None, Some, Full}

Resulting Entropy.

Σ ratio × Subset's Entropy

Man. Finally, Finally, when you subtract this without the entropy from the initial entropy, then that gives you the information gain

You are screen sharing Stop Share

+	4
−	4

Ent( ) = 1.0 — 1.0

|E| = 8

Attr

Cold | Hot

+	
−	4

+	4
−	0

|E_hot| = 4 |E_cold| = 4

$\frac{4}{8}$ Ent( ) = 0.0 $\frac{4}{8}$ Ent( ) = 0.0

Entropy and Information Gain

$$Entropy(E) = \sum_{i \in C} -P_i log_2(P_i)$$

$$Gain(E, A) = Entropy(E) - \sum_{v \in Values(A)} \frac{|E_v|}{|E|} Entropy(E_v)$$

- E: set of examples: $\langle x, f(x) \rangle$ pairs
- C: set of output classes
- A: a single attribute
- E_v: set of examples where attribute $A = v$.
- $|S|$: cardinality of set S.

$1.0 - \left(\frac{4}{8} \cdot 0 + \frac{4}{8} \cdot 0 \right) = 1.0$

0

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save

Who can see what you share here? Recording On

Ent( ) = 1.0 − 1.0

|E| = 8

Attr

Cold Hot

|E_hot| = 4

|E_cold| = 4

4/8 Ent( ) = 0.0   4/8 Ent( ) = 0.0

Entropy and Information Gain

$$Entropy(E) = \sum_{i \in C} -P_i log_2(P_i)$$

$$Gain(E, A) = Entropy(E) - \sum_{v \in Values(A)} \frac{|E_v|}{|E|} Entropy(E_v)$$

- E: set of examples: $\langle x, f(x) \rangle$ pairs
- C: set of output classes
- A: a single attribute
- E_v: set of examples where attribute $A = v$.
- $|S|$: cardinality of set S.

$$1.0 - \left(\frac{4}{8} \cdot 0 + \frac{4}{8} \cdot 0 \right) = 1.0$$

0

So you gain a much, much information after testing is this attribute so that's 1 one trivial case. Yeah.

You are screen sharing Stop Share

Entropy and Information Gain

$$Entropy(E) = \sum_{i \in C} -P_i log_2(P_i)$$

$$Gain(E, A) = Entropy(E) - \sum_{v \in Values(A)} \frac{|E_v|}{|E|} Entropy(E_v)$$

- E: set of examples: $\langle x, f(x) \rangle$ pairs
- C: set of output classes
- A: a single attribute
- E_v: set of examples where attribute $A = v$.
- $|S|$: cardinality of set S.

+	4
−	4

Ent( ) = 1.0

$|E| = 8$

Attr

Cold Hot

$|E_{hot}| = 4$

$|E_{cold}| = 4$

$\frac{4}{8}$ Ent( ) = 0.0 $\quad \frac{4}{8}$ Ent( ) = 0.0

$1.0 - \left(\frac{4}{8} \cdot 0 + \frac{4}{8} \cdot 0\right)$

0

Ent 1.0

+	4
−	4

Attr

Strng Weak

+	1
−	1

+	3
−	3

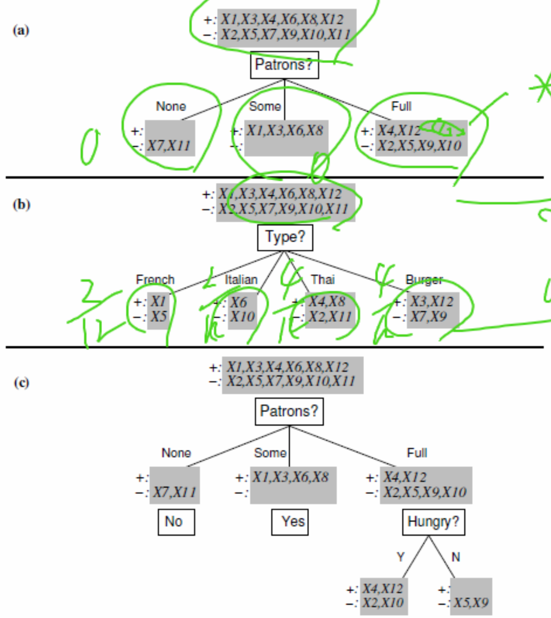
$\frac{2}{8} \times$ Ent = 1.0 $\quad \frac{6}{8} \times$ Ent = 1.0

$1.0 - \left(\frac{2}{8} \times 1.0 + \frac{6}{8} \times 1.0\right) = 0$

$\frac{8}{8} \times 1.0 = 1.0$

There was 50 50 before, and 50 50 after there's no information gained. but of course you

Finding a Concise Decision Tree (cont'd)



