acm.ai

Logistic Regression and Intro to Neural Networks

Advanced Track Workshop #1

Anonymous Feedback: tinyurl.com/w21-atrack1-fb

Attendance Code: **icehockey**

Discord: bit.ly/ACMdiscord

Officers Helping with Advanced Track

Sudhanshu Agrawal

(he/him)

Jenson Choi

(he/him)

Adithya Nair

(he/him)

Slides originally developed by Ben, Justin, Paymon, Vaishnavi, and previous AI officers

Our Mission

To build and develop a community of students interested in AI at UCLA and Beyond.

Our Values

- Technical Proficiency and Awareness in AI
- Creating a Positive Impact on Society
- Diversity and Inclusion

Our Initiatives

Workshops

Projects

Events

Outreach

Projects: CassavaNet

What?

- Build an automatic system for detecting what disease a Cassava leaf has

Why?

- Cassava leaf is an important food source in Africa--helps to determine the viability of a crop

How to get involved

- Take Advanced track and Apply!
- Will be taking people during Spring Quarter and next year

Our Workshops

- **Beginner Track - *What is ML?***
  - Basics of machine learning
  - implement linear and logistic regression

- **Advanced Track - *Deep Learning***
  - Concepts like deep neural networks, CNNs, RNNs
  - Basic knowledge of ML concepts expected

- **Advanced++ Track - *More Deep Learning Concepts***
  - Transformers, Autoencoders, Quantum ML
  - Some knowledge of ML concepts expected

1. Overview and Logistics

Advanced Track Tentative Schedule

- Week 1: Logistic Regression + Feedforward Networks Intro
- Week 2: Logistic Regression Review + Feedword/Backpropagation
- Week 3: Convolutional Neural Networks
- Week 4: Convolutional Neural Networks 2
- Week 5: Recurrent Neural Networks (LSTMs)
- Week 6: Introduction to PyTorch
- Week 7: Guided Project
- Week 8: Guided Project

Thursdays 7:00 - 9:00 PM PDT

https://ucla.zoom.us/j/97882744284

Expected Prior Knowledge

- Comfortable with coding
- Basic machine learning knowledge
- Basic knowledge of working with data, i.e. Numpy arrays

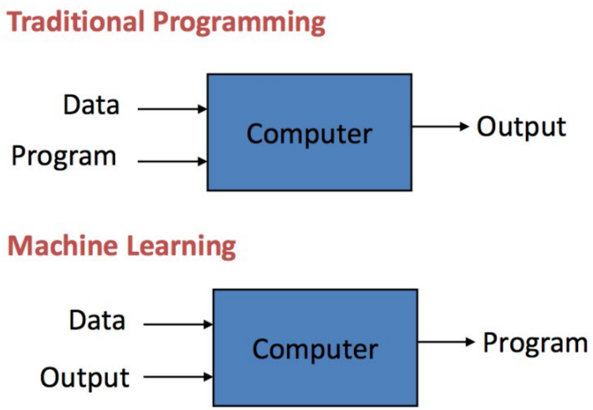
Today's Contents

- Review of machine learning
- Linear + Logistic Regression
- Introduction to neural networks

