CS 546 – Advanced Topics in NLP

Dilek Hakkani-Tür

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

CONVAI

CDS Siebel School of Computing and Data Science

Topic for Today: Reminders

- Machine Learning (ML) for NLP
- ML Examples: NLP Tasks
- ML Basics
- Calculus Reminders
- Deep Learning
- Linear Regression – Case Study

What is Machine Learning?

- Machine learning (ML) is the field of artificial intelligence that develops **algorithms** and **statistical models** that **computers use to perform a specific task**
  - without using explicit instructions (i.e., coding the specific steps for performing the task),
  - relying on patterns and inference instead.

Example Task: Sentiment Detection for Product Reviews

Approach for solving tasks:

Input: “I love this camera!” → `program.py` → **+**

if input contains “love”, “like”, etc.
output = positive

“It makes too much noise.” → `program.py` → **-**

if input contains “too much”, “bad”, etc.
output = negative

“It’s a very bulky.” → `program.py` → **?**

Some tasks are complex, and we don’t know how to write a program to solve them
Example: speech recognition, summarization, …

Example Task: Sentiment Detection for Product Reviews

- Learning ≈ Looking for a Function

Input: "I love this camera!" → f → +

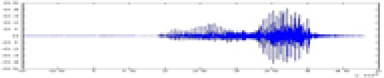
"It makes too much noise." → f → -

"It's a very bulky." → f → ?

Given a large amount of data, the machine learns what the function f should be.

Other Example Tasks

- Speech Recognition

$f($  $) =$ “the book”

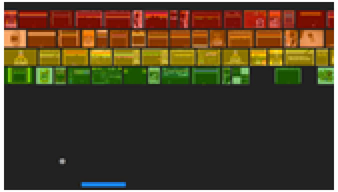
- Handwriting Recognition

$f($  $) =$ “2”

- Weather forecast

$f($ Thursday $) =$ “  Saturday”

- Playing video games

$f($  $) =$ “move left”

Topic for Today: Reminders

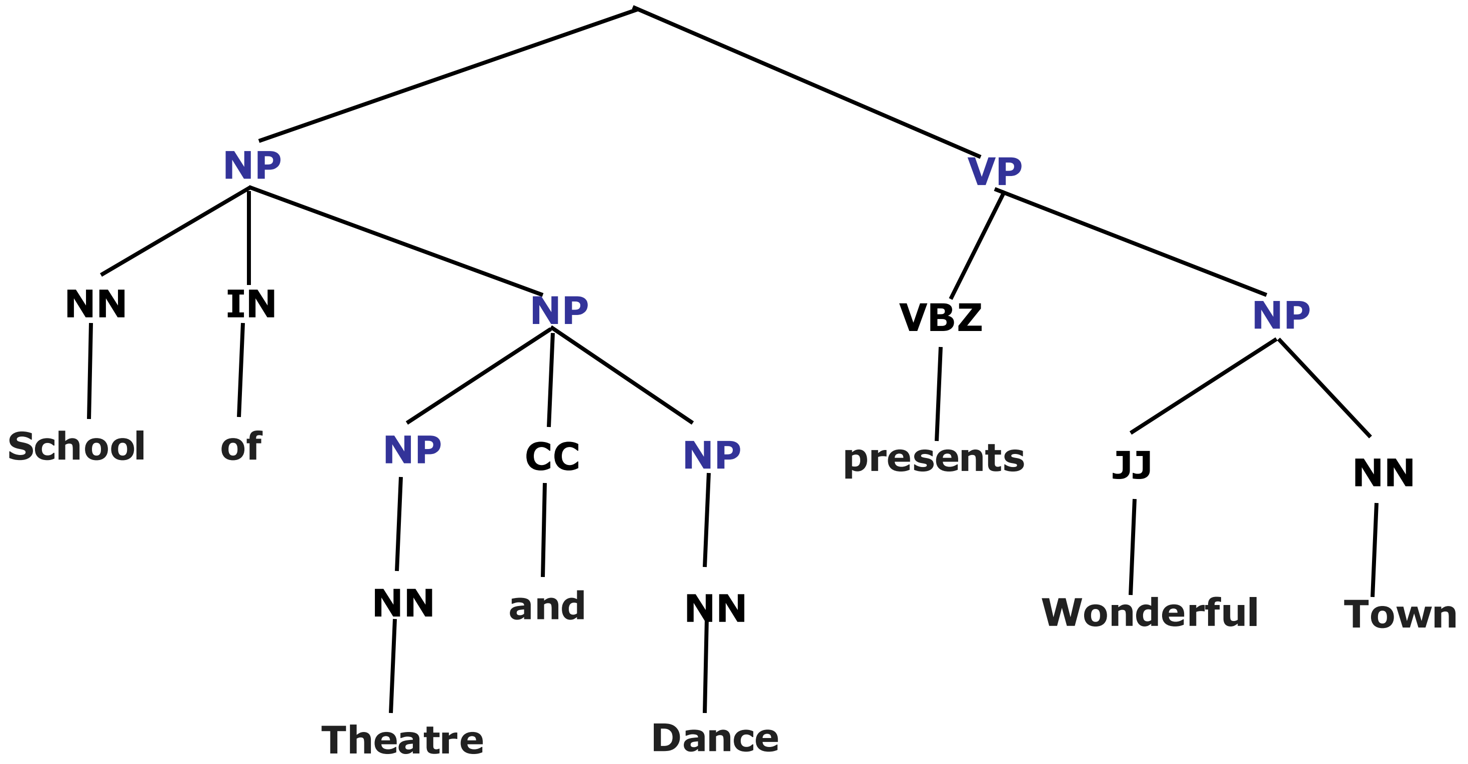
- Machine Learning (ML) for NLP
- ML Examples: NLP Tasks
- ML Basics
- Calculus Reminders
- Deep Learning
- Linear Regression – Case Study

