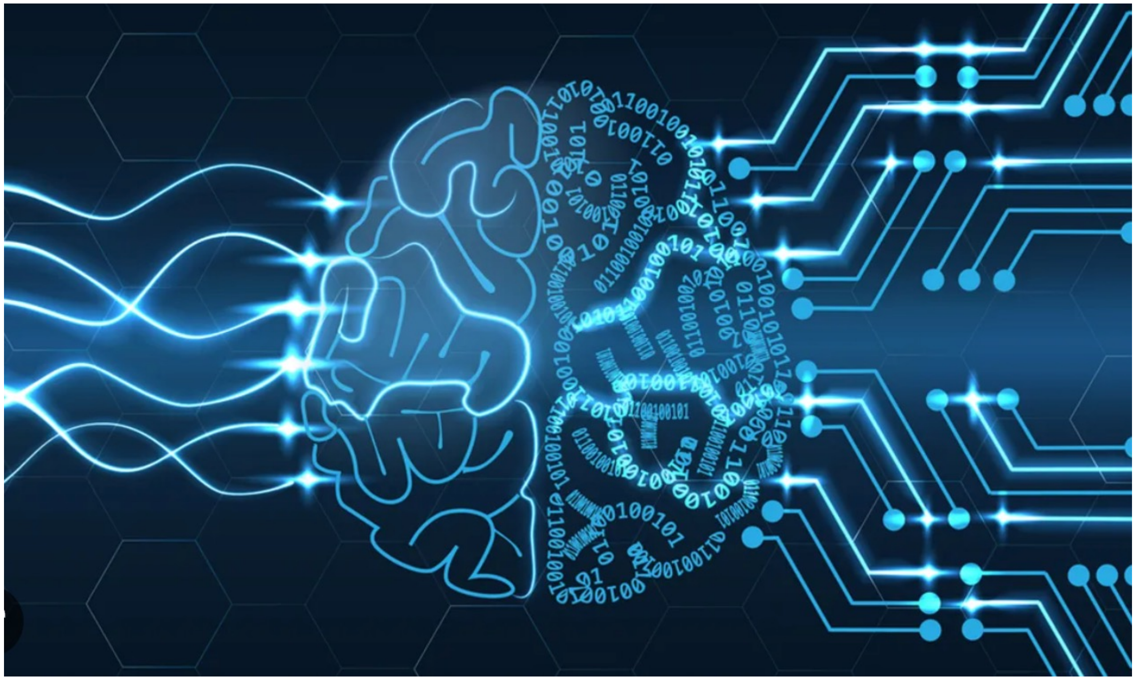
CS 4100: Introduction to AI

Wayne Snyder
Northeastern University

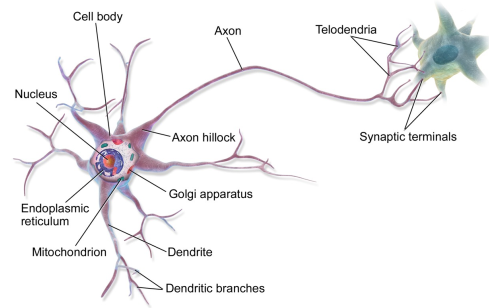
Lecture 19: Introduction to Deep Learning

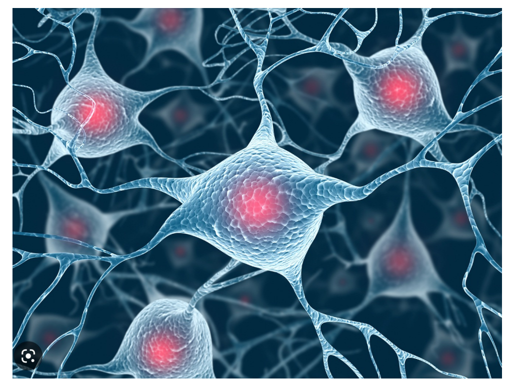


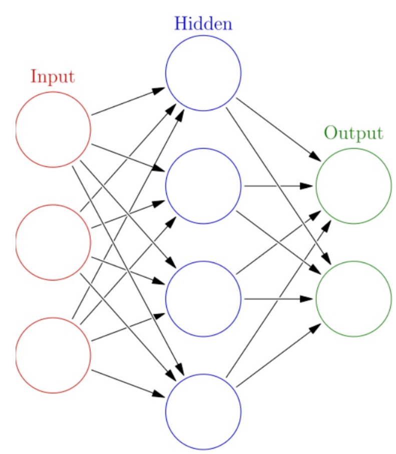
Introduction to Deep Learning

Deep Learning refers to Supervised Learning using an Artificial Neural Network, which has the following features:

- It is a network/graph of small computation units called artificial neurons, loosely modeled on the neurons in our brains, which send signals to each other. The signals are floating-point numbers.
- The network is typically organized in layers: the first layer is the input layer, the last is the output layer, and others are called hidden layers.
- A shallow network might have as few as 3 layers, and there is no theoretical limit to how many layers, or how wide the layers are.
- Generally, networks are very wide (many neurons in a layer) but not very deep.