x 6/12

non-zero

zero gain

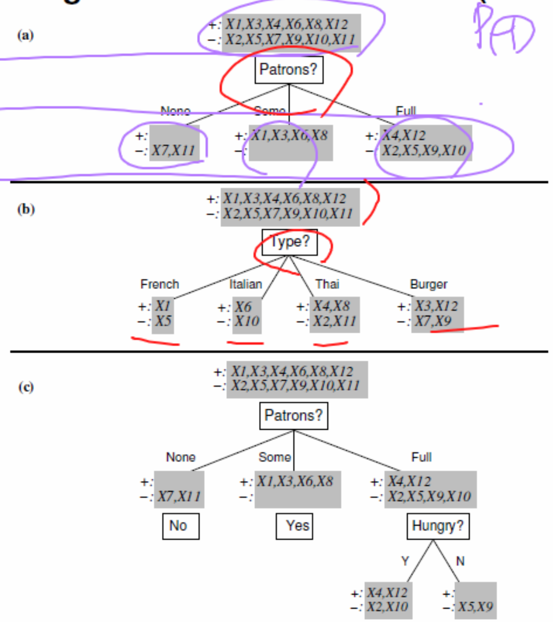
So 0 gain

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save

Who can see what you share here? Recording On

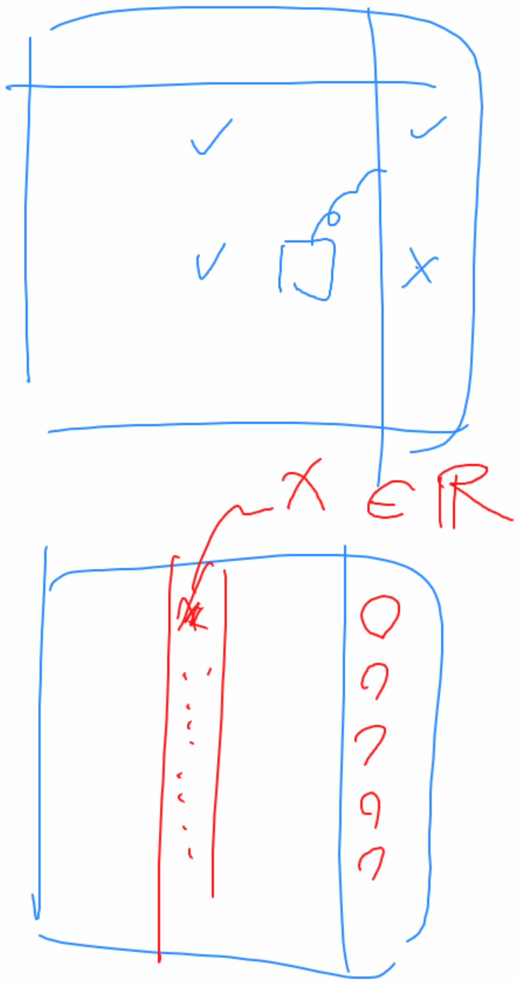
Finding a Concise Decision Tree (cont'd)

P(+) P(-) → Entropy



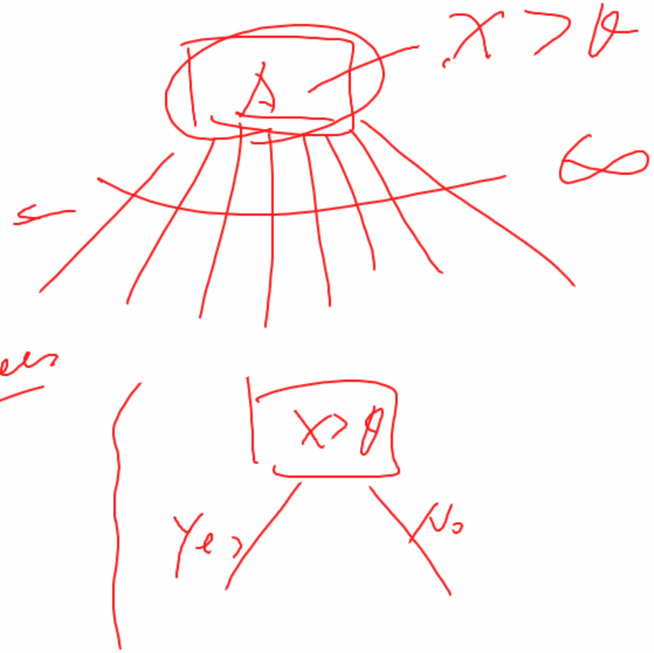
So if you check the different attribute, even with the same data set, you get a difference that. And so

You are screen sharing Stop Share



Issues in Decision Tree Learning

- Noise and overfitting
- Missing attribute values from examples
- Multi-valued attributes with large number of possible values
- Continuous-valued attributes.



Regression Trees

Key Points

Decision tree learning:

- What is the embodied principle (or "inductive bias")?
- How to choose the best attribute? Given a set of examples, choose the best attribute to test first.
- What are the issues? noise, overfitting, etc.

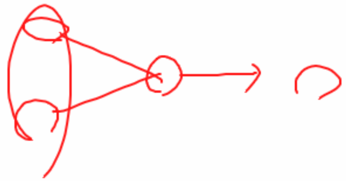
want shorter trees

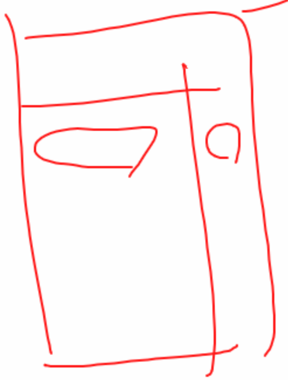
use of infogain

Jane in full sorry gain, is what implements this inductive bias

1950

Perceptron





Neural Networks

Neural networks is one particular form of learning from data.

- simple processing elements: called "units", or "neurons"
- connective structure and associated connection weights
- learning: adaptation of connection weights

Neural networks mimic the human (or animal) nervous system.

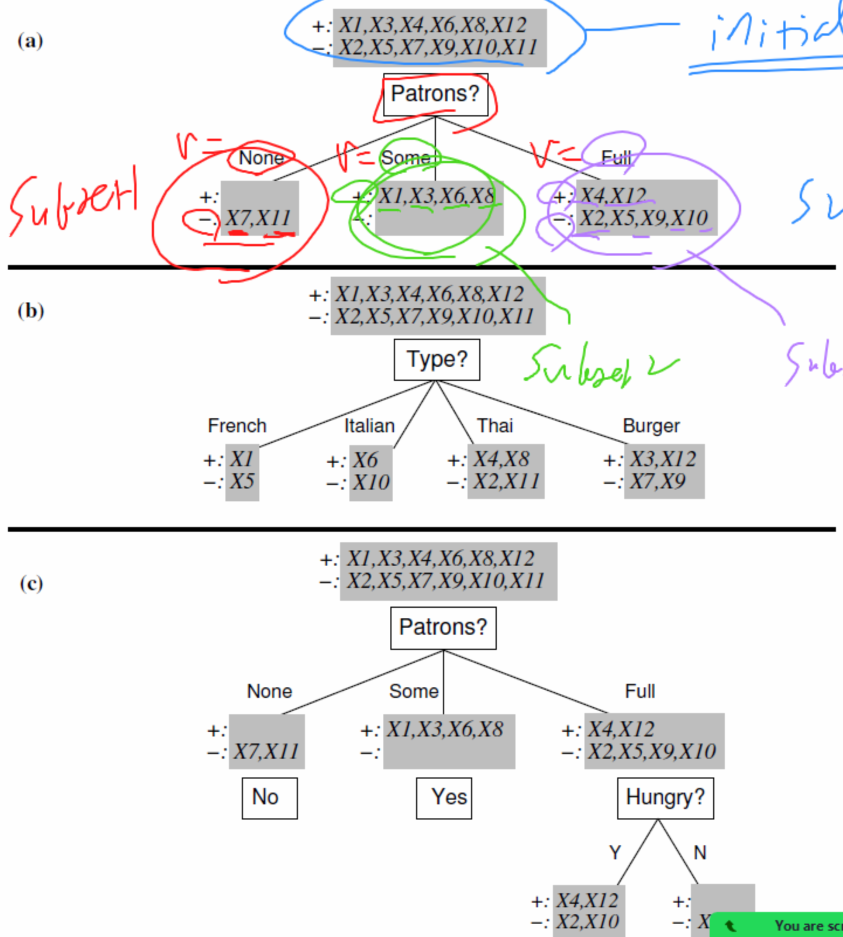
So you plug in the input and I give you the output that you