Other Examples: Classical NLP Pipeline

- Sentence Segmentation
- Tokenization

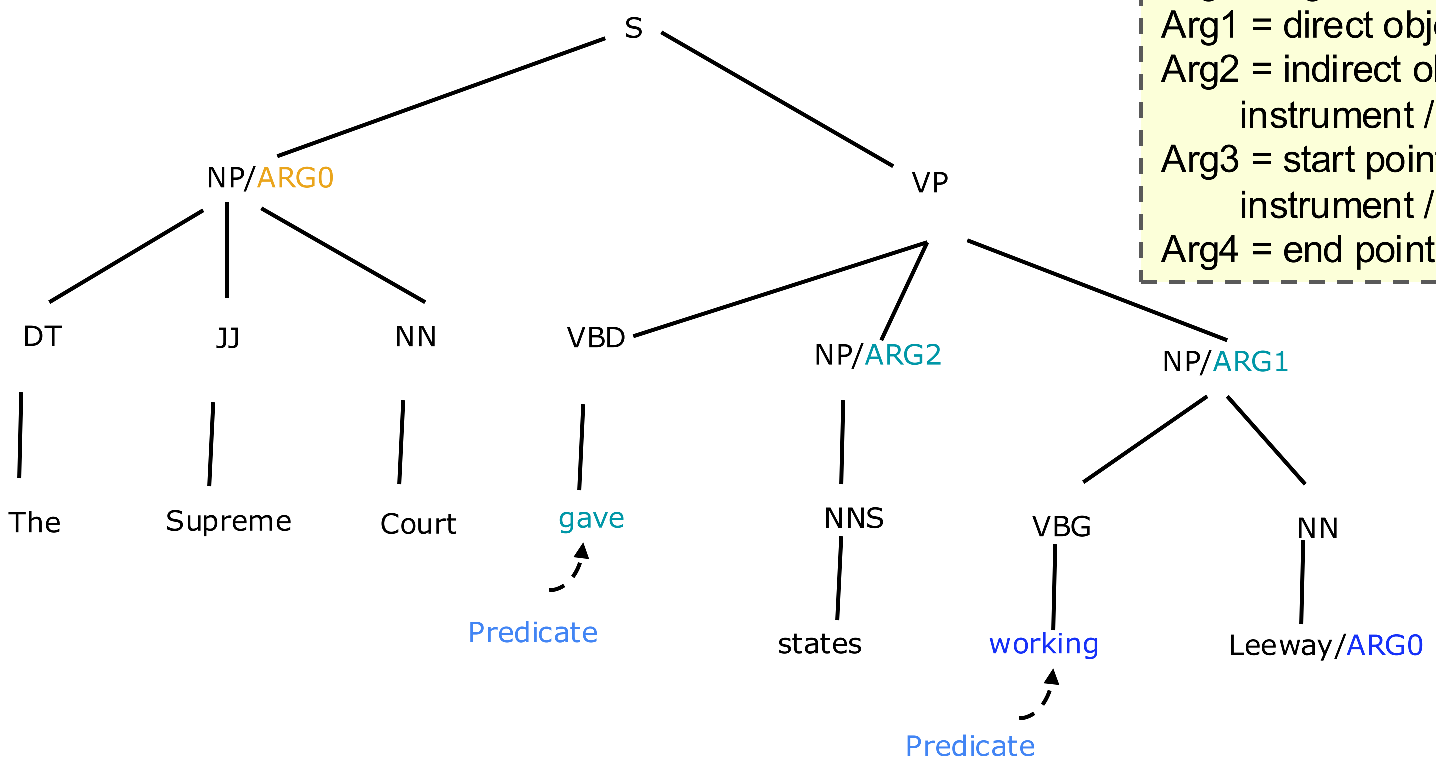
Other Examples: Classical NLP Pipeline

- POS tagging and Syntactic Parsing



Other Examples: Classical NLP Pipeline

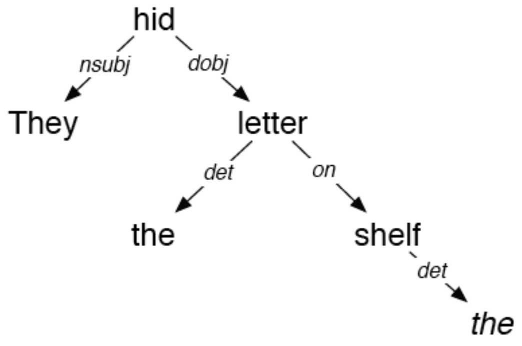
- Semantic Role Labeling



CORE ARGUMENTS
Arg0 = agent
Arg1 = direct object / theme / patient
Arg2 = indirect object / benefactive / instrument / attribute / end state
Arg3 = start point / benefactive / instrument / attribute
Arg4 = end point

Other Examples: Classical NLP Pipeline

- Dependency Parsing



They hid the letter on the shelf.

Argument Dependencies	Description
nsubj	nominal subject
csubj	clausal subject
dobj	direct object
iobj	indirect object
pobj	object of preposition
Modifier Dependencies	**Description**
tmod	temporal modifier
appos	appositional modifier
det	determiner
prep	prepositional modifier

Other Examples: Classical NLP Pipeline

CONVAI

- Named Entity Extraction

Since its inception in 2001, <name ID="1" type="organization">Red</name>has caused a stir in <name ID="2" type="location">Northeast Ohio</name> by stretching the boundaries of classical by adding multi-media elements to performances and looking beyond the expected canon of composers.

Under the baton of <name ID="3" type="organization"> Red</name> Artistic Director <name ID="4" type="person">Jonathan Sheffer</name>, <name ID="5" type="organization">Red</name> makes its debut appearance at <name ID="6" type="organization"> Kent State University</name> on March 7 at 7:30 p.m.

Other Examples: Classical NLP Pipeline

- Coreference Resolution

But the little prince could not restrain admiration:

"Oh! How beautiful you are!"

"Am I not?" the flower responded, sweetly. "And I was born at the same moment as the sun . . ."

The little prince could guess easily enough that she was not any too modest--but how moving--and exciting--she was!

"I think it is time for breakfast," she added an instant later. "If you would have the kindness to think of my needs--"

And the little prince, completely abashed, went to look for a sprinkling-can of fresh water. So, he tended the flower.

Other Examples: NLP Tasks

- Machine Translation

  $f($ “Kitap masada.” $) =$ “The book is on the table.”

- Question Answering

  $f($“Who wrote Leviathan?”$) =$ “Paul Auster”

- Summarization

  $f($  $) =$ 

Topic for Today: Reminders

- Machine Learning (ML) for NLP
- ML Examples: NLP Tasks
- $\boxed{\text{ML Basics}}$
- Calculus Reminders
- Deep Learning
- Linear Regression – Case Study

Kinds of Machine Learning

- Supervised learning
- Unsupervised learning
- Self-supervised learning
- Reinforcement learning

Supervised Learning

- Learning functions for predicting **targets** from given inputs, using pairs of **inputs and targets.**

- The targets, which we often call ***labels***, are generally denoted by y.

- The input data, also called the *features* or covariates, are typically denoted $\mathbf{x}$.

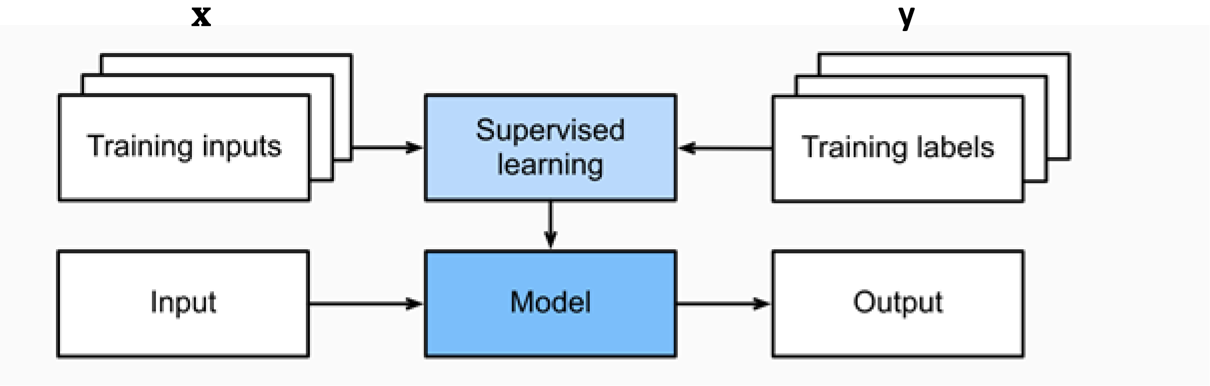
Supervised Learning (cont.)

x : "It makes too much noise."
function input

y: - (negative)
function output

Training
:

Testing
:



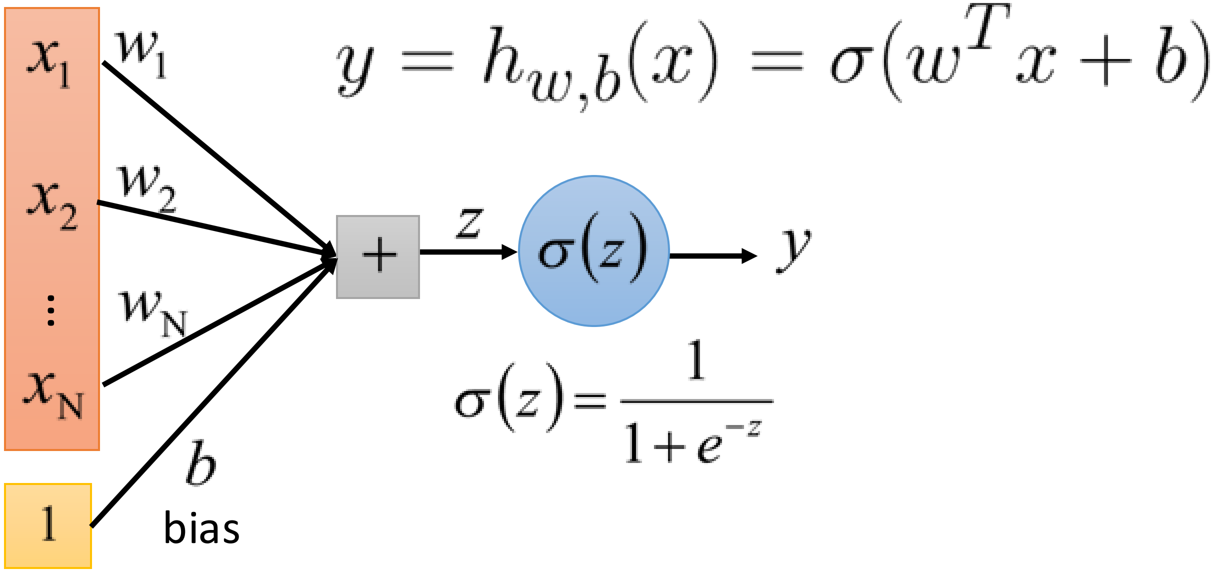
Supervised Learning (cont.)