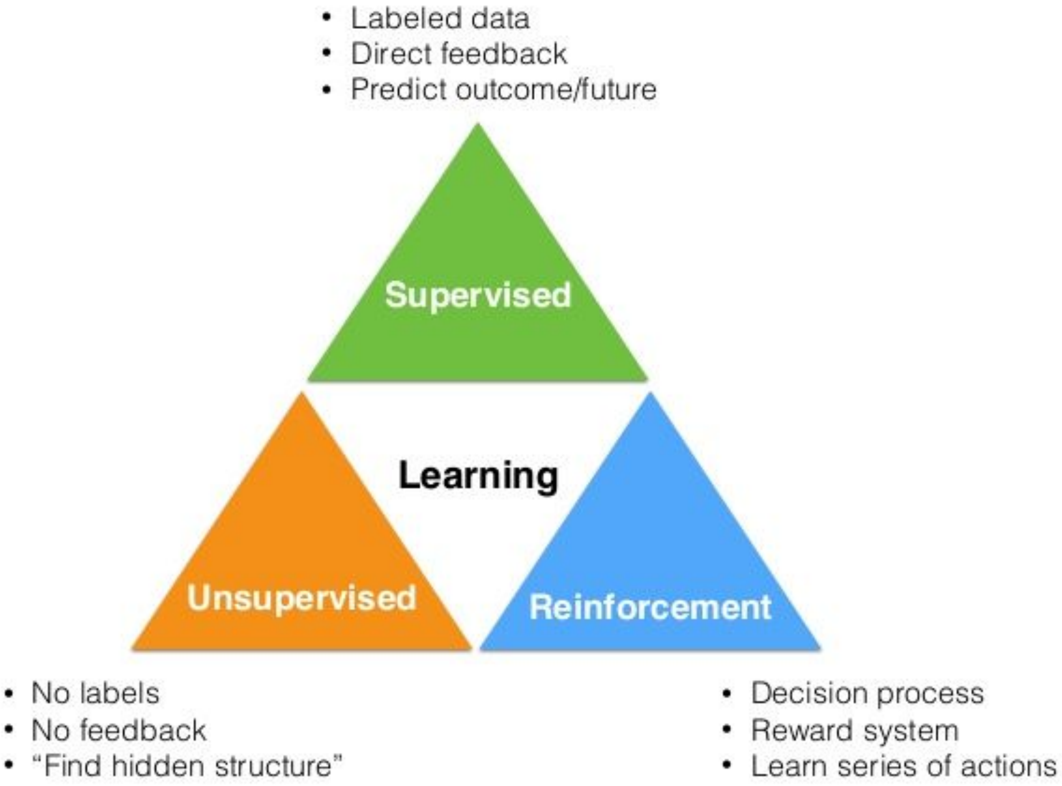
Questions?

2. Review of Linear/Logistic Regression

Machine Learning Refresher

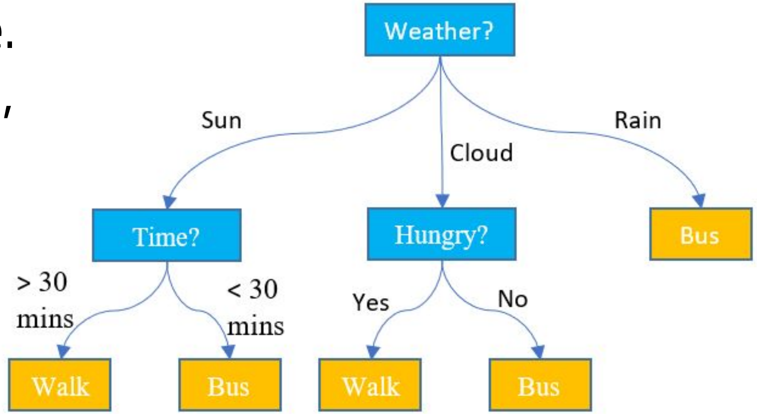


- Machine Learning is using data to teach computers how to answer our questions


- Labeled data
- Direct feedback
- Predict outcome/future
Supervised
Learning
Unsupervised
Reinforcement
- No labels
- No feedback
- "Find hidden structure"
- Decision process
- Reward system
- Learn series of actions