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save

Who can see what you share here? Recording On

#	Attr	Target
X1	Some	+
X2	Full	−
X3	Some	+
X4	Full	+
X5	Full	−
X6	Some	+
X7	None	−
X8	Some	+
X9	Full	−
X10	Full	−
X11	None	−
X12	Full	+

Finding a Concise Decision Tree (cont'd)



initial data set.

subsets based on attr. value.

You are screen sharing Stop Share

Let me create

Mute Stop Video Security Participants 8 Chat Polls New Share Pause Share Annotate Remote Control Apps More

1950

Perceptron

w_1

w_2

Neural Networks

Neural networks is one particular form of learning from data.

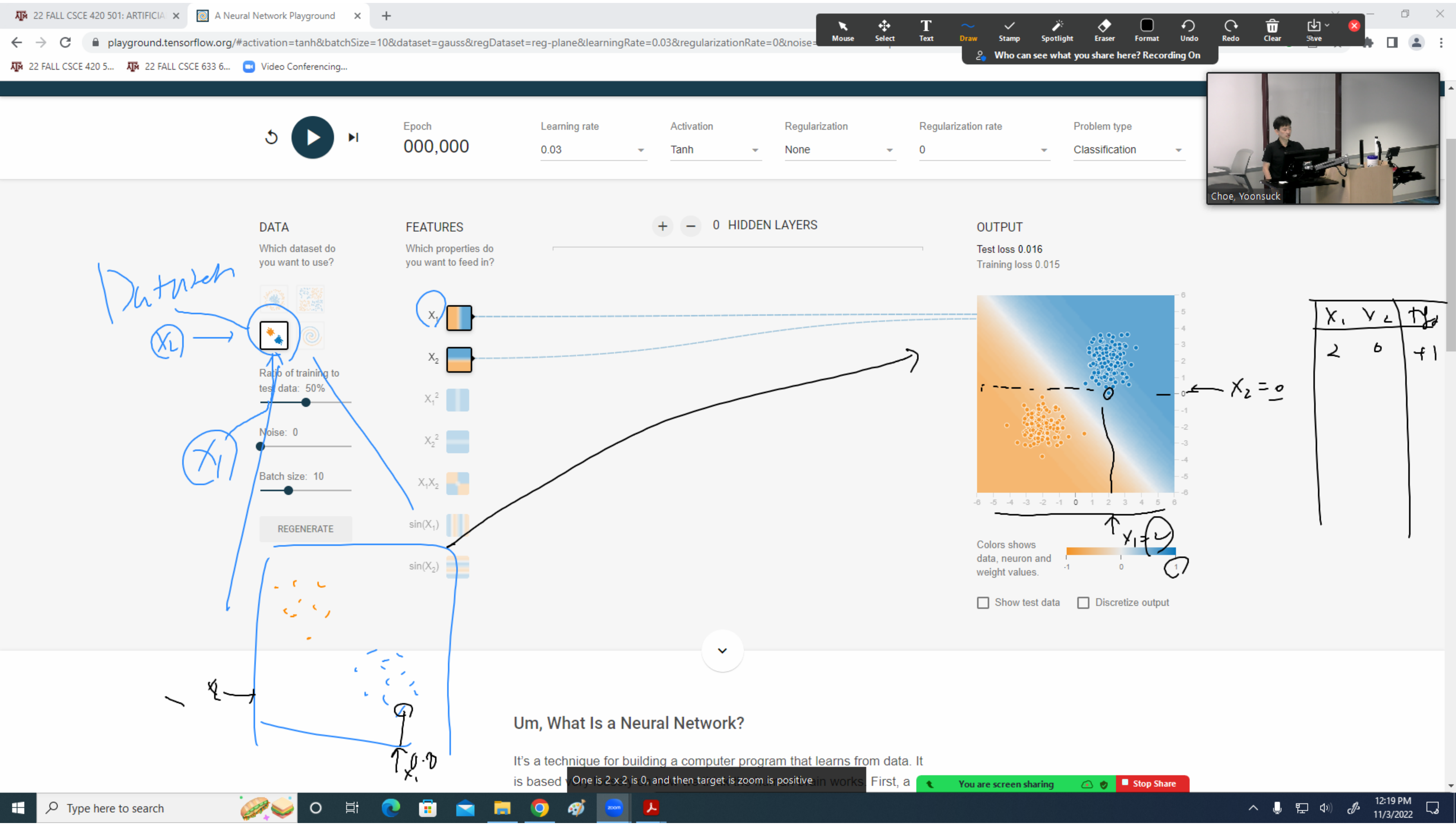
- simple processing elements: called “units”, or “neurons”
- connective structure and associated connection weights
- learning: adaptation of connection weights

Neural networks mimic the human (or animal) nervous system.

