Intro to Deep Learning and Logistic Regression

CMSC 389A: Lecture 1

Sujith Vishwajith

University of Maryland

Table of Contents

Course Logistics

Course Information

When: Fridays 12:00 - 12:50 p.m.

Location: CSIC 2118

Office Hours: Fridays 1:00-2:00 p.m. or by appointment

Instructor: Sujith Vishwajith

Contact Info: Piazza or svishwaj@terpmail.umd.edu

Advisor: Dr. Jordan Boyd-Graber

Website: https://umd-cs-stics.gitbooks.io/cmsc389a-practical-deep-learning/

Textbooks

Required: *None*

Recommended

- ***Deep Learning*** by Ian Goodfellow, Yoshua Bengio, & Aaron Courville
- ***Deep Learning with Python*** by Francois Chollet

Grading

$$\text{Grade} = 0.4P + 0.2M + 0.4F$$

P = Practicals

M = Midterm

F = Final Project

Practical Schedule

Practical 1: Logistic Regression & Perceptron Algorithm

Practical 2: Feed-Forward Neural Networks

Practical 3: Self-Driving Car

Practical 4: Text Classification

Practical 5: Image Captioning

Practicals

Aimed to help develop a practical understanding of deep learning models and theory.

Practicals may be submitted up to 24 hours late for a 10% penalty. If you submit both on-time & late, your project will receive the maximum of the penalty-adjusted scores. You may submit multiple times.

Midterm

A straightforward multiple choice and free response question exam based on all the course lectures and projects till that point.

Focuses on concepts and understanding rather than specifics.

Final

Final project to demonstrate mastery of all topics learned and apply knowledge to create a new application from scratch.

Submit the project code, write up explaining the code/approach, and a 5 minute video going over the project.

Team members (groups of up to 3) and project proposal due by 4/13.

Goal

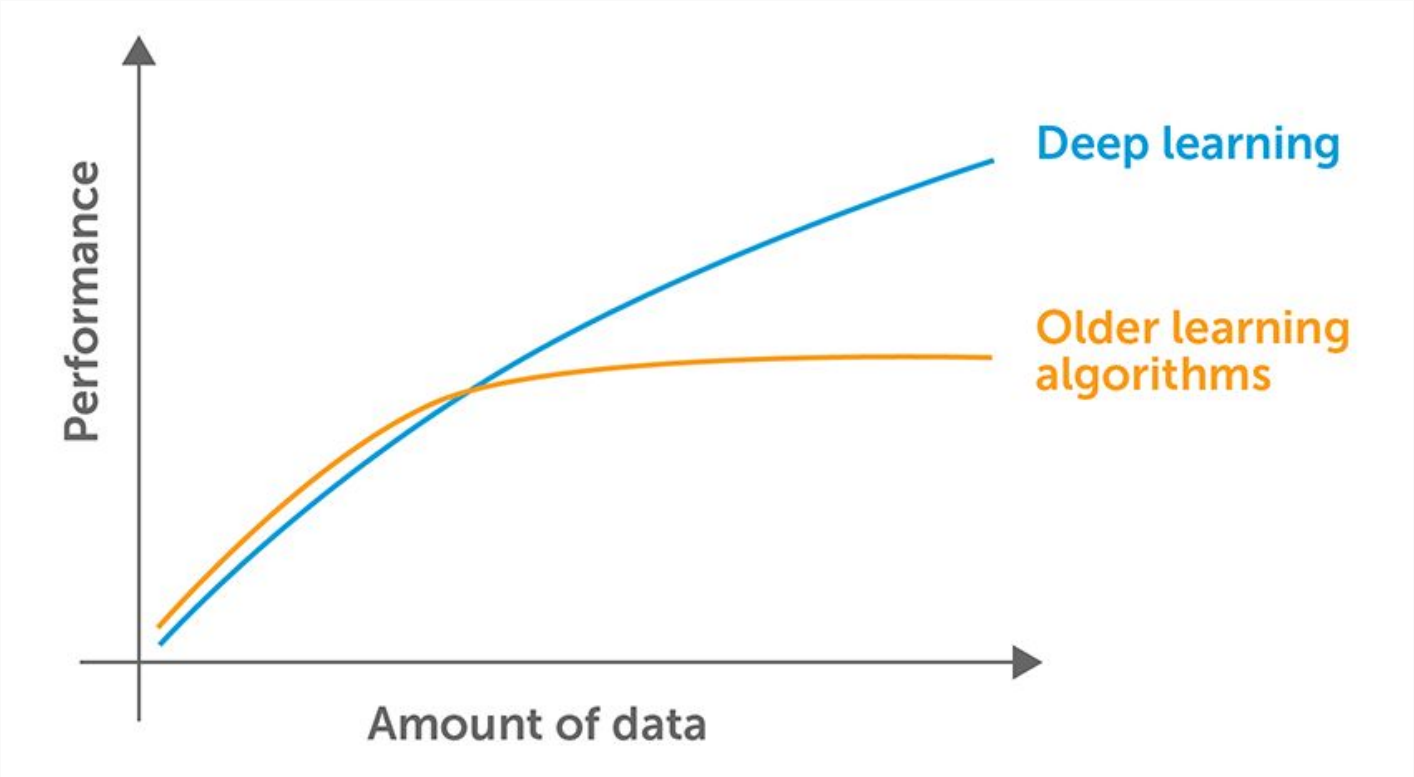
Get practical hands-on experience with Deep Learning and be able to design and develop models to solve a variety of tasks.