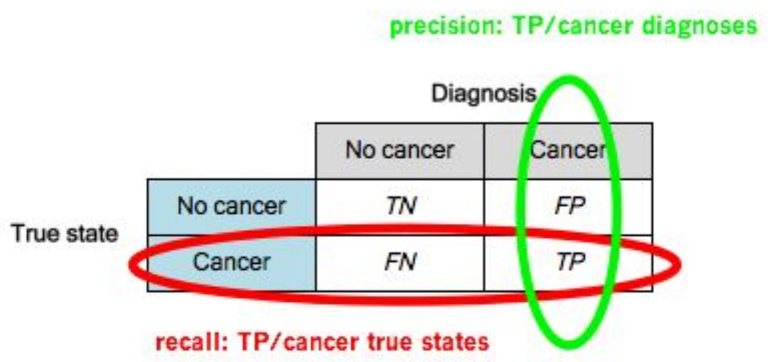
Key elements of Machine Learning

- Representation: Represent knowledge.
  - Eg: decision trees, neural networks



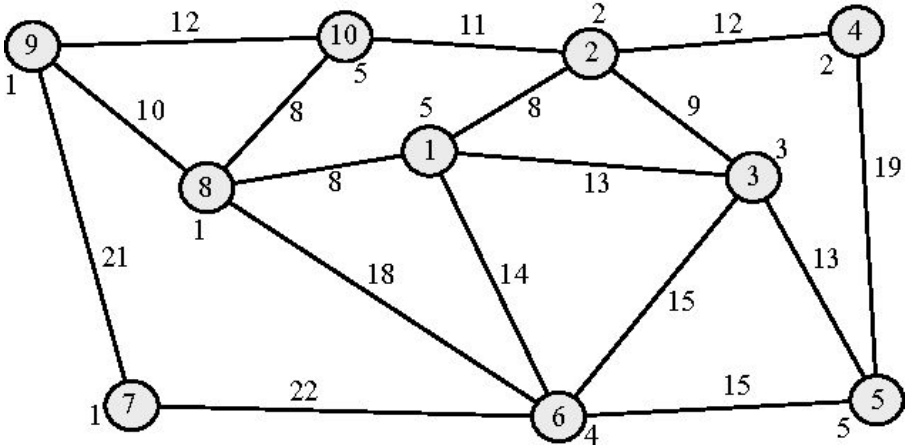
Key elements of Machine Learning

- **Evaluation:** Evaluate candidate programs (hypotheses).
  - Eg: Cost, margin, entropy, squared error



Key elements of Machine Learning

- **Optimization:** Candidate programs are generated (search process)
    - Eg: combinatorial optimization, gradient descent



Classification vs Regression

Identifying a digit

Classification Model

Prob. for #0 | Prob. for #1 | Prob. for #9

[0.05, 0.01, ..., 0.94]

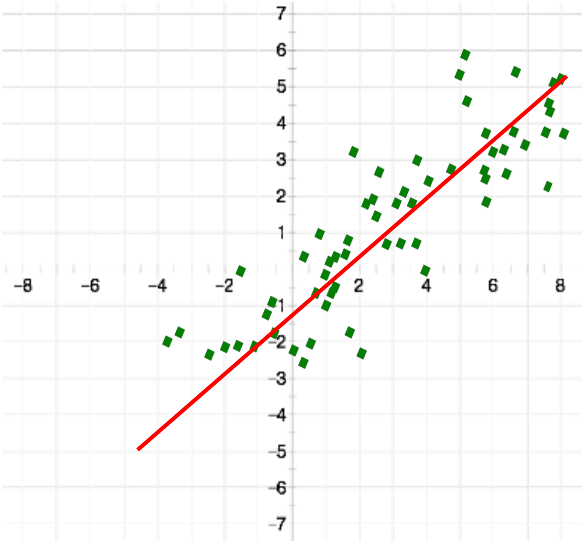
Predicting house prices

[4000 sqft]

Linear Regression Model

[$198000]

What is linear regression?



Best-fit line for plotted data

- Goal: to find the equation of a line that best fits our data
  - We want to be able to use this line to predict outputs from given inputs
- Notice that the outputs are continuous
  - Classification or regression?

$$\hat{y}(x_1, x_2 \ldots x_n) = b + w_1x_1 + w_2x \ldots + w_nx_n$$

An input **X** is an **n-dimensional vector** for the n features

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}$$

The weight **W** is also an n-dimensional vector.

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix}$$

The bias **b** is a real number.

$$b$$

acm.ai

Cost/Loss Function

$$L(\hat{y_1}, \hat{y_2}, ..., \hat{y_m}) = \frac{1}{m}\sum_{i=1}^{m}(\hat{y_i} - y_i)^2$$

- ŷ is the prediction, y is the ground truth, m is the number of samples
- Mean squared error

Gradient Descent

- Our goal is to minimize the loss
- We do this by adjusting the values for our weight matrices and bias terms
- Intuitively, we want to find out the impact of a particular weight on the overall error.