1985

2010-2012

DL


22 FALL CSCE 420 501: ARTIFICIA
A Neural Network Playground
playground.tensorflow.org/#activation=tanh&batchSize=10&dataset=gauss&regDataset=reg-plane&learningRate=0.03&regularizationRate=0&noise=
22 FALL CSCE 420 5...
22 FALL CSCE 633 6...
Video Conferencing...
Mouse
Select
Text
Draw
Stamp
Spotlight
Eraser
Format
Undo
Redo
Clear
Save
Who can see what you share here? Recording On
Choe, Yoonsuck
Epoch
000,000
Learning rate
0.03
Activation
Tanh
Regularization
None
Regularization rate
0
Problem type
Classification
DATA
Which dataset do you want to use?
Ratio of training to test data: 50%
Noise: 0
Batch size: 10
REGENERATE
FEATURES
Which properties do you want to feed in?
0 HIDDEN LAYERS
OUTPUT
Test loss 0.016
Training loss 0.015
Colors shows data, neuron and weight values.
Show test data
Discretize output
Dataset
$X_2 = 0$
$X_1 = 2$
X_1 X_2 t
2 0 +1
Um, What Is a Neural Network?
It's a technique for building a computer program that learns from data. It is based
One is 2 x 2 is 0, and then target is zoom is positive
First, a
You are screen sharing
Stop Share
Type here to search
12:19 PM
11/3/2022

22 FALL CSCE 420 501: ARTIFICIA | A Neural Network Playground

playground.tensorflow.org/#activation=tanh&batchSize=10&dataset=gauss&regDataset=reg-plane&learningRate=0.03&regularizationRate=0&noise=

22 FALL CSCE 420 5... | 22 FALL CSCE 633 6... | Video Conferencing...

Mouse | Select | Text | Draw | Stamp | Spotlight | Eraser | Format | Undo | Redo | Clear | Save

Who can see what you share here? Recording On

Choe, Yoonsuck

Epoch	Learning rate	Activation	Regularization	Regularization rate	Problem type
000,000	0.03	Tanh	None	0	Classification

DATA

Which dataset do you want to use?

Dataset

X_2 →

X_1

Ratio of training to test data: 50%

Noise: 0

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

X_1

X_2

X_1^2

X_2^2

X_1X_2

$\sin(X_1)$

$\sin(X_2)$

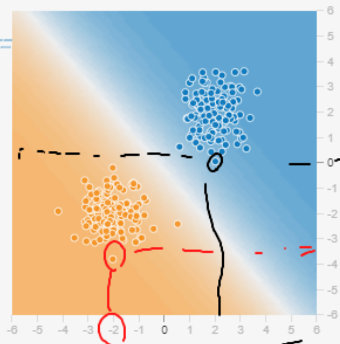
$(0,0)$

+ − 0 HIDDEN LAYERS

OUTPUT

Test loss 0.016

Training loss 0.015



$X_2 = 0$

$X_2 = -3.5$

$X_1 = 2$

$X_1 = -2$

Colors shows data, neuron and weight values.

Show test data

Discretize output

X_1	X_2	t
2	0	+1
−2	−3.5	−1

Um, What Is a Neural Network?

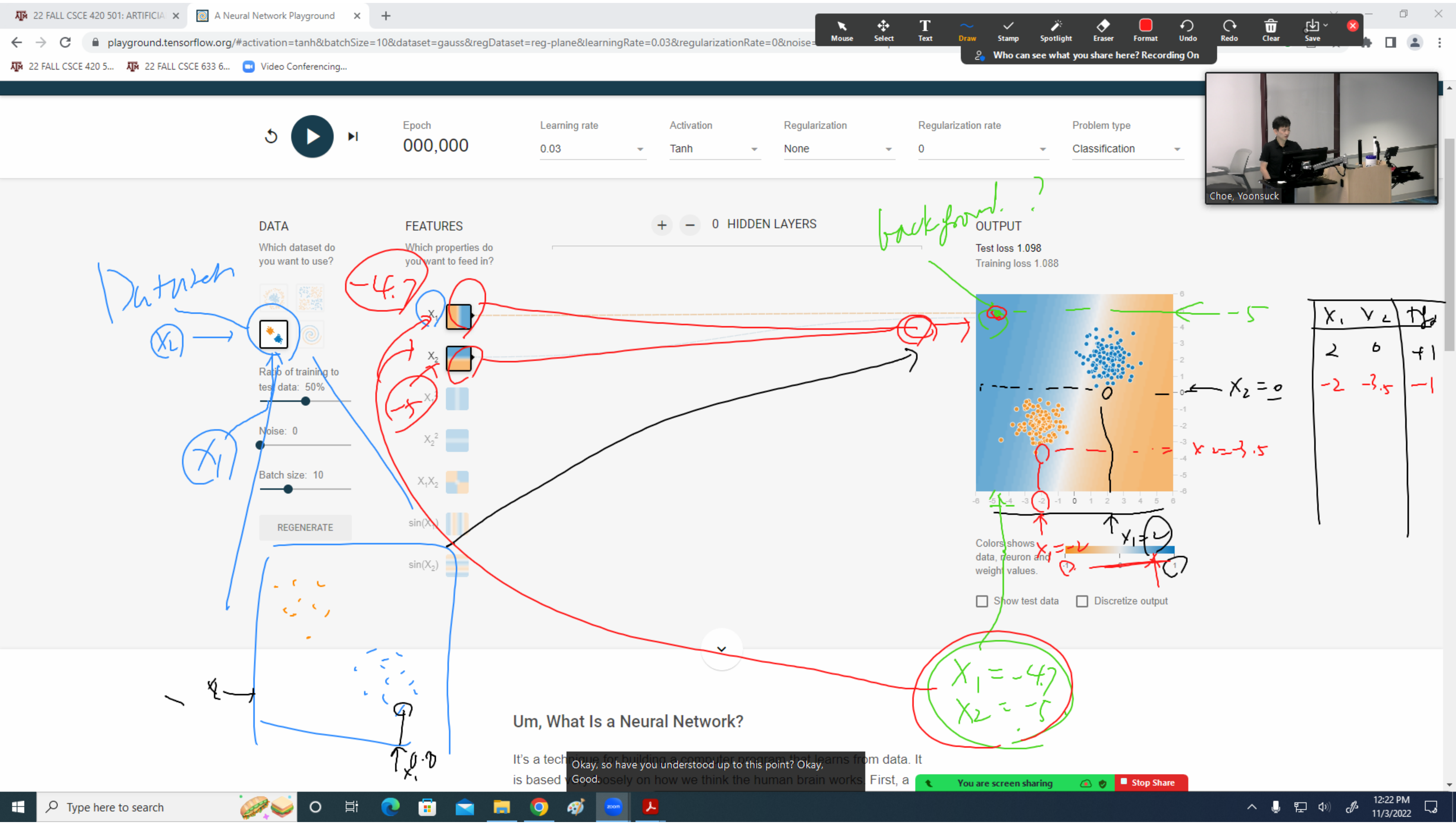
It's a technique for building a computer program that learns from data. It is based ... think the human brain works. First, a