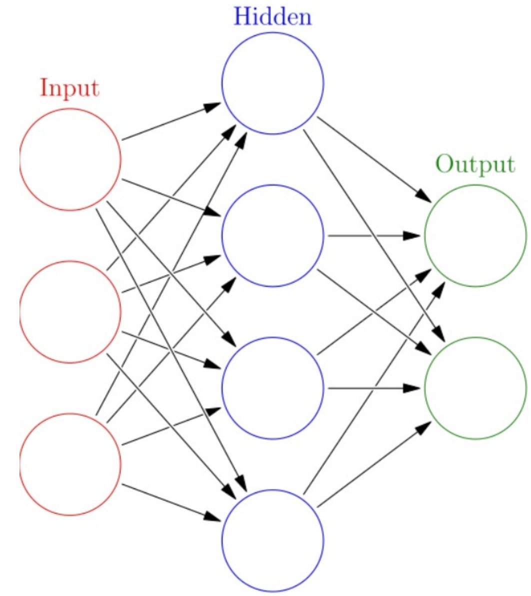


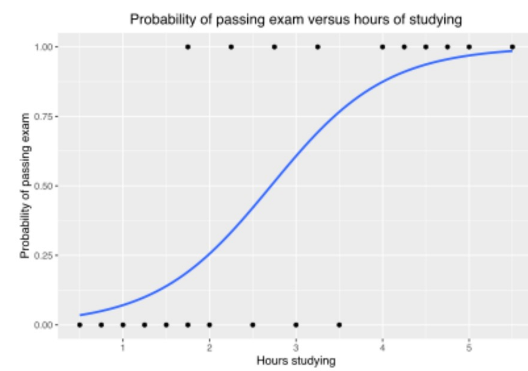


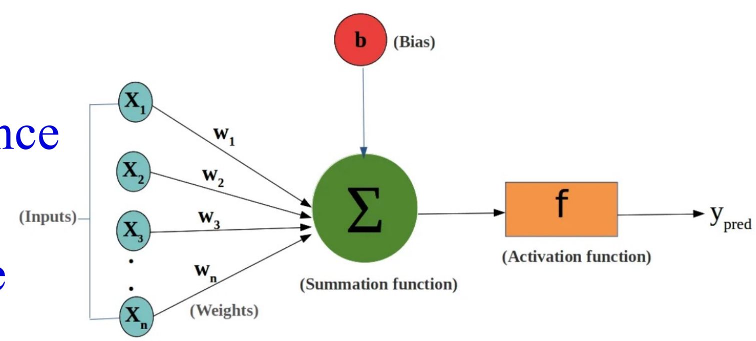
Introduction to Deep Learning

Features of artificial neural networks:

- The input layer takes an array/vector of floats, and the output layer produces an array of floats (sometimes just a single float or just 0/1). Thus, the network computes a function from vectors to vectors.
- In a feedforward network, each neuron in a hidden layer receives signals from all the neurons in the previous layer, computes a single floating-point number, which is sent to all the neurons in the next layer.
- The neuron processes its inputs using a non-linear function (typically, logistic regression), using a threshold function which determines the value of the output signal (typically in the range [0..1]).
- Each input to a neuron has a floating-point weight which determines the strength of the signal (importance of this float to the neuron).
- When a network is trained, it learns what weights are necessary to produce the required output.



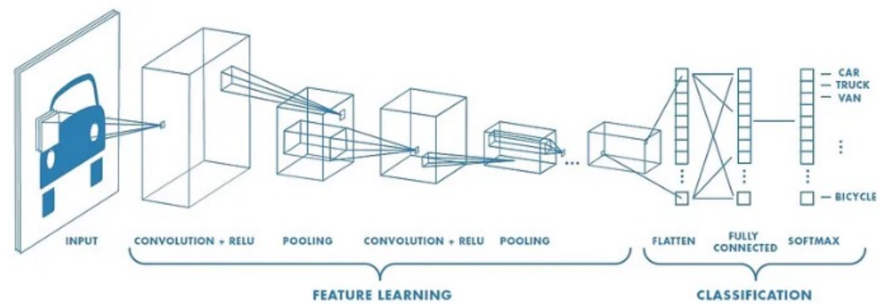


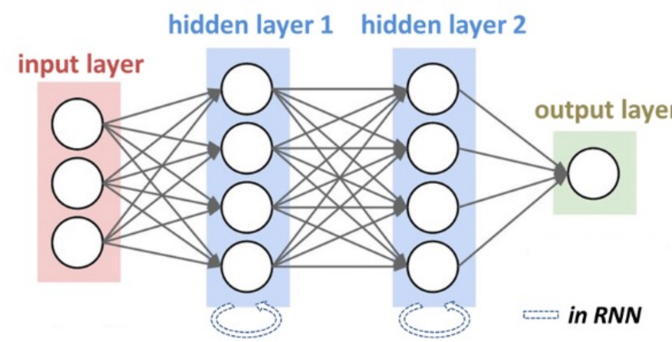


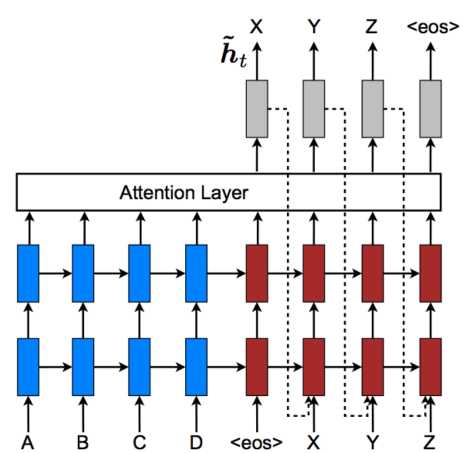
Introduction to Deep Learning

Features of artificial neural networks:

- Additional layers may perform data aggregation (e.g., convolution and pooling) or other kinds of data manipulation (e.g., softmax = transforming the output into a probability distribution).
- In a feedforward network, the network transforms an array of floats through the layers into another array of floats; in a sequence model, the inputs and outputs are sequences of vectors; and recurrent layers have cyclical connections which act as memory.
- BERT, GPT, and other large networks learn to pay Attention to complex patterns in the input sequence (e.g., words in a sentence).