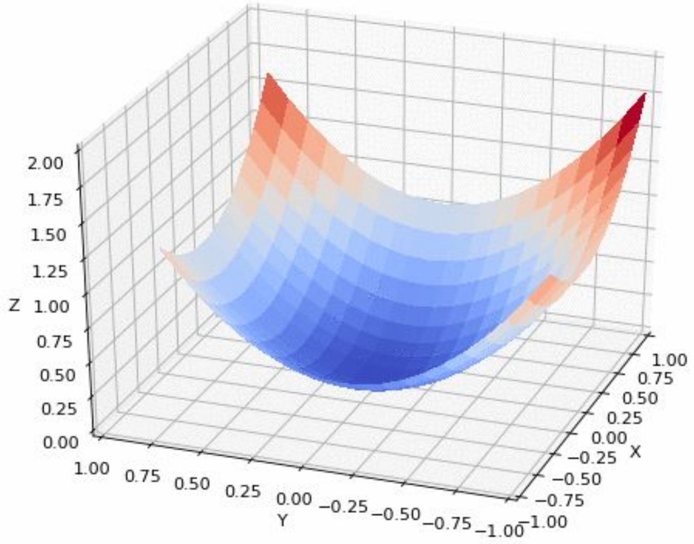
$$\frac{\delta L}{\delta w_j} = \frac{2}{m} \sum_{i=1}^{m} (\hat{y_i} - y_i) x_j$$

- Adjust the current weights:

$$w_j = w_j - \alpha \frac{\delta L}{\delta w_j}$$

Gradient Descent (cont.)

- Iterations of gradient descent continue until the cost function converges to a local minimum



acm.ai

Review

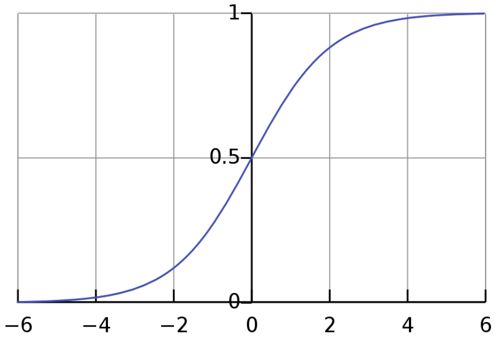
1. Which of these best describes machine learning (during training)?
    a. Data + Program = Output
    b. Data + Output = Program
2. The output of linear regression is:
    a. discrete
    b. continuous
    c. any real number
    d. only numbers between 0 and 1
3. Gradient descent _____ the cost function.
    a. maximizes
    b. minimizes

Classification Models

- Machine learning model used primarily for problem spaces where a selection from a group of categories/classes is needed.
- Example Problem Spaces
  - Cancer Detection (Binary 0/1)
  - Digit Classification (10 Classes)
  - Object Classification (Multiple Classes)

Logistic Regression

- Very similar to linear regression:
  - $Y = Wx + b$
- Logistic Regression Equation
  - $Y = \sigma(Wx + b)$



For binary classification: σ = Sigmoid Function

For Multi-class classification: σ = Softmax Function

Activation function

- Logistic Regression is just linear regression, with an activation function
- The activation function takes the continuous output of linear regression, and separates the result into classes in a non-linear fashion
- We can use the activation function to predict two classes (sigmoid function) or multiple classes (softmax function)

Linear Regression code

X: input (M samples x N features)

Y: target/label (M samples)

```python
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=5)

reg = LinearRegression().fit(X_train, Y_train)

r2 = reg.score(X_test, Y_test)

Y_pred = reg.predict(X_test)
```

acm.ai

Logistic Regression Code

X: input (M samples x N features)

Y: target/label (M samples)

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=5)

clf = LogisticRegression(random_state=0).fit(X_train, Y_train)

accuracy = clf.score(X_test, Y_test)

Y_pred = clf.predict(X_test)