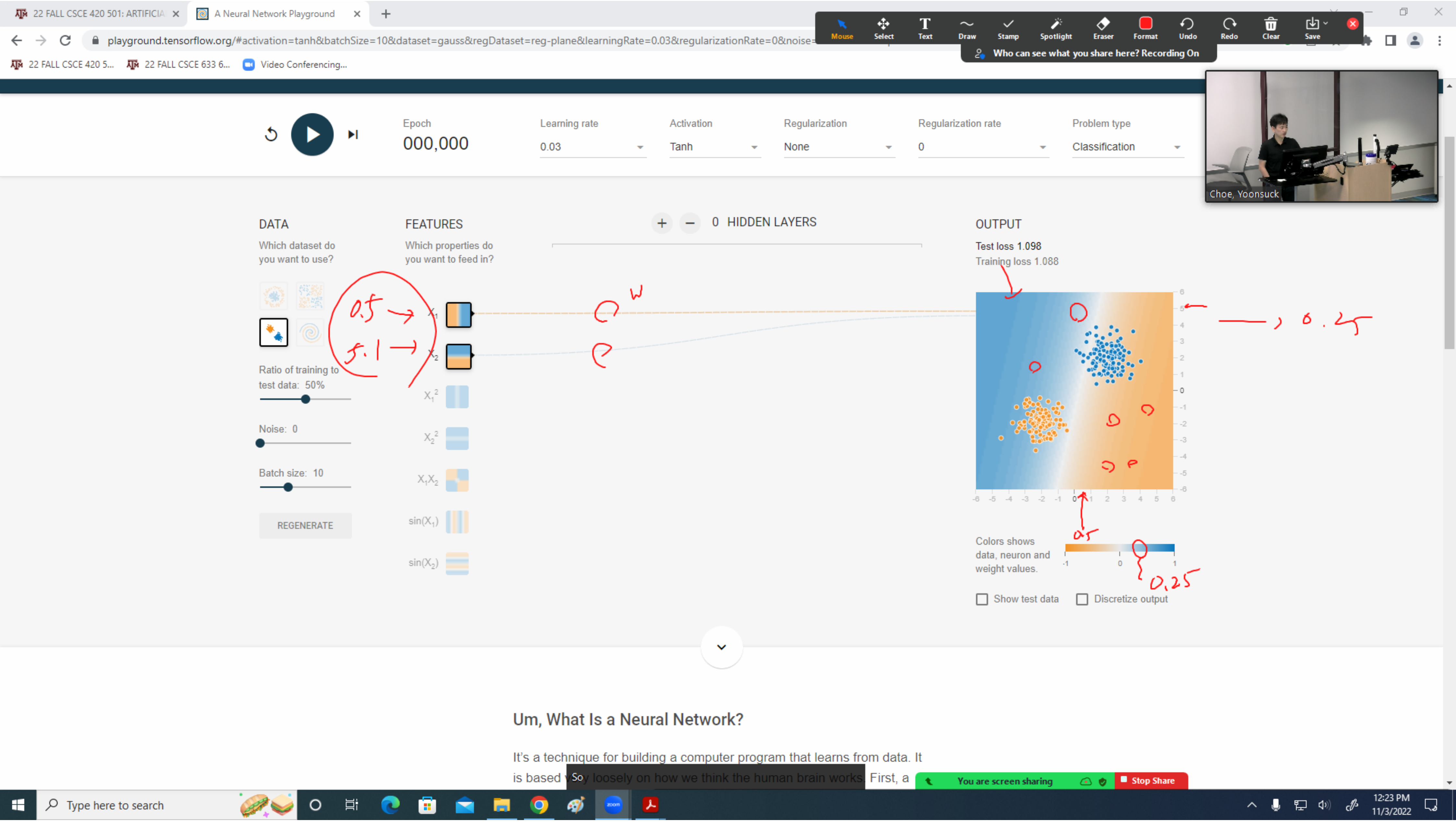
So that's the data set so let me

You are screen sharing | Stop Share

Type here to search

12:20 PM
11/3/2022


22 FALL CSCE 420 501: ARTIFICIA
A Neural Network Playground
playground.tensorflow.org/#activation=tanh&batchSize=10&dataset=gauss&regDataset=reg-plane&learningRate=0.03&regularizationRate=0&noise=
22 FALL CSCE 420 5...
22 FALL CSCE 633 6...
Video Conferencing...
Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On
Choe, Yoonsuck
Epoch
000,000
Learning rate
0.03
Activation
Tanh
Regularization
None
Regularization rate
0
Problem type
Classification
DATA
Which dataset do you want to use?
Ratio of training to test data: 50%
Noise: 0
Batch size: 10
REGENERATE
FEATURES
Which properties do you want to feed in?
0 HIDDEN LAYERS
OUTPUT
Test loss 1.098
Training loss 1.088
Colors shows data, neuron and weight values.
Show test data
Discretize output
Um, What Is a Neural Network?
It's a technique for building a computer program that learns from data. It is based very loosely on how we think the human brain works. First, a
Okay, so have you understood up to this point? Okay, Good.
You are screen sharing
Stop Share
Type here to search
12:22 PM
11/3/2022
Dataset
backward?
-4.7
-5
X₂ = 0
X₁ = -4.7
X₂ = -5
X₁ = 2
X₁ = -2
x₂ = -3.5
X₁ X₂ Tgt
2 6 +1
-2 -3.5 -1


22 FALL CSCE 420 501: ARTIFICIA
A Neural Network Playground
playground.tensorflow.org/#activation=tanh&batchSize=10&dataset=gauss&regDataset=reg-plane&learningRate=0.03&regularizationRate=0&noise=
Mouse
Select
Text
Draw
Stamp
Spotlight
Eraser
Format
Undo
Redo
Clear
Save
Who can see what you share here? Recording On
22 FALL CSCE 420 5...
22 FALL CSCE 633 6...
Video Conferencing...
Choe, Yoonsuck
Epoch
000,000
Learning rate
0.03
Activation
Tanh
Regularization
None
Regularization rate
0
Problem type
Classification
DATA
Which dataset do you want to use?
FEATURES
Which properties do you want to feed in?
0 HIDDEN LAYERS
OUTPUT
Test loss 1.098
Training loss 1.088
0.5
5.1
W
0.25
0.5
Ratio of training to test data: 50%
Noise: 0
Batch size: 10
REGENERATE
Colors shows data, neuron and weight values.
Show test data
Discretize output
Um, What Is a Neural Network?
It's a technique for building a computer program that learns from data. It is based loosely on how we think the human brain works. First, a
You are screen sharing
Stop Share
Type here to search
12:23 PM
11/3/2022