- Each (input, target) pair is called an ***example*** or an ***instance***.
- Supervised learning learns the model/function from examples.
- For example, machine translation models are trained from pairs of source and target language sentences.

Classification Example: Apartment search

- Whether an apartment will be preferred by a user or not?
- Features:
  - Number of rooms, e.g., 1, 2, 3
  - Whether the apartment has a dish washer or not, e.g. yes, no
  - Style of the apartment building, e.g., high-rise, low-rise, duplex,
  - …

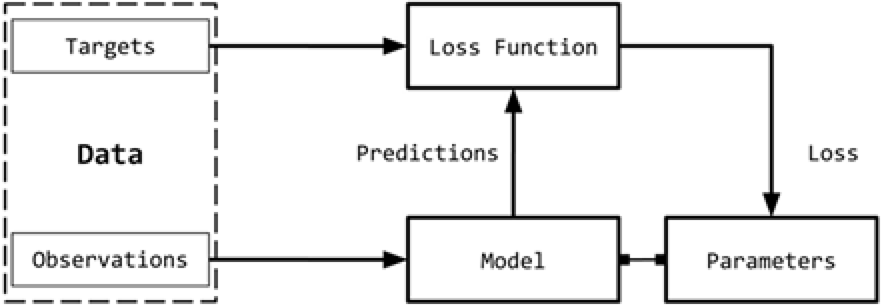
Example Function: a Single Neuron



w, b are the parameters of this neuron

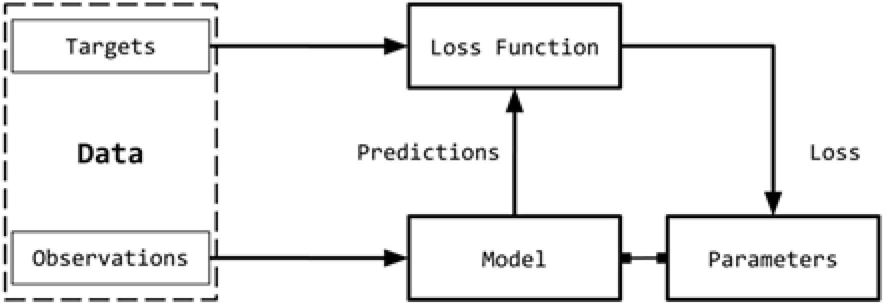
Terminology

- **Observations**: inputs, x.
- **Targets**: labels, y, corresponding to the observations (also called ground truth sometimes).
- **Model**: Function that takes an observation and predicts the value of its target label.



More Terminology

- **Parameters**: weights (for deep learning), w, that parameterize the model.
- **Predictions**: Also called estimates, target values estimated by the model, $\hat{y}$.
- **Loss Function**: a function (denoted $\mathcal{L}$) that compares how far off a prediction is from its target. The lower the loss, the better is the model at predicting the target.



Supervised Learning – Regression

- When the targets take arbitrary values in a range (instead of specific categories such as +/- for sentiment classification), we call it a **regression** problem.
- Examples:
  - Estimating the price of a house given attributes such as location, square footage, number of rooms, etc.
  - Predicting how many miles can a dog run given attributes such as its breed, age, size, etc.

Supervised Learning – Classification

- When the targets take a limited number of pre-defined categories, we call it a classification problem.
- **Binary classification:** only two values
  - Positive/negative sentiment classification
  - Sentence segmentation from speech: each word boundary is classified into sentence boundary versus not a sentence boundary.
  - Sentence segmentation for text: each "." is classified into two classes similarly.
- **Multi-class classification:** more than two values
  - Digit recognition: each digit is classified into one of 10 values for 0,...,9.

Supervised Learning – Tagging

- Some tasks are not simply binary or multi-class classification tasks, but we need to
  - Tagging images with multiple objects
  - Tagging tokens/words in natural language utterances

Locations in utterances:
I want to fly from Boston to Seattle via Chicago. =>
I want to fly from Boston to Seattle via Chicago.

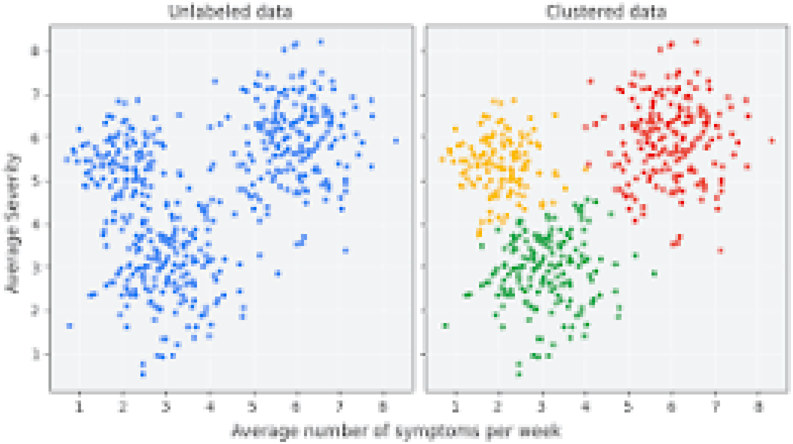
Sequence Tagging

Kinds of Machine Learning

- Supervised learning
- Unsupervised learning
- Self-supervised learning
- Reinforcement learning

Unsupervised Learning

- Finding previously unknown patterns in data set without pre-existing labels.
  - Clustering
  - Zero-Shot Learning
  - Few-Shot Learning
  - Semi-supervised Learning
  - Weakly-supervised Learning
  - Transfer Learning



Kinds of Machine Learning

- Supervised learning
- Unsupervised learning
- Self-supervised learning
- Reinforcement learning

Self-supervised Learning

- Benefiting from enormous amount of data available on the web and other resources.
- Examples:
    - For text, masking tokens and predicting them or predicting the next token,
    - For image, image reconstruction, and
    - For videos, predicting video frames.

Kinds of Machine Learning

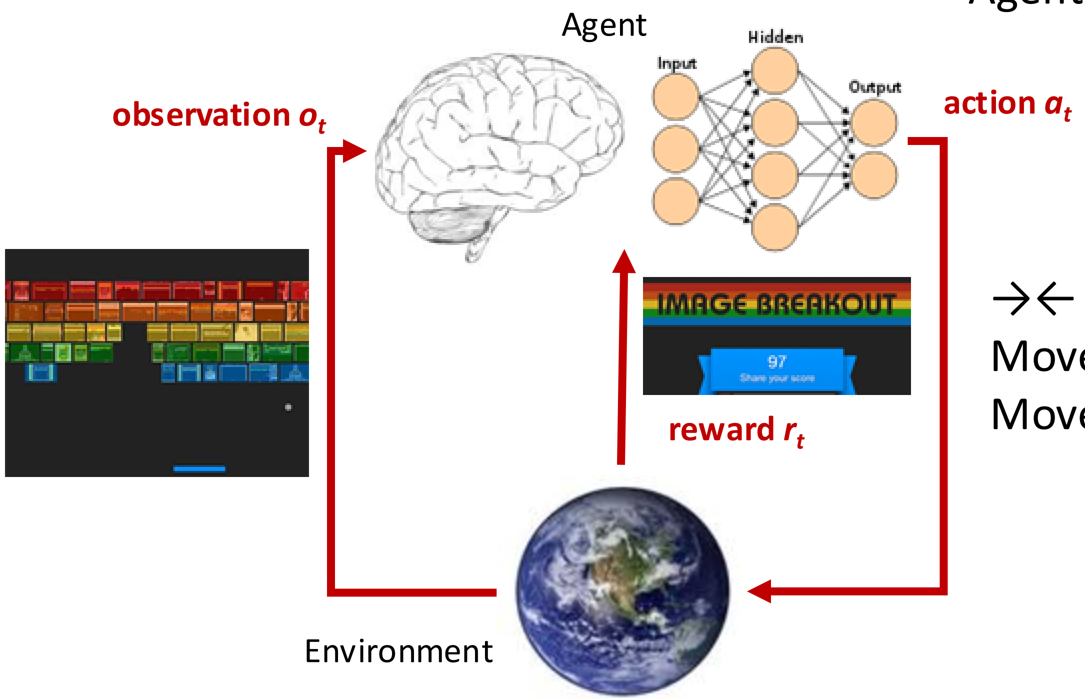
- Supervised learning
- Unsupervised learning
- Self-supervised learning
- Reinforcement learning

Reinforcement Learning

- Learning action prediction policies through interactions with an environment over a series of *timesteps*.
  - At each timestep t, the learning agent receives some observation o_t from the environment and must choose an action a_t that is subsequently transmitted back to the environment via some mechanism.
  - Finally, the agent receives a reward r_t from the environment.
  - The agent then receives a subsequent observation, and chooses a subsequent action, and so on.
  - A *policy* is a function that maps from observations to actions.