Introduction to Deep Learning

Deep background: Linear Regression

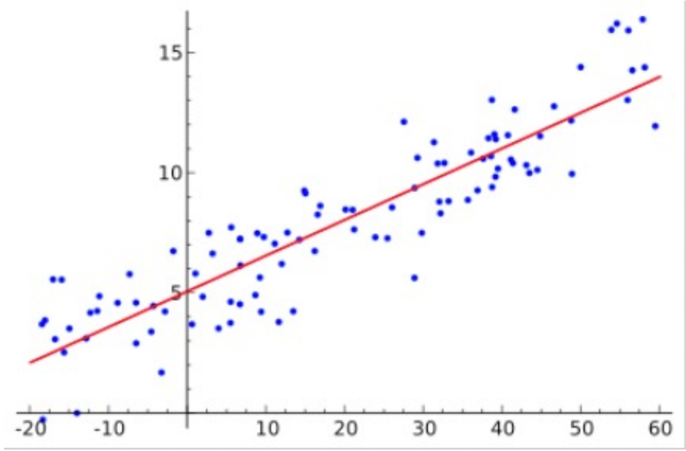
Introduction to Deep Learning

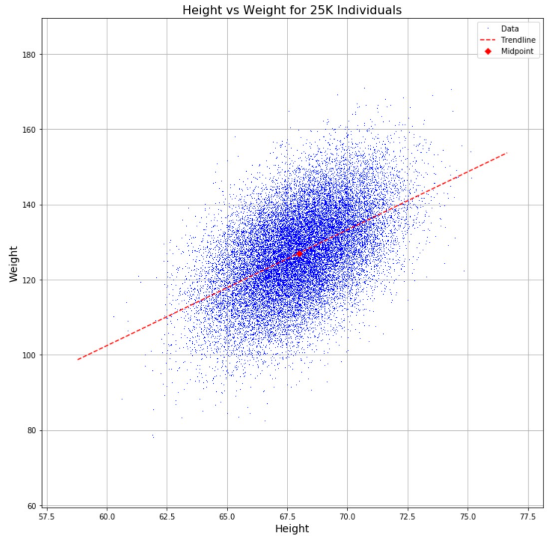
Digression: Linear Regression

Linear Regression relates some number of independent variables

$$X_1, \; X_2, \; \ldots, \; X_n$$

with a dependent or response variable Y. All are assumed to be real numbers. The values of Y form a trend line (= linear) showing the linear relationship of the input variables.





Introduction to Deep Learning

Digression: Linear Regression

There is a very simple formula from linear algebra which can be used to calculate the output line Y:

We thus have $Y = X \cdot W + E$ or

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \dots & x_n^{(1)} \\ 1 & x_1^{(2)} & x_1^{(2)} & \dots & x_n^{(2)} \\ & & \vdots & & \\ 1 & x_1^{(m)} & x_1^{(m)} & \dots & x_n^{(m)} \end{bmatrix} \times \begin{bmatrix} b \\ w_1 \\ \vdots \\ w_n \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_m \end{bmatrix}.$$

The least-squares estimates for W are given by the following formula:

$$W = \begin{bmatrix} b \\ w_1 \\ \vdots \\ w_n \end{bmatrix} = (X^T X)^{-1} X^T Y$$

Introduction to Deep Learning

Linear Regression: What is the "cost function"?

In linear regression, we define the error of the prediction as the MSE (mean square error) of the predictions:

$$\begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_m \end{bmatrix} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_m \end{bmatrix} - \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}.$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n} e_i^2 = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

and we have explicit formulae for finding the parameters for the slope and y-intercept of the regression line which minimizes the MSE:

$$W = (X^T X)^{-1} X^T Y$$

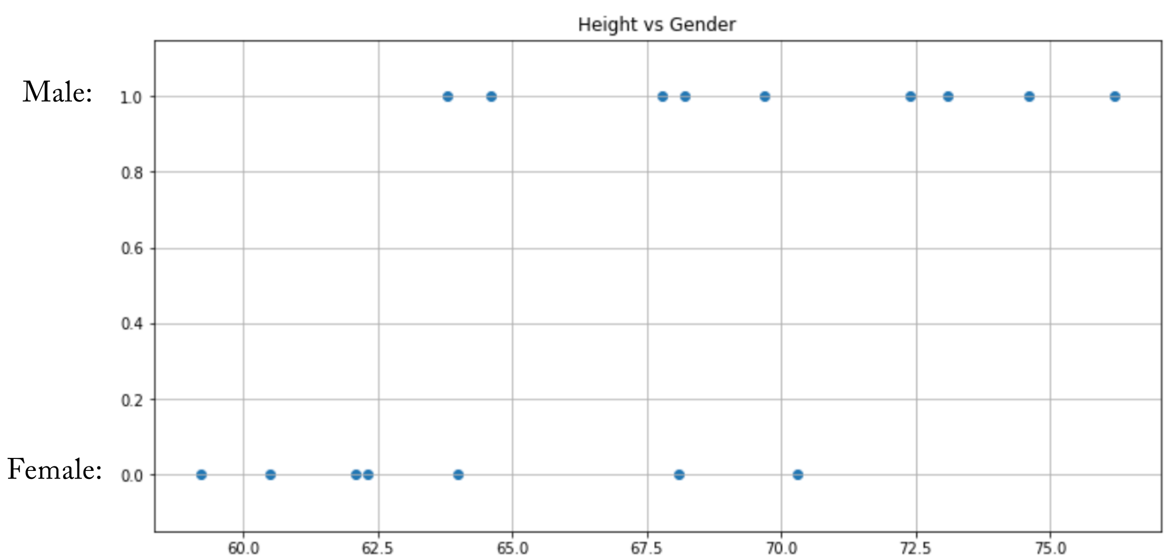
But what if we didn't have such an explicit formula?

Introduction to Deep Learning

Logistic Regression: A Motivating Example

But linear regression doesn't work for many problems! Suppose we attempt to classify 16 people as male or female depending on a single feature: their height. Men in general are taller than women (the average height of an American man is 5’ 9” and for women 5’ 4”),
X = height against Y = gender (1 for male, 0 for female):