22 FALL CSCE 420 501: ARTIFICIA | A Neural Network Playground

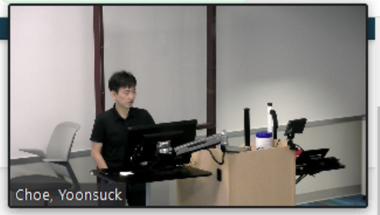
playground.tensorflow.org/#activation=tanh&batchSize=10&dataset=gauss&regDataset=reg-plane&learningRate=0.03&regularizationRate=0&noise=

22 FALL CSCE 420 5... | 22 FALL CSCE 633 6... | Video Conferencing...

Mouse | Select | Text | Draw | Stamp | Spotlight | Eraser | Format | Undo | Redo | Clear | Save

Who can see what you share here? Recording On

Saved as PNG | Show in Folder



Epoch	Learning rate	Activation	Regularization	Regularization rate	Problem type
000,000	0.03	Tanh	None	0	Classification

DATA

Which dataset do you want to use?

Ratio of training to test data: 50%

Noise: 0

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

0.5 →

5.1 →

X_1

X_2

X_1^2

X_2^2

X_1X_2

$\sin(X_1)$

$\sin(X_2)$

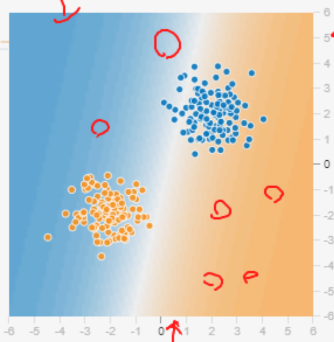
+ − 0 HIDDEN LAYERS

w

OUTPUT

Test loss 1.098

Training loss 1.088



→ 0.25

Colors shows data, neuron and weight values.

-1 0 1

0.25

Show test data | Discretize output

Um, What Is a Neural Network?

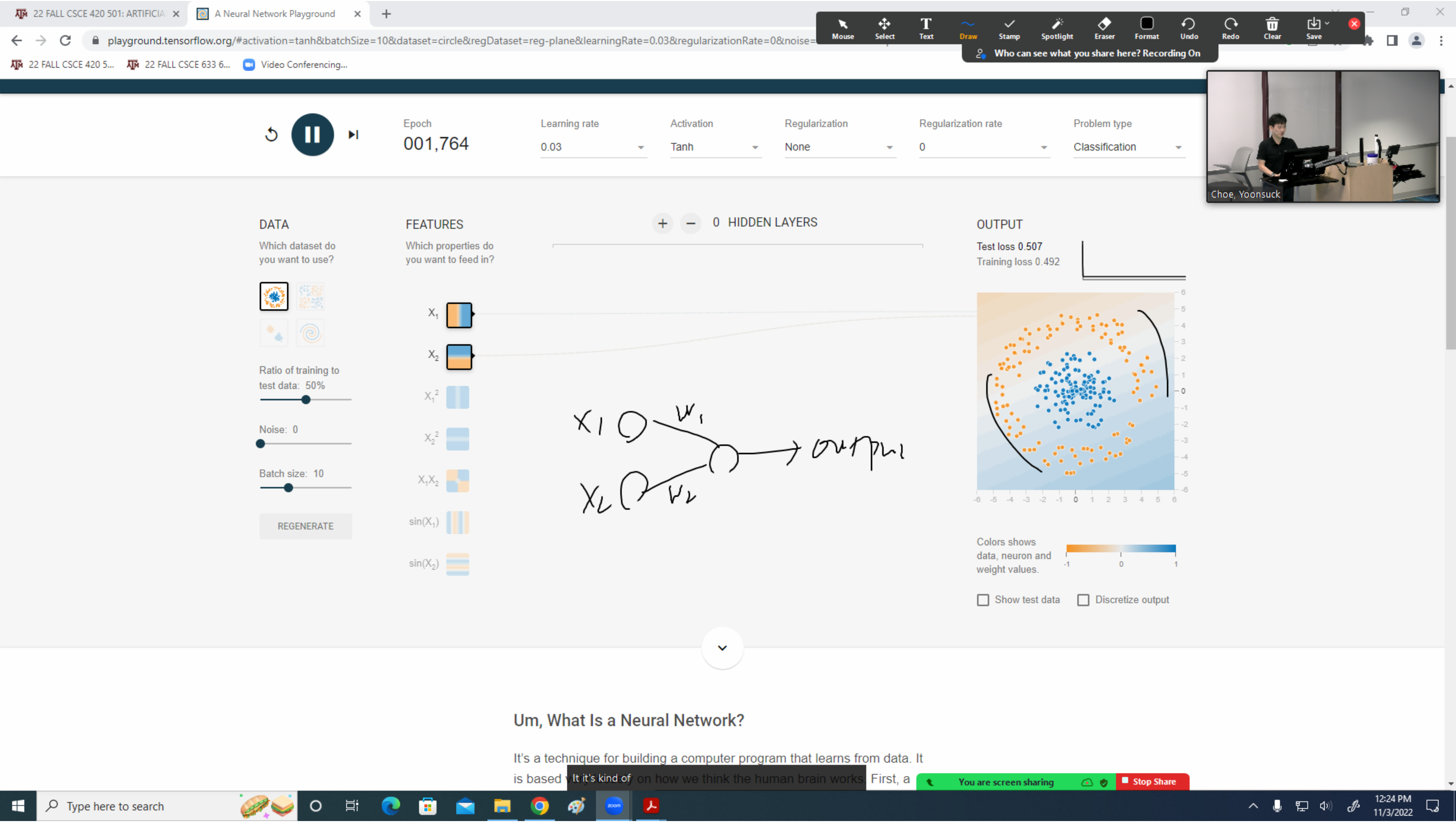
It's a technique for building a computer program that learns from data. It is based very loosely on how we think the human brain works. First, a