Agent and Environment

- No labels, just a reward
- Agent can exploit and explore



→←
MoveRight
MoveLeft

Topic for Today: Reminders

- Machine Learning (ML) for NLP
- ML Examples: NLP Tasks
- ML Basics
- Calculus Reminders
- Deep Learning
- Linear Regression – Case Study

Derivatives and Differentiation

- Training deep learning models: updating them successively so that they get better as they see more data

- Getting better: minimizing a loss function

- A crucial step in nearly all deep learning optimization algorithms.

- Loss functions that are differentiable with respect to our model’s parameters.

Derivatives and Differentiation (cont.)

- Suppose we have function $f:\mathbb{R}\rightarrow\mathbb{R}$ with scalar input and outputs.
- The *derivative* of f is defined as:

$$f'(x) = \lim_{h\rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

- If $f'(a)$ exists, f is said to be *differentiable* at a.
- If f is differentiable at every number of an interval, then this function is differentiable on this interval.
- Alternative notations for derivatives:

$$f'(x) = y' = \frac{dy}{dx} = \frac{df}{dx} = \frac{d}{dx}f(x) = Df(x) = D_x f(x),$$

Rules for Differentiating Common Functions

- $DC = 0$ (C is a constant),
- $Dx^n = nx^{n-1}$ (the *power rule*, n is any real number),
- $De^x = e^x$,
- $D\ln(x) = 1/x$.

Rules for Differentiating Common Functions

CONVAI

$$\frac{d}{dx}[Cf(x)] = C\frac{d}{dx}f(x),$$

Constant multiple rule

$$\frac{d}{dx}[f(x) + g(x)] = \frac{d}{dx}f(x) + \frac{d}{dx}g(x),$$

Sum rule

$$\frac{d}{dx}[f(x)g(x)] = f(x)\frac{d}{dx}[g(x)] + g(x)\frac{d}{dx}[f(x)],$$

Product rule

$$\frac{d}{dx}\left[\frac{f(x)}{g(x)}\right] = \frac{g(x)\frac{d}{dx}[f(x)] - f(x)\frac{d}{dx}[g(x)]}{[g(x)]^2}.$$

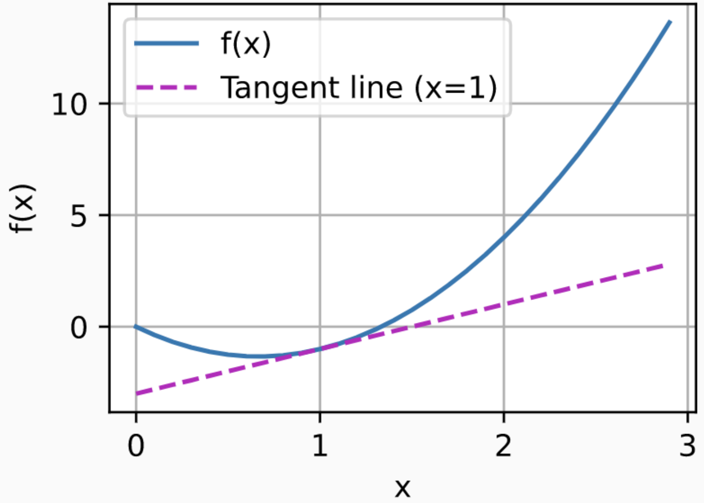
Quotient rule

Example

- Given function, $u = f(x) = 3x^2 - 4x$
- We can apply the previous set of rules to get its derivative:

$$u' = f'(x) = 3\frac{d}{dx}x^2 - 4\frac{d}{dx}x = 6x - 4$$

- For x=1, we have u'=2.
- This derivative is also the slope of the tangent line to the curve u=$f(x)$ when x=1.



Partial Derivatives

- If y is a function with n variables, i.e.,

$$y = f(x_1, x_2, \ldots, x_n)$$

- The **partial derivative** of y with respect to its i^{th} parameter x_i is:

$$\frac{\partial y}{\partial x_i} = \lim_{h \to 0} \frac{f(x_1, \ldots, x_{i-1}, x_i + h, x_{i+1}, \ldots, x_n) - f(x_1, \ldots, x_i, \ldots, x_n)}{h}.$$

Gradients

- For functions with multiple variables, we obtain the **gradient vector** of the function by concatenating partial derivatives of that function w.r.t. all its variables.

- For example, for $f:\mathbb{R}^n \rightarrow \mathbb{R}$ is a function with an n-dimensional vector $\mathbf{x}=[x_1,x_2,\ldots,x_n]^\top$ input and a scalar output. The gradient of the function $f(\mathbf{x})$ w.r.t. $\mathbf{x}$ is a vector of n partial derivatives:

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \left[\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}, \ldots, \frac{\partial f(\mathbf{x})}{\partial x_n}\right]^\top$$

The Chain Rule

- Suppose that functions $y=f(u)$ and $u=g(x)$ are both differentiable, then,

$$\frac{dy}{dx} = \frac{dy}{du}\frac{du}{dx}$$

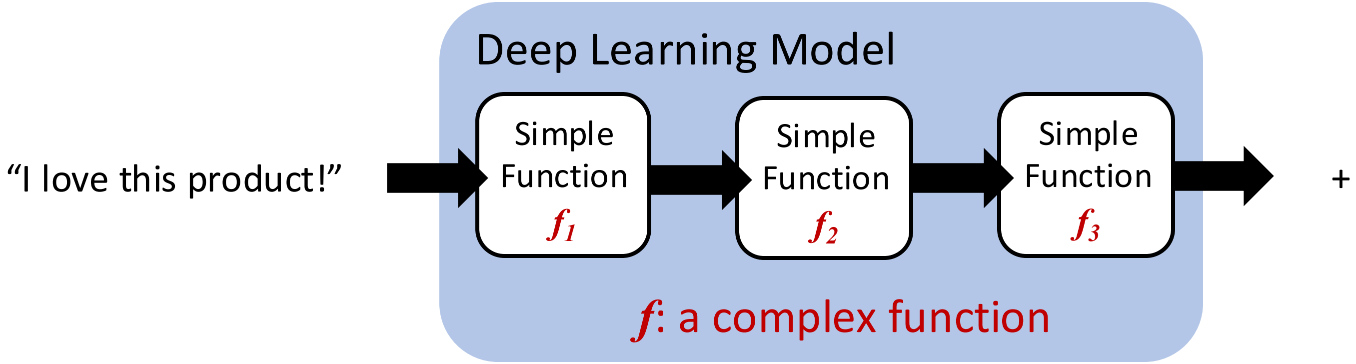
- More general scenario, where a differentiable function y has variables $u_1, u_2, \ldots, u_m$, where each differentiable function u_i has variables $x_1, x_2, \ldots, x_n$. Then, for any $i = 1, 2, \ldots, n$,

$$\frac{dy}{dx_i} = \frac{dy}{du_1}\frac{du_1}{dx_i} + \frac{dy}{du_2}\frac{du_2}{dx_i} + \cdots + \frac{dy}{du_m}\frac{du_m}{dx_i}$$

Topic for Today: Reminders

- Machine Learning (ML) for NLP
- ML Examples: NLP Tasks
- ML Basics
- Calculus Reminders
- Deep Learning
- Linear Regression – Case Study