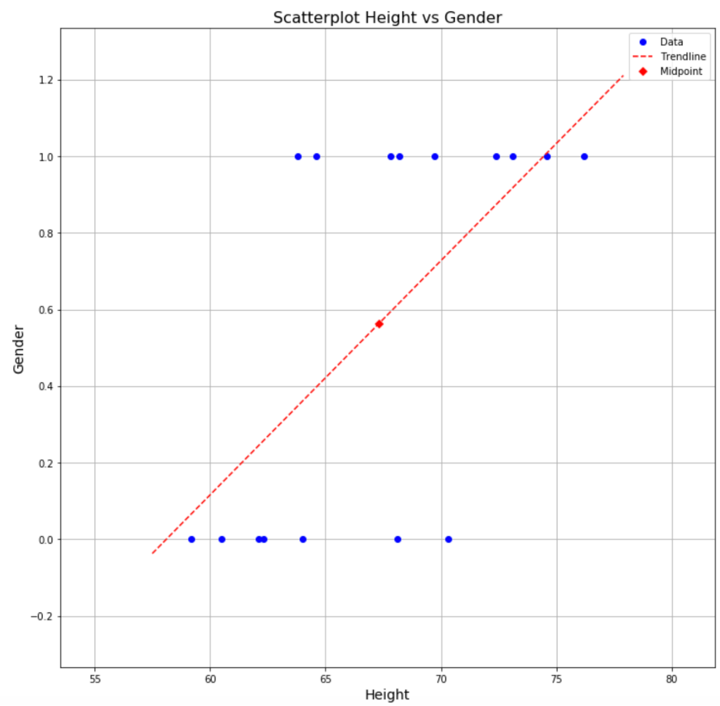
```
Heights: [59.2, 60.5, 62.1, 62.3, 63.8, 64.0, 64.6, 67.8, 68.1, 68.2, 69.7, 70.3, 72.4, 73.1, 74.6, 76.2]
Gender: [0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1]
```



Introduction to Deep Learning

Logistic Regression: Motivating Example

If we plug this into the linear regression algorithm, we get the following:




```
rho:   0.6187   r^2:   0.3828

Residual SS:   2.4303   Explained SS: 1.5072    Total SS:   3.9375

Regression Line: y = 0.0612 * x - 3.554
```

There are many issues with this:

How can we use this to predict someone's gender from their height?

How to give the probability of their gender?

There is clearly no linear trend, so what does the line even mean?

Introduction to Deep Learning

Logistic Regression: The Logit Transformation

In order to solve this, we will transform the scale of Y into a new domain, in this case into the real interval [0..1] used for probabilities. This is called the **Logit Transformation**, and is based on the notion of a **sigmoid function** $s : \mathcal{R} \rightarrow [0..1]$ form

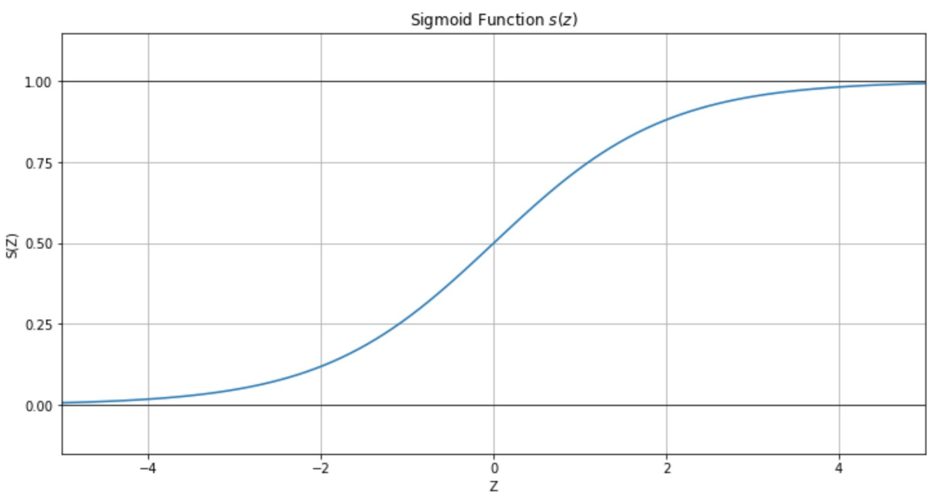
$$s(z) = \frac{e^z}{e^z + 1} = \frac{1}{1 + e^{-z}}$$

```python
def s(z):
    return 1/(1+np.exp(-z))
```

$$\lim_{z \to \infty} \frac{1}{1 + e^{-z}} = 1$$

$$\lim_{z \to -\infty} \frac{1}{1 + e^{-z}} = 0$$

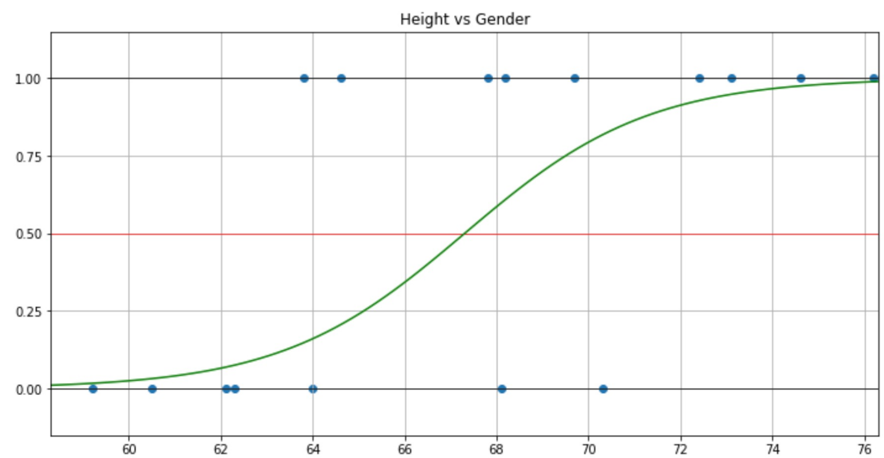
$$s(0) = \frac{1}{1 + e^0} = \frac{1}{1 + 1} = 0.5$$



Introduction to Deep Learning

Logistic Regression: The Logit Transformation

The punchline here is that we will transform the regression line into a sigmoid, and use it to give us the probability that a given individual is male, and then define as a **decision boundary** a threshold (typically 0.5) by which we will decide if the binary output class is 1 or 0:



Caveat: Such decision boundaries are typically not used in neural networks, so the output is between 0 and 1.

But in fact it is not that simple, because the **least squares technique does not work**
any more, and we will have to recast the regression framework around the sigmoid function.....

Introduction to Deep Learning

Linear Regression: What is the "cost function"?