What you will learn:

- How Neural Networks work
- Building and training custom models
- Convolutional Neural Networks
- Recurrent Neural Networks
- Tuning and debugging Neural Networks
- How to utilize pre-trained models
- Python libraries such as Keras, PyTorch, Pandas, etc.

Why Deep Learning?

Performance Scales with Data



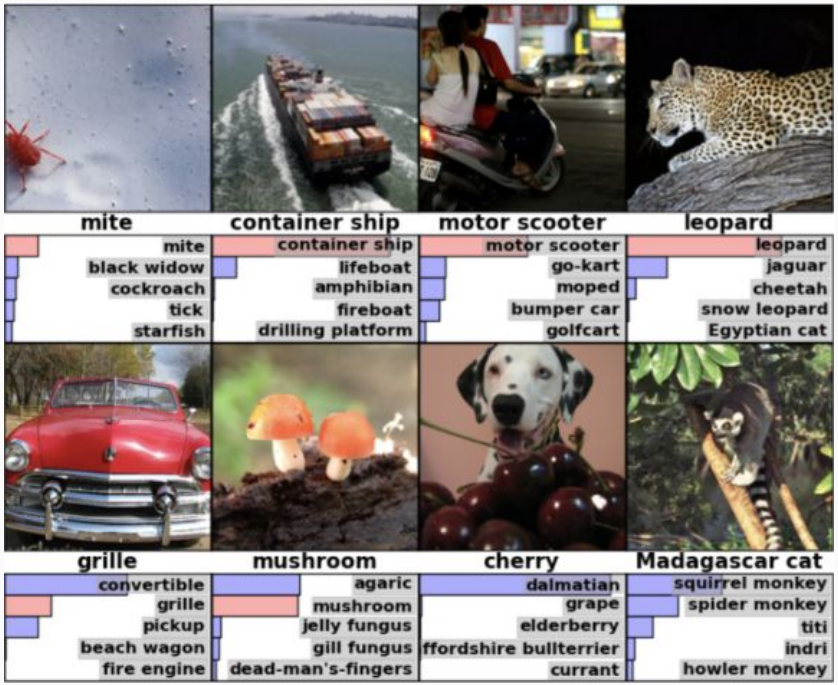
Diverse Applications

Deep learning has achieved major success in various areas including image recognition, video games, speech recognition, natural language, robotics, art, etc.

Why is this amazing? Because these tasks are all easy for humans but are incredibly hard for computers to model.

Deep Learning for Images

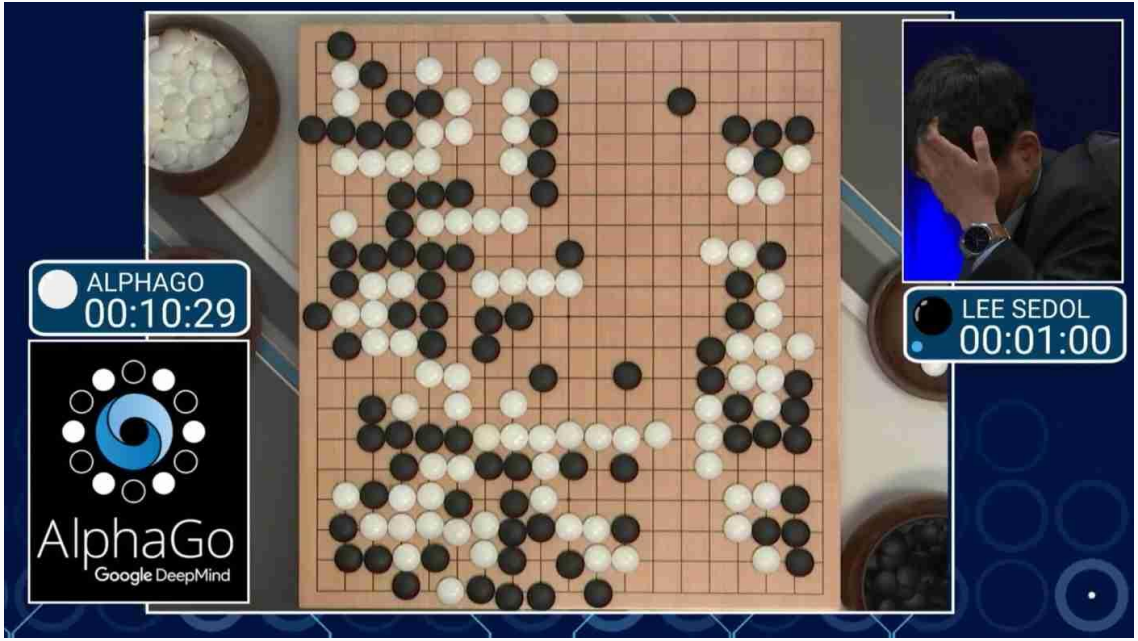
1. **Images**
2. Games
3. Language
4. Robotics
5. Art