Learning Stacked Functions

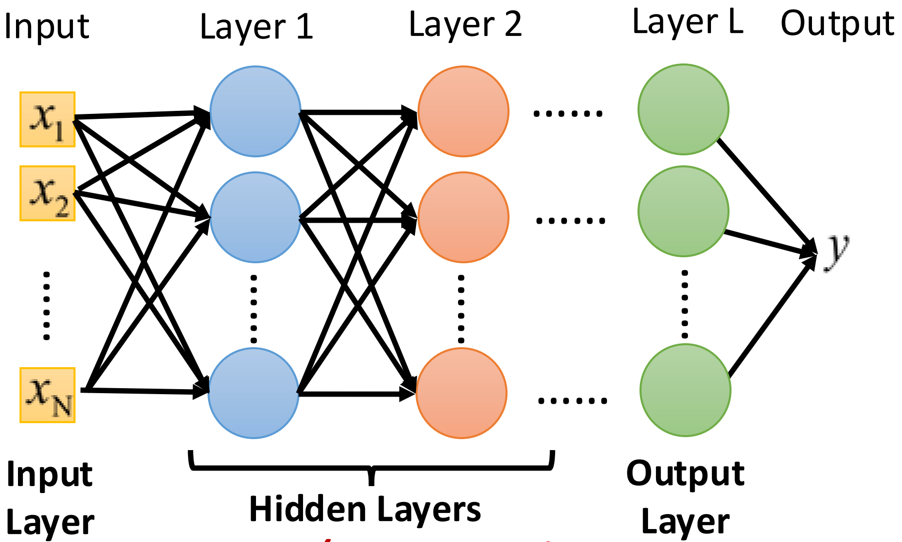


End-to-end training: what each function should do is learned automatically

Deep learning usually refers to *neural network*-based model

Learning Stacked Functions (cont.)

vector $\boldsymbol{x}$
"I love this product."



label $\boldsymbol{y}$
+

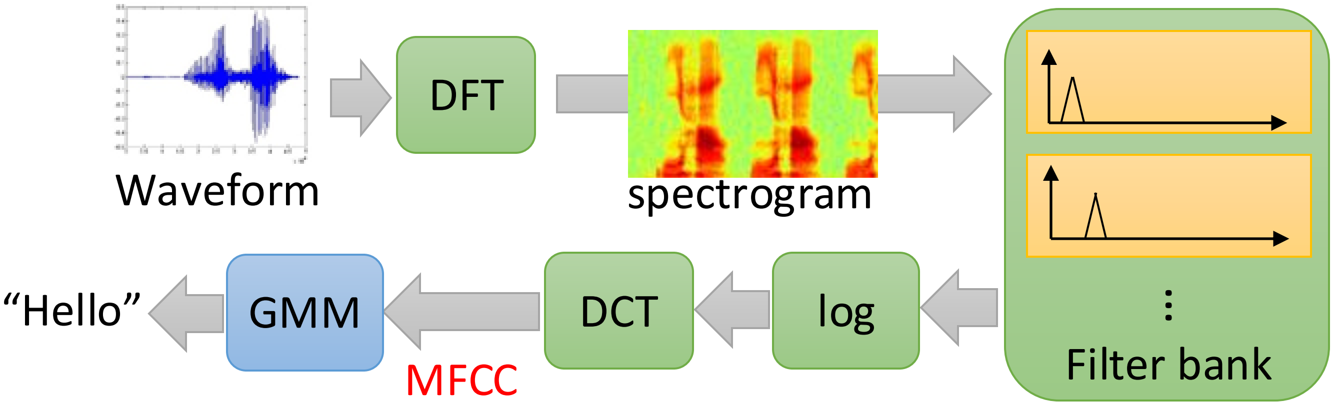
Features / Representations

Representation Learning attempts to learn good features/representations

Deep Learning attempts to learn (multiple levels of) representations and an output

Example: Speech Recognition (Before DL)

- Shallow Model

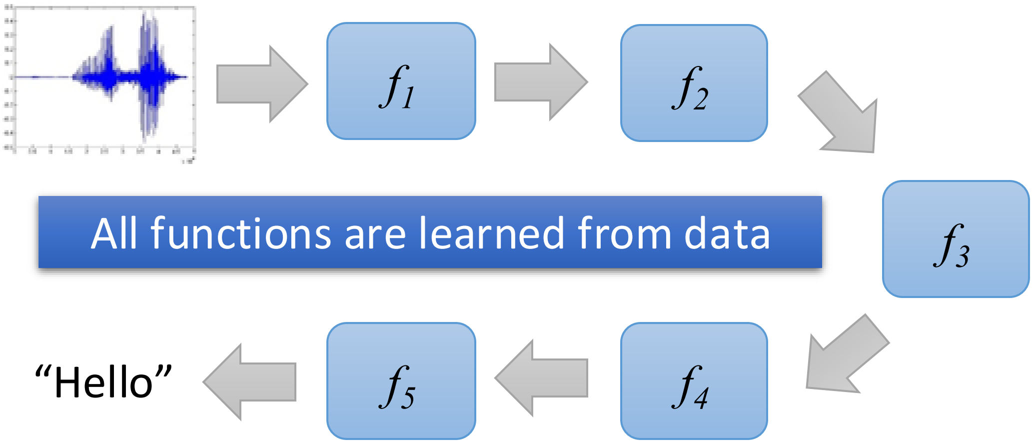


Each box is a simple function in the production line:

:hand-crafted :learned from data

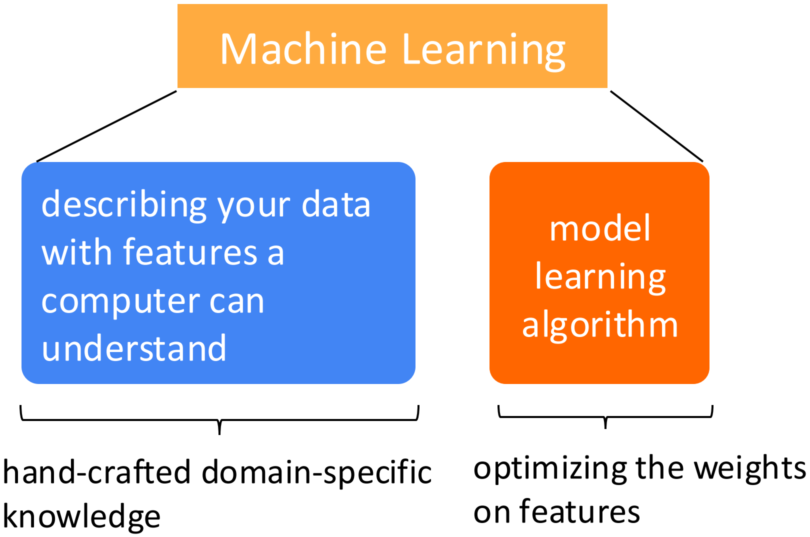
Example: Speech Recognition (With DL)

- Deep Model



Less engineering labor, but machine learns more

Machine Learning



Deep Learning

Deep Learning

representations learned by machine

model learning algorithm

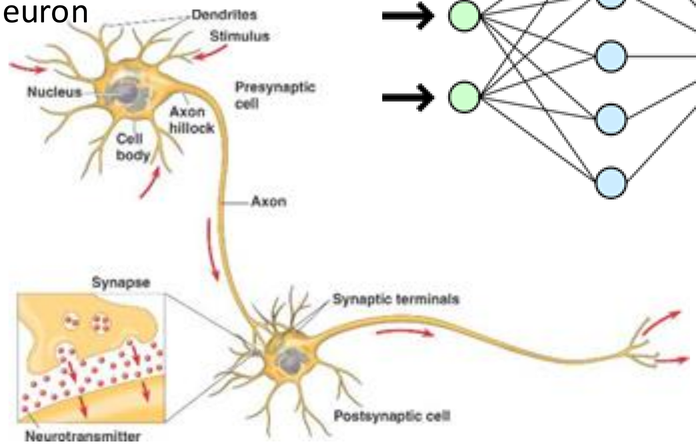
automatically learned internal knowledge