In linear regression, we define the error of the prediction as the MSE (mean square error) of the predictions:

$$\begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_m \end{bmatrix} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_m \end{bmatrix} - \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}.$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n} e_i^2 = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

and we have explicit formulae for finding the parameters for the slope and y-intercept of the regression line which minimizes the MSE:

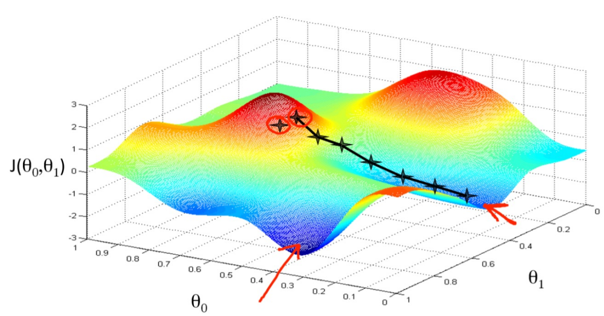
$$W = (X^T X)^{-1} X^T Y$$

But what if we didn't have such an explicit formula?

Introduction to Deep Learning

Linear Regression Concluded: Gradient Descent to find weights W

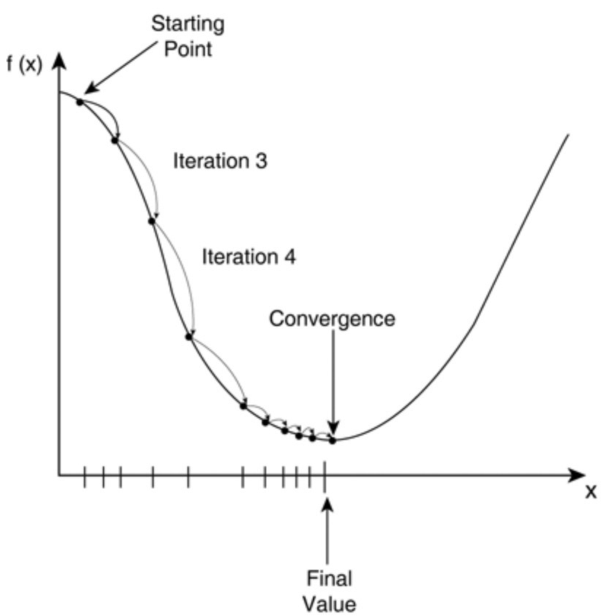
But what if we didn't? If there is no analytical solution (a formula), then we must define the error explicitly using a cost function, and then use a search algorithm called **Gradient Descent** to find the values for W which minimize this error.

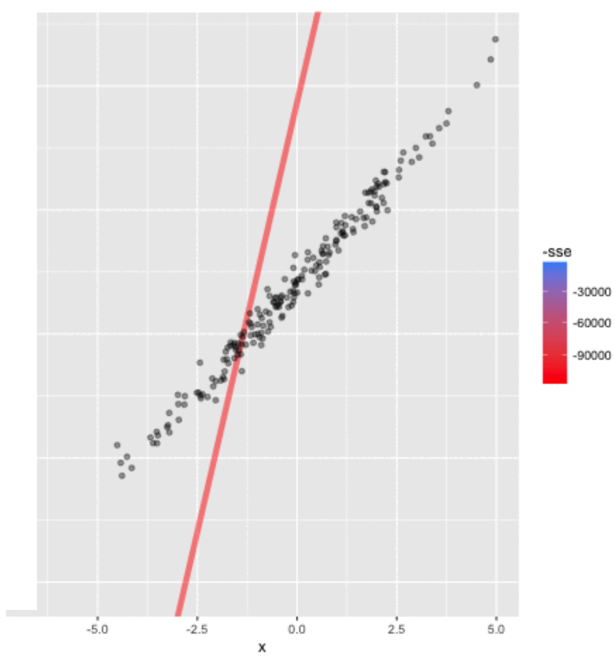


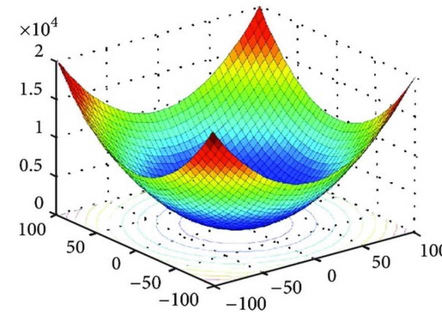
Introduction to Deep Learning

Linear Regression Concluded: Gradient Descent to find W

Gradient Descent is an iterative approximation algorithm, which "tweaks" the weights in W to move in the direction of smaller errors/lower "cost."