So loosely on how we think the human brain works

You are screen sharing | Stop Share

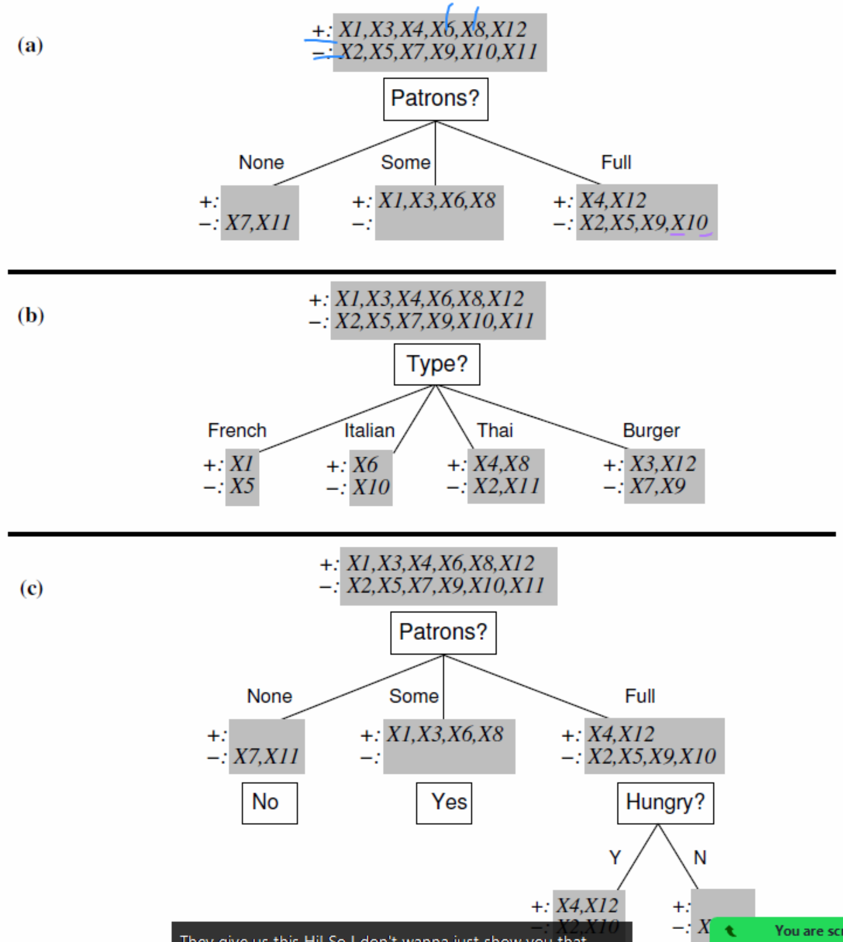
Type here to search

12:23 PM 11/3/2022


22 FALL CSCE 420 501: ARTIFICIA
A Neural Network Playground
playground.tensorflow.org/#activation=tanh&batchSize=10&dataset=circle&regDataset=reg-plane&learningRate=0.03&regularizationRate=0&noise=
Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On
22 FALL CSCE 420 5...
22 FALL CSCE 633 6...
Video Conferencing...
Choe, Yoonsuck
Epoch
001,764
Learning rate
0.03
Activation
Tanh
Regularization
None
Regularization rate
0
Problem type
Classification
DATA
Which dataset do you want to use?
Ratio of training to test data: 50%
Noise: 0
Batch size: 10
REGENERATE
FEATURES
Which properties do you want to feed in?
0 HIDDEN LAYERS
OUTPUT
Test loss 0.507
Training loss 0.492
Colors shows data, neuron and weight values.
Show test data
Discretize output
X1
w1
X2
w2
output
Um, What Is a Neural Network?
It's a technique for building a computer program that learns from data. It is based
It it's kind of
on how we think the human brain works. First, a
You are screen sharing
Stop Share
Type here to search
12:24 PM
11/3/2022

Finding a Concise Decision Tree (cont'd)

#	Attr	Target
X1	Some	+
X2	Full	−
X3	Some	+
X4	Full	+
X5	Full	−
X6	Some	+
X7	None	−
X8	Some	+
X9	Full	−
X10	Full	−
X11	None	−
X12	Full	+



Attr

Target

Constructing Decision Trees from Examples

Example	Attributes										Goal
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	*Yes*	*No*	*No*	*Yes*	*Some*	*\$\$\$*	*No*	*Yes*	*French*	*0–10*	*Yes*
X_2	*Yes*	*No*	*No*	*Yes*	*Full*	*\$*	*No*	*No*	*Thai*	*30–60*	*No*
X_3	*No*	*Yes*	*No*	*No*	*Some*	*\$*	*No*	*No*	*Burger*	*0–10*	*Yes*
X_4	*Yes*	*No*	*Yes*	*Yes*	*Full*	*\$*	*No*	*No*	*Thai*	*10–30*	*Yes*
X_5	*Yes*	*No*	*Yes*	*No*	*Full*	*\$\$\$*	*No*	*Yes*	*French*	*>60*	*No*
X_6	*No*	*Yes*	*No*	*Yes*	*Some*	*\$\$*	*Yes*	*Yes*	*Italian*	*0–10*	*Yes*
X_7	*No*	*Yes*	*No*	*No*	*None*	*\$*	*Yes*	*No*	*Burger*	*0–10*	*No*
X_8	*No*	*No*	*No*	*Yes*	*Some*	*\$\$*	*Yes*	*Yes*	*Thai*	*0–10*	*Yes*
X_9	*No*	*Yes*	*Yes*	*No*	*Full*	*\$*	*Yes*	*No*	*Burger*	*>60*	*No*
X_{10}	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*\$\$\$*	*No*	*Yes*	*Italian*	*10–30*	*No*
X_{11}	*No*	*No*	*No*	*No*	*None*	*\$*	*No*	*No*	*Thai*	*0–10*	*No*
X_{12}	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*\$*	*No*	*No*	*Burger*	*30–60*	*Yes*

#		
X1	Some	+
X2	Full	−
X3	Some	+
X4	Full	+
X5	Full	−
X6	Some	+
X7	None	−
X8	Some	+
X9	Full	−
X10	Full	−
X11	None	−
X12	Full	+

- Given a set of examples (**training set**), both **positive** and **negative**, the task is to construct a decision tree that describes a concise decision path.
- Using the resulting decision tree, we want to **classify** new instances of examples (either as **yes** or **no**).