ImageNet Classification Challenge

Deep Learning for Games

1. Images
2. **Games**
3. Language
4. Robotics
5. Art



DeepMind's AlphaGo which beat world champion Lee Sedol

Deep Learning for Language

1. Images
2. Games
3. **Language**
4. Robotics
5. Art

Google Translate and Speech-Recognition uses Deep Learning

Deep Learning for Robotics

1. Images
2. Games
3. Language
4. **Robotics**
5. Art

Uber's Self-Driving Car

Deep Learning for Art

1. Images
2. Games
3. Language
4. Robotics
5. **Art**

Neural Style-Transfer for Imitating Painter Styles

History of Deep Learning

Timeline

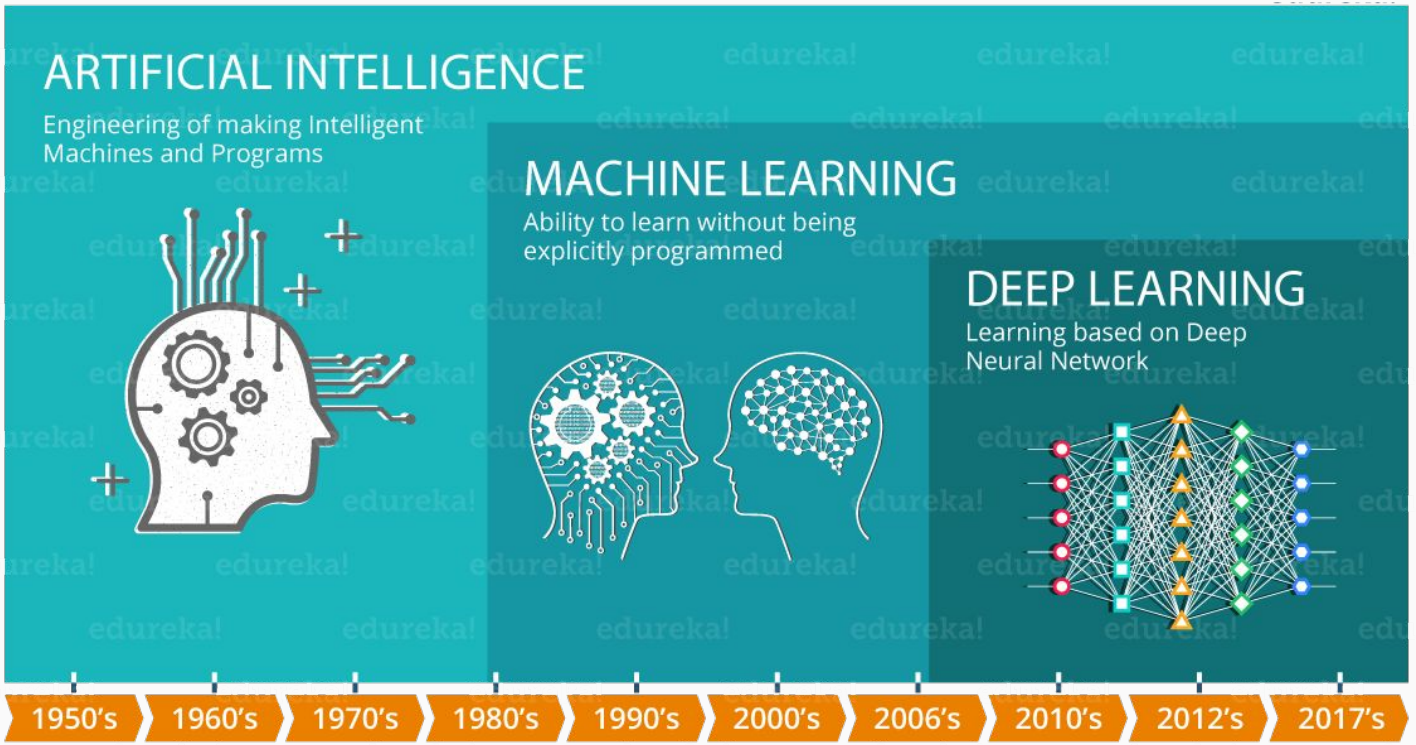
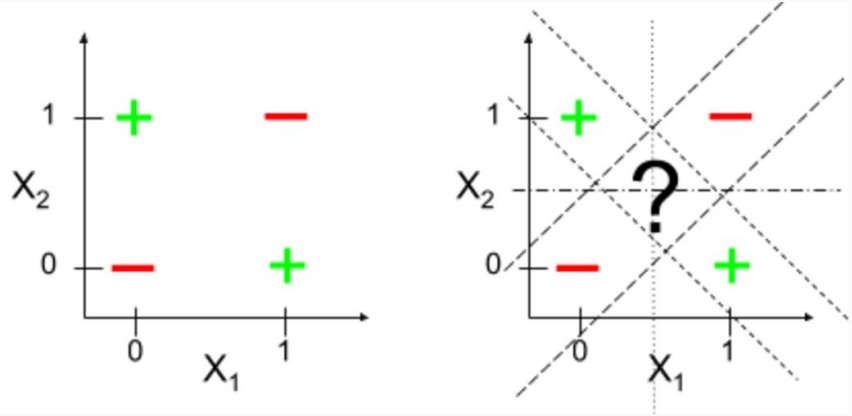