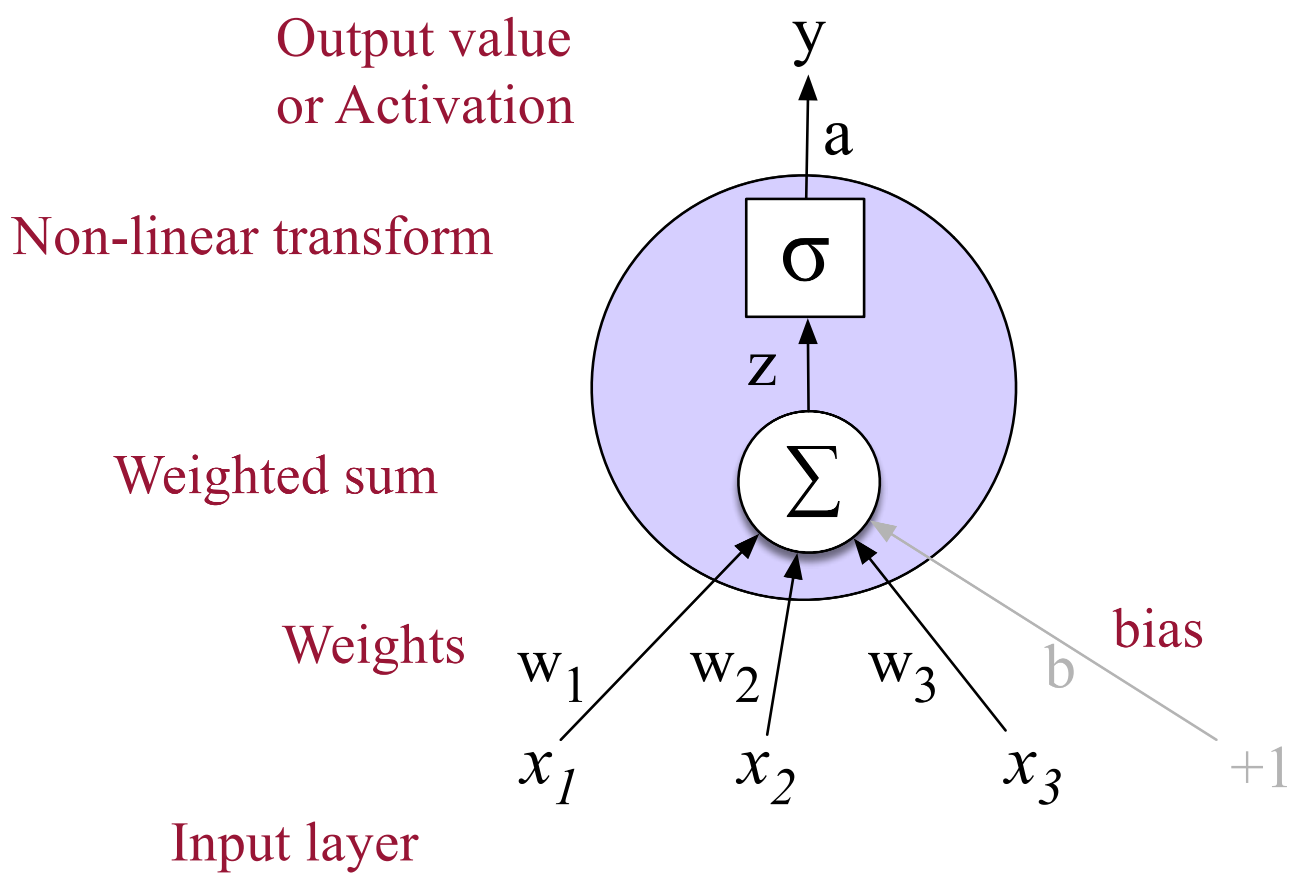


Hyperparameters:

- λ = learning rate (how far to jump!)
- termination criterion

Introduction to Deep Learning

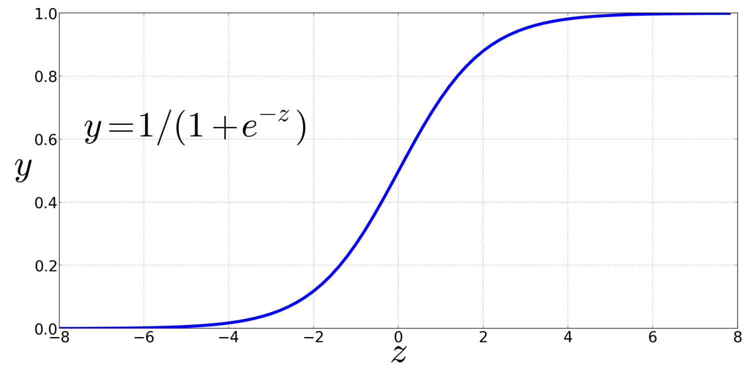
Each neuron in a neural network is implemented as a logistic regression algorithm, with an additional input called the bias (to scale the inputs).



Introduction to Deep Learning

One possible activation function f is the sigmoid which is typical in logistic regression:

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



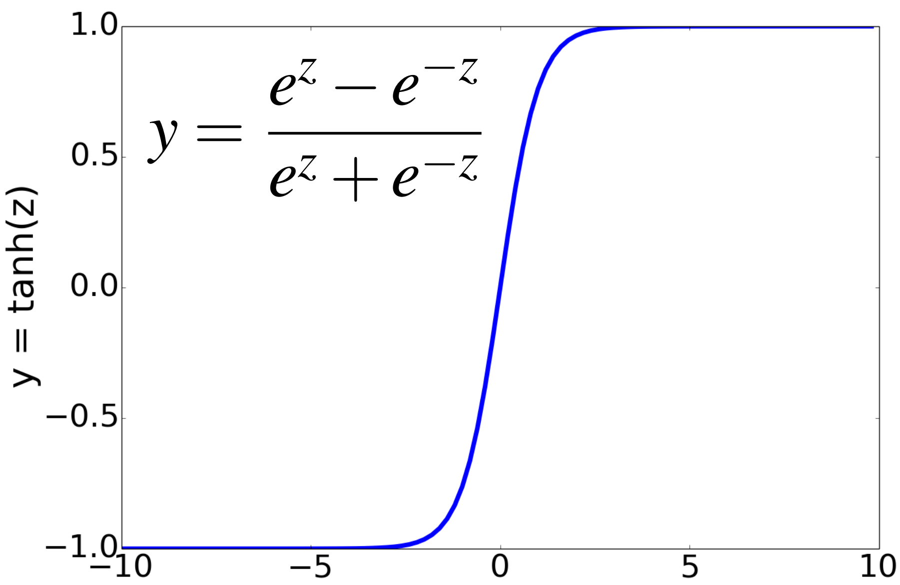
Thus:

$$y = \sigma(w \cdot x + b) = \frac{1}{1+\exp(-(w \cdot x + b))}$$

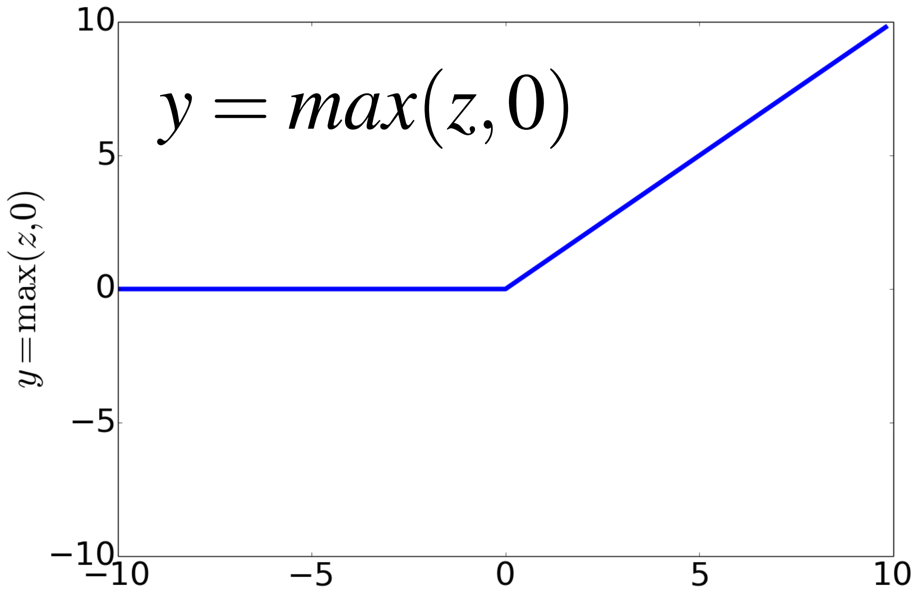
Introduction to Deep Learning

But Non-Linear Activation Functions besides sigmoid are often used!

Most Common:



tanh



ReLU
Rectified Linear Unit

Introduction to Deep Learning

- ***Suppose a unit has:***
- ***w* = [0.2,0.3,0.9]**
- ***b* = 0.5**
- **What happens with input x:**
- ***x* = [0.5,0.6,0.1]**

$$y = \sigma(w \cdot x + b) = \frac{1}{1 + e^{-(w \cdot x + b)}} =$$

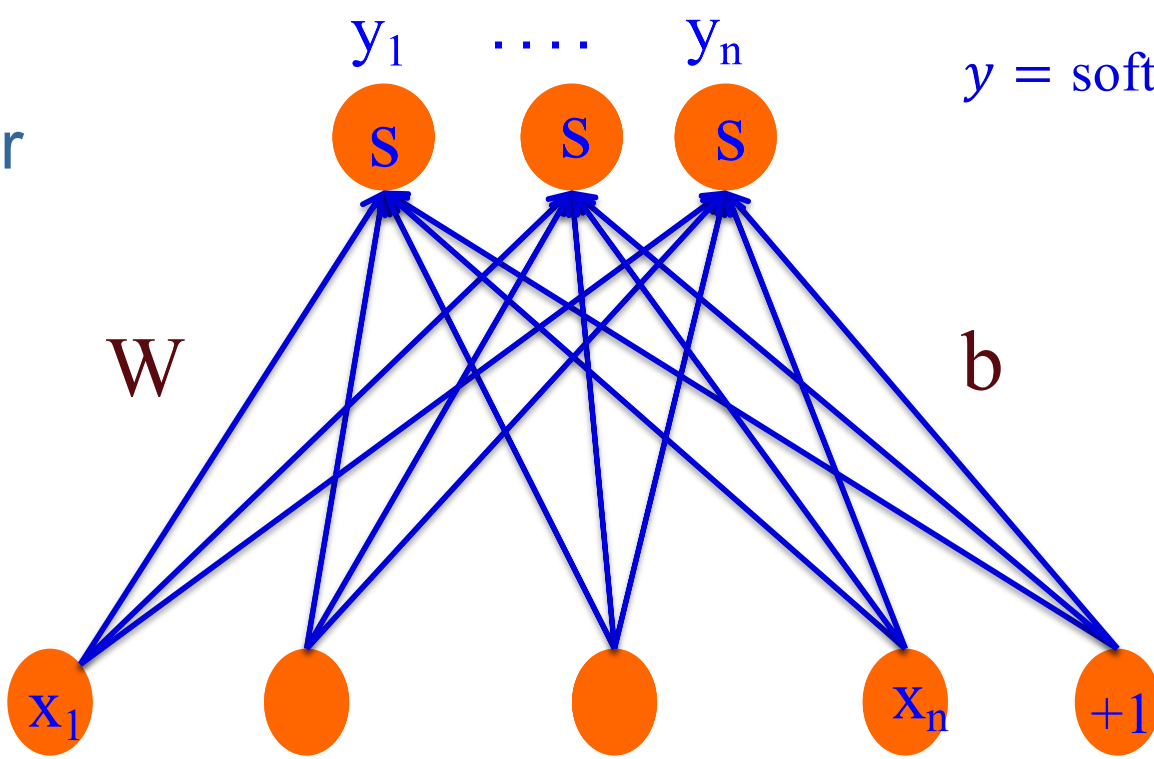
$$\frac{1}{1 + e^{-(.5 * .2 + .6 * .3 + .1 * .9 + .5)}} = \frac{1}{1 + e^{-0.87}} = .70$$

Introduction to Deep Learning

When the output is a vector, a generalization of the sigmoid function, called the softmax, is used:

$$y = \text{softmax}(Wx + b)$$

Output layer (softmax nodes)



Input layer scalars

Introduction to Deep Learning

Softmax = a generalization of sigmoid which scales k numbers into a probability distribution.

- For a vector *z* of dimensionality *k*, the softmax is:

$$\text{softmax}(z) = \left[\frac{\exp(z_1)}{\sum_{i=1}^{k} \exp(z_i)}, \frac{\exp(z_2)}{\sum_{i=1}^{k} \exp(z_i)}, \ldots, \frac{\exp(z_k)}{\sum_{i=1}^{k} \exp(z_i)} \right]$$

- Example:

$$z = [0.6, 1.1, -1.5, 1.2, 3.2, -1.1]$$

$$\text{softmax}(z) = [0.055, 0.090, 0.006, 0.099, 0.74, 0.010]$$

Text Classification: Is this spam?

Subject: **Important notice!**
From: Stanford University <newsforum@stanford.edu>
Date: October 28, 2011 12:34:16 PM PDT
To: undisclosed-recipients:;

Greats News!