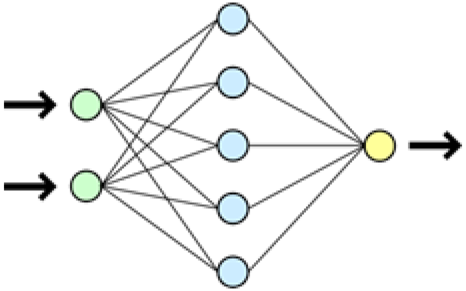
optimizing the weights on features

Deep learning usually refers to *neural network* based model

Inspired by the Human Brain

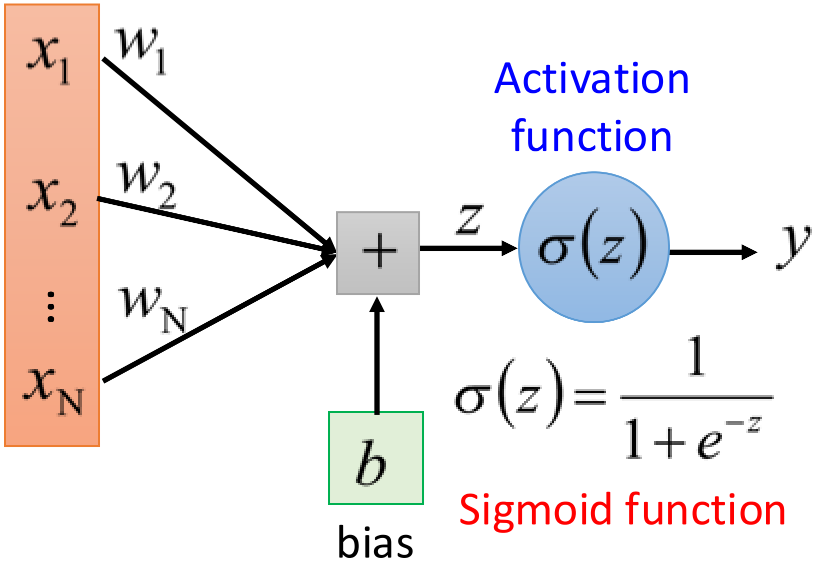
neuron



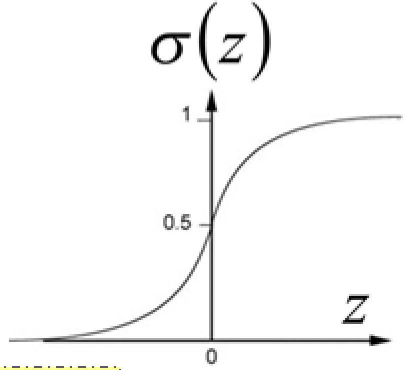


Neurotransmission section of: https://en.wikipedia.org/wiki/Human_brain

A Single Neuron



$$\sigma(z) = \frac{1}{1+e^{-z}}$$



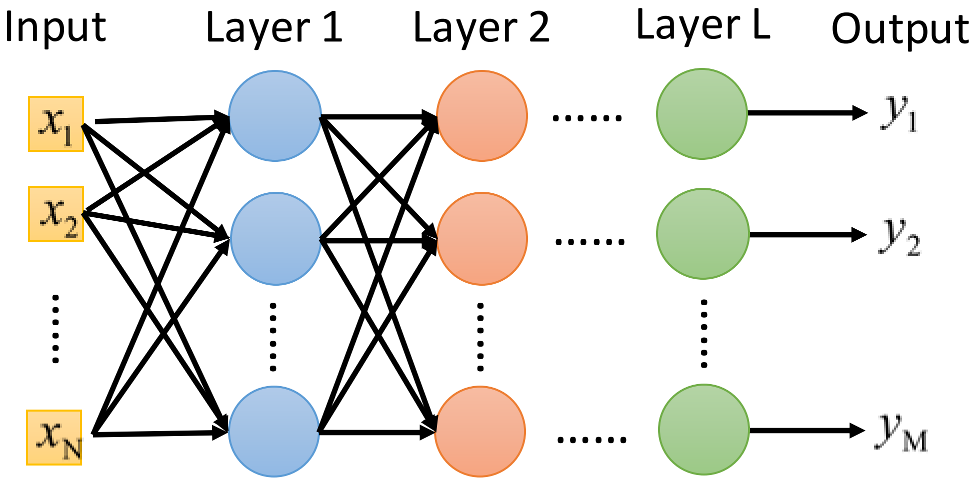
Each neuron is a very simple function

Deep Neural Network

A neural network is a complex function:

$$f : R^N \rightarrow R^M$$

- Cascading the neurons to form a neural network



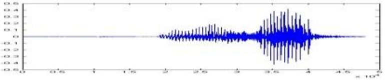
Each layer is a simple function in the production line

Topic for Today: Reminders

- Machine Learning (ML) for NLP
- ML Examples: NLP Tasks
- ML Basics
- Calculus Reminders
- Deep Learning
- Linear Regression – Case Study

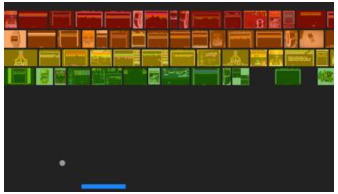
Learning ≈ Looking for a Function

- Speech Recognition
- Handwriting Recognition
- Weather forecast
- Play video games

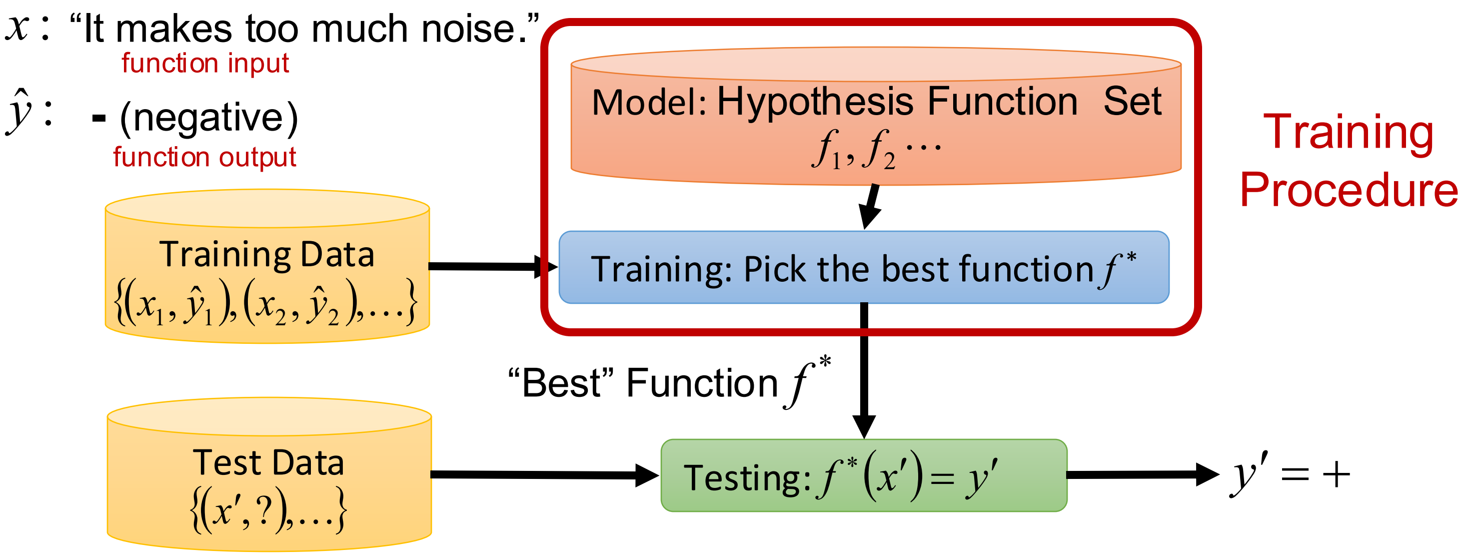
$f($  $)=$ "the book"

$f($  $)=$ "2"

$f($ Thursday $)=$ "  Saturday"

$f($  $)=$ "move left"

Machine Learning Framework



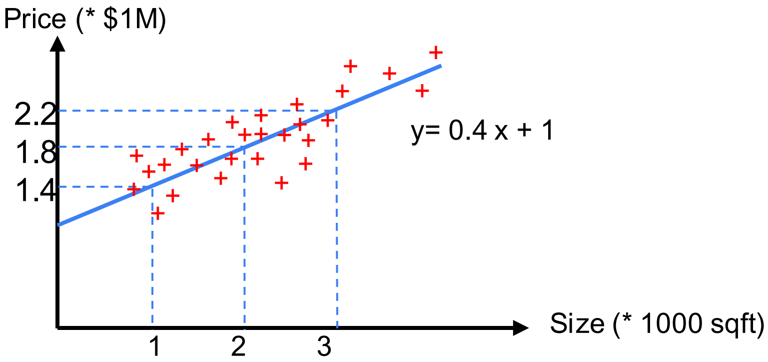
Training aims to pick the best function given the observed examples
Testing predicts the label using the learned function

Linear Regression

- **Regression:** modeling the relationship between data points $\mathbf{x}$ and corresponding real-valued targets y
  - Example: Predicting prices of homes
  - Label: Price, $y^{(i)}$
  - Features: Size, Age
    - $x^{(i)}=[x^{(i)}_1, x^{(i)}_2]$

2019 Sales:

Size (sqft)	Price
2100	$1.90M
1600	$1.56M
3200	$2.40M
…	…



Linear Regression (cont.)