Attr Target

#	Attr	Target
X1	Some	+
X2	Full	−
X3	Some	+
X4	Full	+
X5	Full	−
X6	Some	+
X7	None	−
X8	Some	+
X9	Full	−
X10	Full	−
X11	None	−
X12	Full	+

Constructing Decision Trees from Examples

Example	Attributes										Goal
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	*Yes*	*No*	*No*	*Yes*	*Some*	*\$\$\$*	*No*	*Yes*	*French*	*0–10*	*Yes*
X_2	*Yes*	*No*	*No*	*Yes*	*Full*	*\$*	*No*	*No*	*Thai*	*30–60*	*No*
X_3	*No*	*Yes*	*No*	*No*	*Some*	*\$*	*No*	*No*	*Burger*	*0–10*	*Yes*
X_4	*Yes*	*No*	*Yes*	*Yes*	*Full*	*\$*	*No*	*No*	*Thai*	*10–30*	*Yes*
X_5	*Yes*	*No*	*Yes*	*No*	*Full*	*\$\$\$*	*No*	*Yes*	*French*	*>60*	*No*
X_6	*No*	*Yes*	*No*	*Yes*	*Some*	*\$\$*	*Yes*	*Yes*	*Italian*	*0–10*	*Yes*
X_7	*No*	*Yes*	*No*	*No*	*None*	*\$*	*Yes*	*No*	*Burger*	*0–10*	*No*
X_8	*No*	*No*	*No*	*Yes*	*Some*	*\$\$*	*Yes*	*Yes*	*Thai*	*0–10*	*Yes*
X_9	*No*	*Yes*	*Yes*	*No*	*Full*	*\$*	*Yes*	*No*	*Burger*	*>60*	*No*
X_{10}	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*\$\$\$*	*No*	*Yes*	*Italian*	*10–30*	*No*
X_{11}	*No*	*No*	*No*	*No*	*None*	*\$*	*No*	*No*	*Thai*	*0–10*	*No*
X_{12}	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*\$*	*No*	*No*	*Burger*	*30–60*	*Yes*

- Given a set of examples (**training set**), both **positive** and **negative**, the task is to construct a decision tree that describes a concise decision path.
- Using the resulting decision tree, we want to **classify** new instances of examples (either as **yes** or **no**).

Attr Target

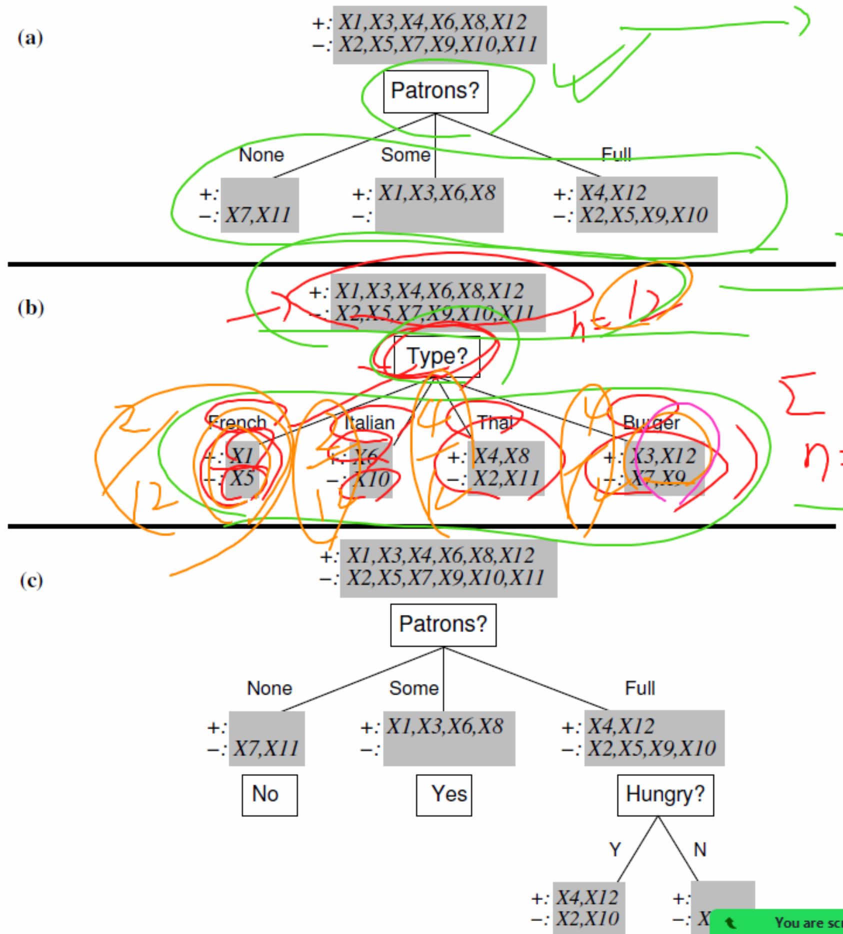
Constructing Decision Trees from Examples

Example	Attributes										Goal
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0–10*	*Yes*
X_2	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30–60*	*No*
X_3	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0–10*	*Yes*
X_4	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*10–30*	*Yes*
X_5	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	*No*
X_6	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0–10*	*Yes*
X_7	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0–10*	*No*
X_8	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0–10*	*Yes*
X_9	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	*No*
X_{10}	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10–30*	*No*
X_{11}	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0–10*	*No*
X_{12}	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30–60*	*Yes*

- Given a set of examples (**training set**), both **positive** and **negative**, the task is to construct a decision tree that describes a concise decision path.
- Using the resulting decision tree, we want to **classify** new instances of examples (either as **yes** or **no**).

Entropy

Finding a Concise Decision Tree (cont'd)



Uncertainty

huge drop

uncertainty

Σ n=h

info gained = 0.0

Finding a Concise Decision Tree (cont'd)