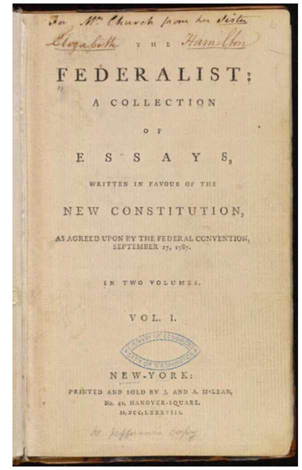
You can now access the latest news by using the link below to login to Stanford University News Forum.

http://www.123contactform.com/contact-form-StanfordNew1-236335.html

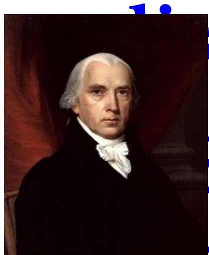
Click on the above link to login for more information about this new exciting forum. You can also copy the above link to your browser bar and login for more information
about the new services.

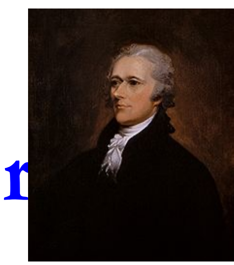
© Stanford University. All Rights Reserved.

Who wrote which Federalist papers?



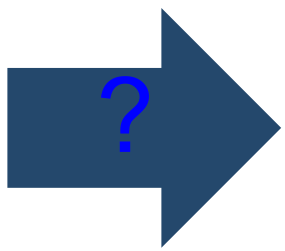
- 1787-8: anonymous essays try to convince New York to ratify U.S Constitution: Jay, Madison, Hamilton.
- Authorship of 12 of the letters in dispute
- 63: solved by Mosteller Wallace using Bayesian methods


James Madison


Alexander Hamilton

What is the subject of this medical article?

MEDLINE Article



MeSH Subject Category Hierar

- Antogonists and Inhibitors
- Blood Supply
- Chemistry
- Drug Therapy
- Embryology
- Epidemiology
- …

Positive or negative movie review?

- \+ *...zany characters and richly applied satire, and some great plot twists*
- − *It was pathetic. The worst part about it was the boxing scenes...*
- \+ *...awesome caramel sauce and sweet toasty almonds. I love this place!*
- − *...awful pizza and ridiculously overpriced...*

Positive or negative movie review?

+ *...zany characters and richly applied satire, and some great plot twists*

− *It was pathetic. The worst part about it was the boxing scenes...*

+ *...awesome caramel sauce and sweet toasty almonds. I love this place!*

− *...awful pizza and ridiculously overpriced...*

Text Classification: Definition

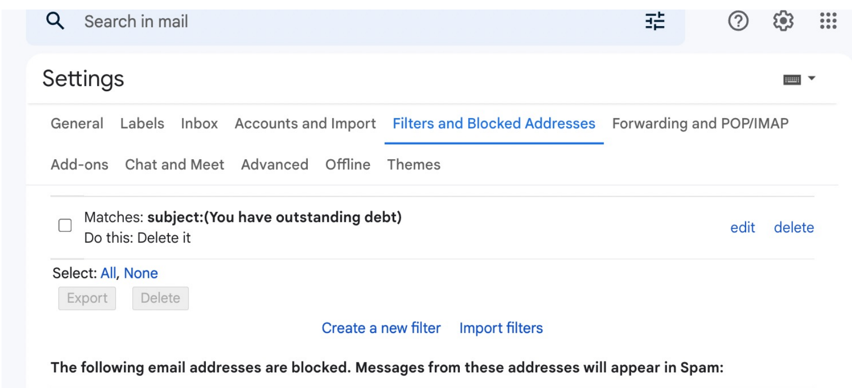
- ***Input***:
  - a document d
  - a fixed set of labels/classes $C = \{c_1, c_2,\ldots, c_J\}$

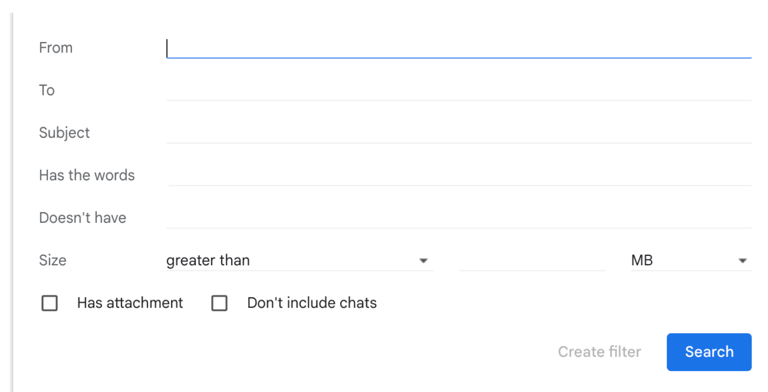
- ***Output*: a predicted class $c \in C$**

Caveats: In general, an algorithm will return probabilities for all document classes: this can be used to find the single best class, or—by setting a threshold or a bound on the number of classes—a set of classes.

Classification Methods: Hand-coded rules

- **Rules based on combinations of words or other features**
  - spam: black-list-address OR ("dollars" AND "you have been selected")
- **Accuracy can be high**
  - If rules carefully refined by expert
- **But building and maintaining these rules is expensive**





Classification Methods: Supervised ML

- ***Input:***
  - a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$
  - a randomly-permuted set of labeled documents $(d_1,c_1),\ldots,(d_n,c_n)$ split into
    - a training set $(d_1,c_1),\ldots,(d_m,c)$
    - a testing set $d_{m+1},\ldots,d_n$ (labels withheld)
- ***Output:***
  - A classifier $\gamma : d \rightarrow c$ trained the training set
  - The testing set with labels calculated by γ
  - Test results (confusion matrix, metrics, etc.)

Classification Methods: Supervised ML

- **There are many different kinds of classifiers for labeled data**
  - Naïve Bayes
  - Logistic regression
  - Neural networks