- Assumption:
  - the relationship between the *features* $\mathbf{x}$ and targets y is linear,
  - i.e., y can be expressed as a weighted sum of the inputs $\mathbf{x}$, give or take some noise on the observations

price = $w_1 . x_1 + w_2 . x_2 + b$

Bias
(offset or intercept)

weights

Linear Regression: Modeling Bivariate Data

CONVAI

- Bivariate data: $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- Model: $y_i = f(x_i) + e_i$ Random error
- Supervised approach!
- Model allows us to predict the value of y for any given value of x.
  - x is called the independent or predictor variable.
  - y is the dependent or response variable.

Other examples of f(x)

- lines: $f(x) = mx + b$
- polynomials: $f(x) = ax^2 + bx + c$
- others: $f(x) = a/x + b$

  $f(x) = a\ sin(x) + b$

  $f(x) = a\ sqrt(x)$

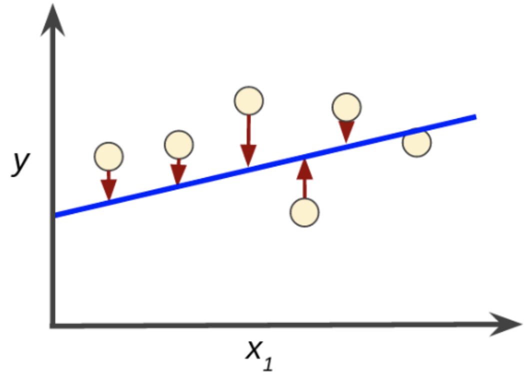
Assumptions of Linear Regression

- The relationship between x and y is linear
- y is distributed normally at each value of x, and the variance of y at every value of x is the same
- The observations are independent

Mean Squared Error

- y_i is the actual value, $f(x_i)$ is the prediction, and their difference is the error.
- Commonly used function, (mean) squared error:

$$MS = \sum_{i=1}^{n} (y_i - f(x_i))^2$$



Case Study: $f(x) = mx + b$

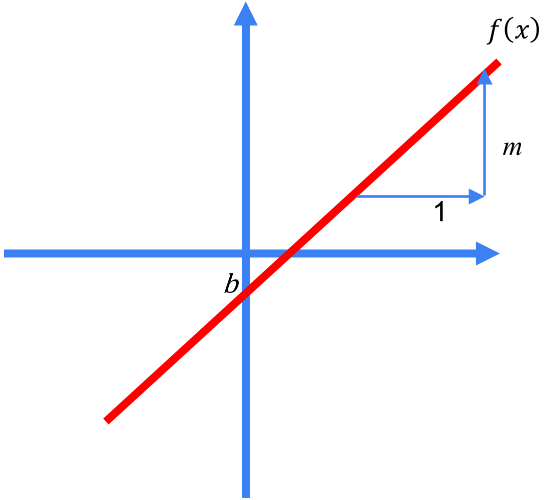
- (Mean) squared error:

$$MS(m,b) = \sum_{i=1}^{n} (y_i - mx_i - b)^2$$

- To estimate the values that minimize the mean squared error, we need to take the derivative with respect to the two variables m and b:

$$\frac{\partial MS(m,b)}{\partial b} = 0 \quad \text{and} \quad \frac{\partial MS(m,b)}{\partial m} = 0$$

Case Study: $f(x) = mx + b$



- b is called bias.
- m is the slope of $f(x)$
- Every time x moves by 1, y moves by m

Case Study: $f(x) = mx + b$

- We need to solve these two equations to find m and b, to be able to find the function $f(x_i)$.
- Beginning with the first one, for b:

$$\frac{\partial SS(m,b)}{\partial b} = \frac{\partial \sum_{i=1}^{n}(y_i - mx_i \ - b)^2}{\partial b} = \sum_{i=1}^{n} 2\,(y_i - mx_i \ - b)(-1) = 0$$

$$\sum_{i=1}^{n} y_i \ - m \sum_{i=1}^{n} x_i - nb = 0$$

$$b = \frac{1}{n}\sum_{i=1}^{n} y_i - m\,\frac{1}{n}\sum_{i=1}^{n} x_i$$

Case Study: $f(x) = mx + b$

- We can use $\bar{y}$ and $\bar{x}$ to represent the mean values of the x and y coordinates, that is all the x and y values in our data:

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \quad \text{and} \quad \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

- Then we can simply re-write the equation for b:

$$b = \frac{1}{n}\sum_{i=1}^{n} y_i - m\,\frac{1}{n}\sum_{i=1}^{n} x_i$$

$$b = \bar{y} - m\,\bar{x}$$

Case Study: $f(x) = mx + b$

- Now, let's find the value of m:

$$\frac{\partial SS(m,b)}{\partial m} = \frac{\partial \sum_{i=1}^{n}(y_i - mx_i - b)^2}{\partial m} = \frac{\partial \sum_{i=1}^{n}(y_i - mx_i - \bar{y} + m\,\bar{x})^2}{\partial m} = 0$$

$$\sum_{i=1}^{n} 2\,(y_i - mx_i - \bar{y} + m\,\bar{x})\,(-x_i + \bar{x}) = 0$$

$$\sum_{i=1}^{n} -y_i x_i + y_i\,\bar{x} + m\,x_i^2 - m\,x_i\,\bar{x} + \bar{y}\,x_i - \bar{y}\,\bar{x} - m\,\bar{x}\,x_i + m\,\bar{x}^2 = 0$$

Case Study: $f(x) = mx + b$

$$\sum_{i=1}^{n} -y_i x_i + y_i\,\bar{x} + m\,x_i^2 - m\,x_i\,\bar{x} + \bar{y}\,x_i - \bar{y}\,\bar{x} - m\,\bar{x}\,x_i + m\,\bar{x}^2 = 0$$

$$\sum_{i=1}^{n} -y_i x_i + y_i\,\bar{x} + \bar{y}\,x_i - \bar{y}\,\bar{x} + m\,x_i^2 - m\,x_i\,\bar{x} - m\,\bar{x}\,x_i + m\,\bar{x}^2 = 0$$

$$\sum_{i=1}^{n} -y_i x_i + y_i\,\bar{x} + \bar{y}\,x_i - \bar{y}\,\bar{x} + m\,x_i^2 - 2m\,x_i\,\bar{x} + m\,\bar{x}^2 = 0$$

Case Study: $f(x) = mx + b$

$$\sum_{i=1}^{n} \underbrace{-y_i x_i + y_i\,\bar{x} + \bar{y}\,x_i - \bar{y}\,\bar{x}} + \underbrace{m\,x_i^2 - 2m\,x_i\,\bar{x} + m\,\bar{x}^2} = 0$$

$$\sum_{i=1}^{n} (\bar{y} - y_i)(x_i - \bar{x}) + m\,(x_i - \bar{x})^2 = 0$$

Therefore,

$$m = \frac{\sum_{i=1}^{n} (\bar{y} - y_i)(\bar{x} - x_i)}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

Case Study: $f(x) = mx + b$

Summary:

$$b = \bar{y} - m\,\bar{x} \qquad\qquad m = \frac{\sum_{i=1}^{n}(\bar{y} - y_i)(\bar{x} - x_i)}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

Linear Regression (cont.)