Y_pmf = clf.predict_proba(X_test)
```

"

Questions?

acm.ai

3. Review

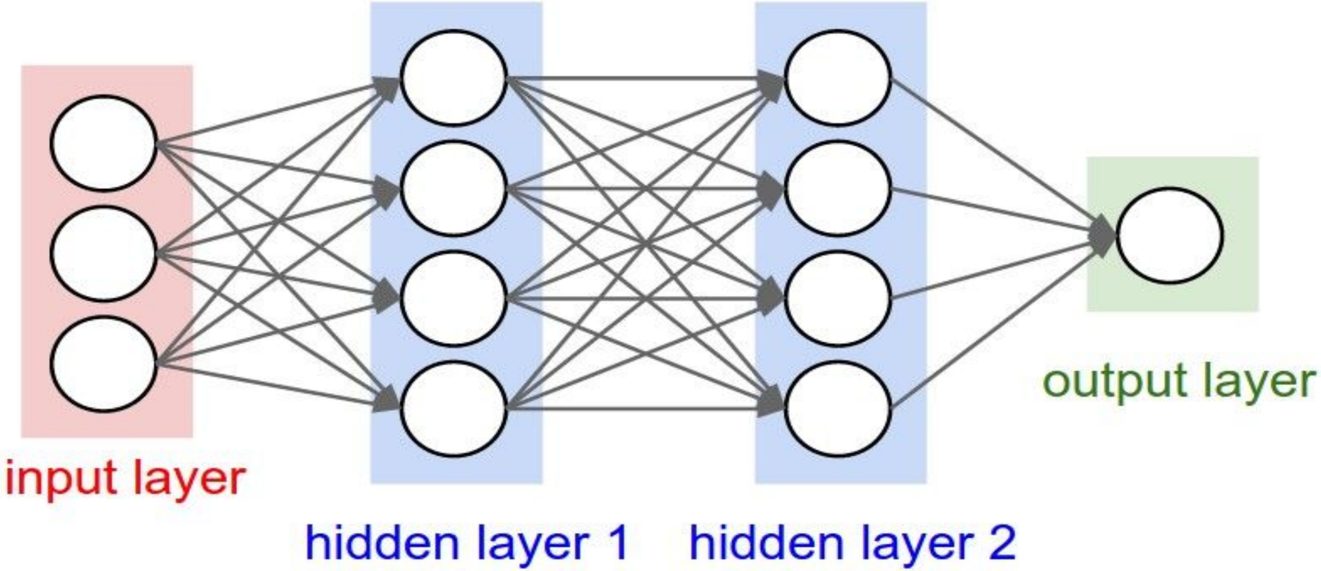
Review

1. Which of these describes a regression problem?
    a. Predict the amount of rainfall in inches based on the temperature
    b. Predict whether it will rain tomorrow based on temperature
2. The output of a classification model is:
    a. discrete
    b. continuous
    c. any real number
    d. only numbers between 0 and 1

4. Introduction to Neural Networks

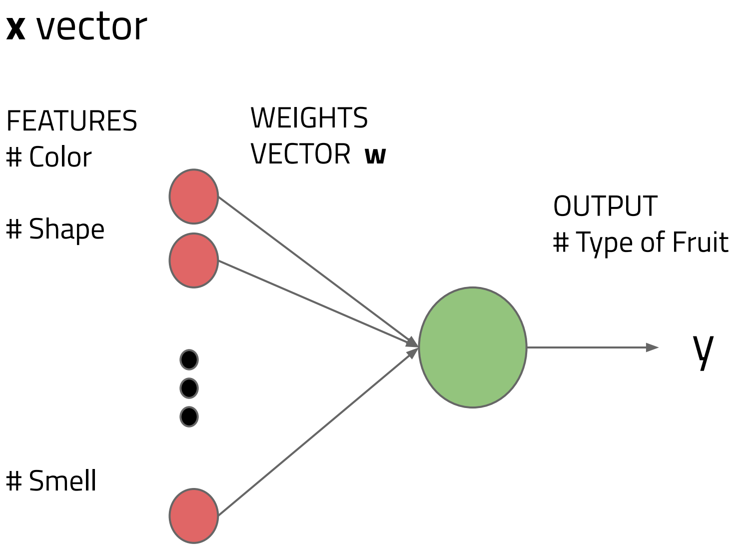
acm.ai

Neural Networks

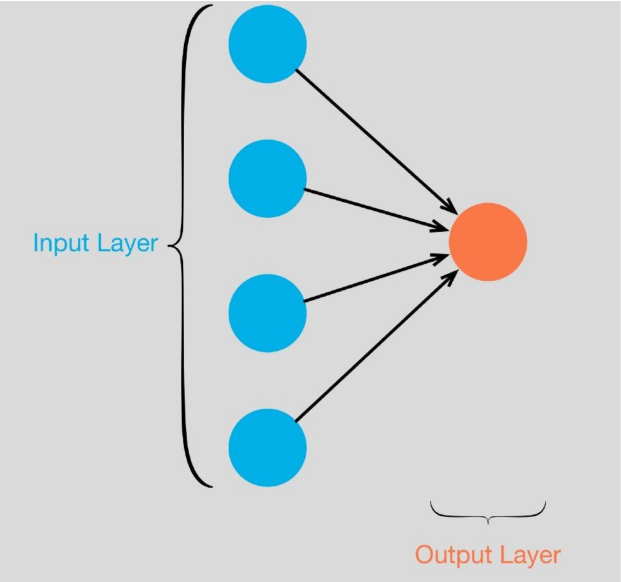


Artificial Neuron

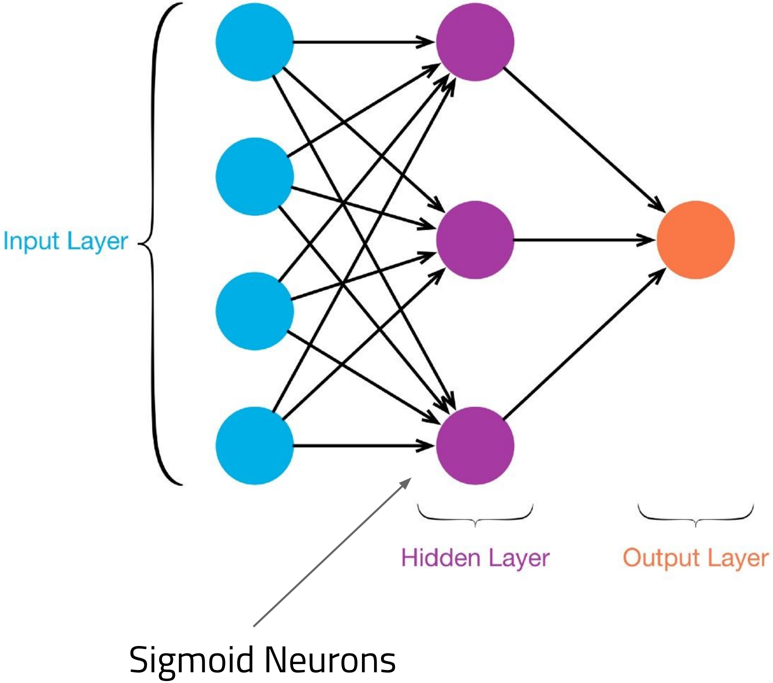
- Consists of:
  - Input **Vector** x
  - Weights **Vector** w
  - Activation Function g
- Example: Logistic Regression Equation
  - $y = g(w.x + b)$



Perceptron



Single layer NN



acm.ai

So why bother?

- Neural nets allow us to model more **complex** data
- Training on a neural net gives us a greater ability to model that information
- More complex architectures help us model relationships that linear classifiers cannot fully represent

Next Time: Neural Networks Pipeline

1. Define architecture
2. Pass inputs forward, get prediction
3. Compute loss
    - function of prediction and actual label
4. Perform gradient descent to get weight adjustments
5. Propagate adjustments backwards
    - backpropagation, chain rule for derivatives

acm.ai

Questions?

acm.ai

5. Upcoming Events

Advanced Track Meeting Next Week

- Next week, we'll be continuing with neural networks and flesh out their structure and implementation
- Backpropagation theory, the basis of almost all machine learning

Thank you all for coming!

Anonymous Feedback: https://tinyurl.com/w21-atrack1-fb

Facebook Group: https://www.facebook.com/groups/uclaacmai/

Github: https://github.com/uclaacmai/advanced-track-winter21