- Goal: choose the weights w and bias b to best fit the true values of y observed in the data
- When our inputs consist of d features, we express our prediction $\hat{y}$ as:

$$\hat{y} = w_1 \cdot x_1 + \ldots + w_d \cdot x_d + b$$

- Collecting all features into a vector $\mathbf{x}$ and all weights into a vector $\mathbf{w}$, our model can be expressed compactly using a dot product:

$$\hat{y} = \mathbf{w}^T \mathbf{x} + b$$

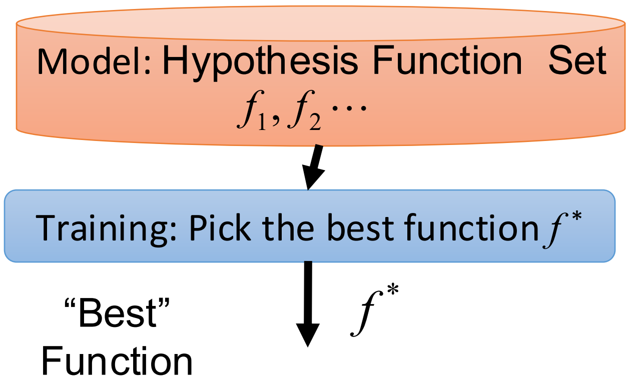
- $\mathbf{x}$: feature vector for a single data point
- $\mathbf{X}$: a collection of data points
- $\hat{\mathbf{y}}$: vector of estimated predictions

$$\hat{\mathbf{y}} = \mathbf{X}\mathbf{w} + b$$

Training Procedure: Searching for the best parameters

$\hat{\mathbf{y}} = \mathbf{X}\mathbf{w} + b$

Parameters: elements of **w and** b



- Q1. What is the model? (function hypothesis set)
- Q2. What does a "good" function mean?
- Q3. How do we find the "best" function?

Loss Function

- Measure of *fitness*, quantifies the distance between **y and y**̂.
- Sum of squared errors:

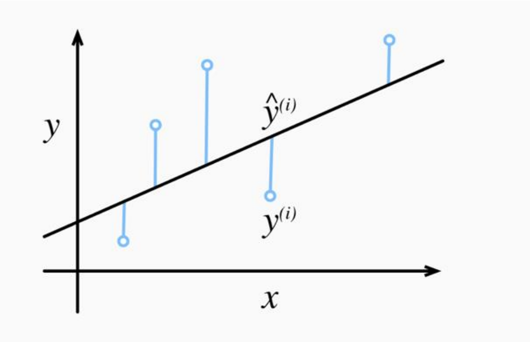
$$l^{(i)}(\mathbf{w},b) = \frac{1}{2}\left(\hat{y}^{(i)} - y^{(i)}\right)^2$$

- To measure quality over the entire training set,

we average over all K training examples:

$$L(\mathbf{w},b) = \frac{1}{K}\sum_{i=1}^{K} l^{(i)}(\mathbf{w},b)$$

$$= \frac{1}{K}\sum_{i=1}^{K} \frac{1}{2}\left(\mathbf{w}^\top\mathbf{x}^{(i)} + b - y^{(i)}\right)^2$$



Residual = truth - predicted

Training Procedure (cont.)

- Training the model ≈ search for parameters ($\mathbf{w}^*, b^*$) that minimize the total loss across all training samples:

$$\mathbf{w}^*, b^* = \mathrm{argmin}_{\mathbf{w},b}\ L(\mathbf{w}, b)$$

Simplification in Notation

- $\hat{\mathbf{y}} = \mathbf{X}\mathbf{w} + b$
- Append 1 to features and bias to the weights: $\mathbf{X}$ -> $[\mathbf{X}\ \mathbf{1}]$ and $\mathbf{w}$ -> $\begin{bmatrix} \mathbf{w} \\ b \end{bmatrix}$

$$\begin{pmatrix} x_1^{(1)}\ x_2^{(1)} \ldots x_n^{(1)} \\ x_1^{(2)}\ x_2^{(2)} \ldots x_n^{(2)} \\ \ldots \\ x_1^{(K)}\ x_2^{(K)} \ldots x_n^{(K)} \end{pmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \ldots \\ w_n \end{bmatrix} + b = \begin{pmatrix} x_1^{(1)}\ x_2^{(1)} \ldots x_n^{(1)}\ 1 \\ x_1^{(2)}\ x_2^{(2)} \ldots x_n^{(2)}\ 1 \\ \ldots \\ x_1^{(K)}\ x_2^{(K)} \ldots x_n^{(K)}\ 1 \end{pmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \ldots \\ w_n \\ b \end{bmatrix} = \mathbf{X}\ \mathbf{w}$$

Solution for Linear Regression

CONVAI

$$L(\mathbf{X}, y, \mathbf{w}) = \frac{1}{K} \, || \, y - \mathbf{X}.\mathbf{w} \, ||^{2}$$

$$\frac{d}{d\mathbf{w}} L(\mathbf{X}, y, \mathbf{w}) = \frac{2}{K} (y - \mathbf{X}.\mathbf{w})^{\mathrm{T}} \mathbf{X}$$

Loss is convex, hence the optimum solution is at: $\frac{d}{d\mathbf{w}} L(\mathbf{X}, y, \mathbf{w}) = 0$

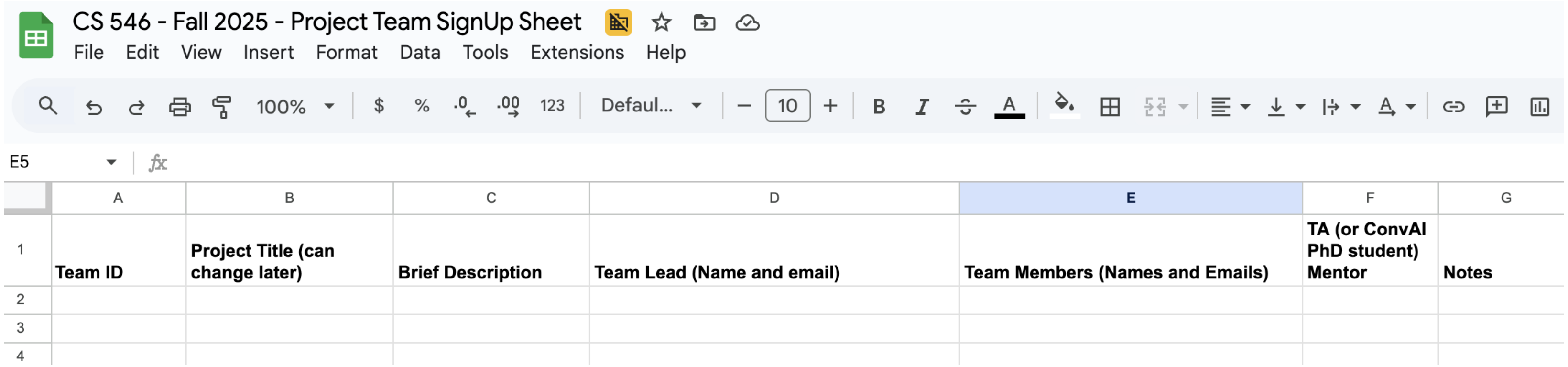
Therefore,

$\frac{2}{K} (y - \mathbf{X}.\mathbf{w^*})^{\mathrm{T}} \mathbf{X} = 0$ and $\mathbf{w^*} = (\mathbf{X}^{\mathrm{T}} \mathbf{X})^{-1} \mathbf{X} \, y$

Final Project Signup Sheet

https://docs.google.com/spreadsheets/d/1EJ_5Xby0mRhHFmSRSmxlv6Gws4Qs5T8P5JUiKYKAZcA/edit?usp=sharing

(Everyone should sign up by: Sept 23rd)
You can start a new team or join an existing one. Teams should be 4-6 people!



Final Project Teaming Tip

- Make sure that you can contribute with meaningful work before joining a team!

- Final reports will include a section on who did what…

Topics for Next Week

Tuesday:

- Gradient Descent
- Softmax Regression
- Multi-layer Perceptron

Thursday:

- Distributional Similarity
- Sparse Word Representations
- Word Embeddings and Word2Vec
- Language Modeling
- Unexpected things we learn with word embeddings