Image from Edureka

1957 - Perceptron

Frank Rosenblatt develops the **Perceptron** when proposing a machine for binary classification. It was the precursor to neural networks.

A major problem however was that the XOR problem couldn't be solved by perceptrons as proven by Minsky in his book ***Perceptrons***.

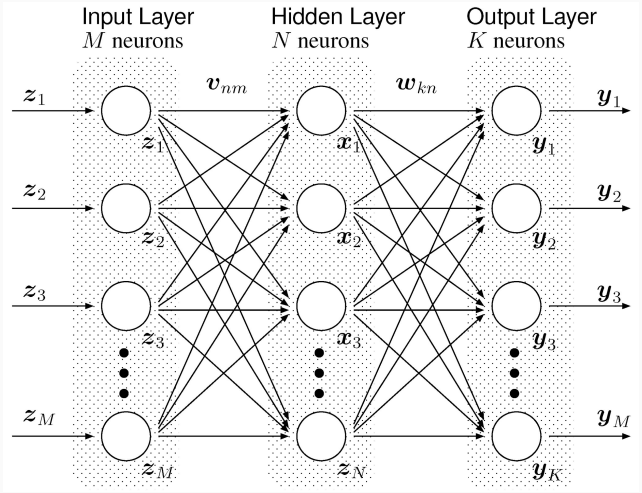


Can't separate + and - symbols with only one line

1969 - Multi-layer Perceptron

Minsky and Papert published a famous book ***Perceptrons: An Introduction to Computational Geometry*** talking about multi-layer perceptrons which could solve more complex tasks.

Unfortunately, no one knew how to train them.



Multi-layer perceptron

AI Winters

The years **1974-1980** and **1987-1993**, were known as the ‘AI winters’.

Due to lack of major progress, AI received a severe budget cut in funding and as a result, deep learning research was mostly abandoned.

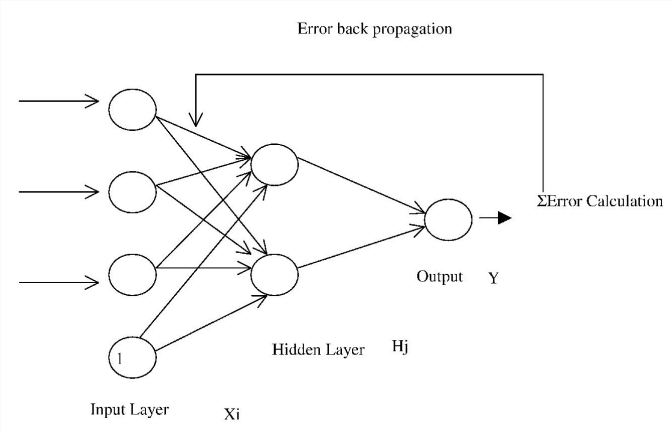
Lack of computational resources was also a huge cause of the winters.

1986 - Backpropagation

Hinton, Rumelhart, and Williams showed that backpropagation could be used to train multi-layer perceptrons.

Allowed for solving XOR and other complicated functions.

Still how we train neural networks today.



Backpropagation Intuition

1989 - Machines read

Yann LeCun and his research team utilized convolutional neural networks to read handwritten digits famously known as MNIST.

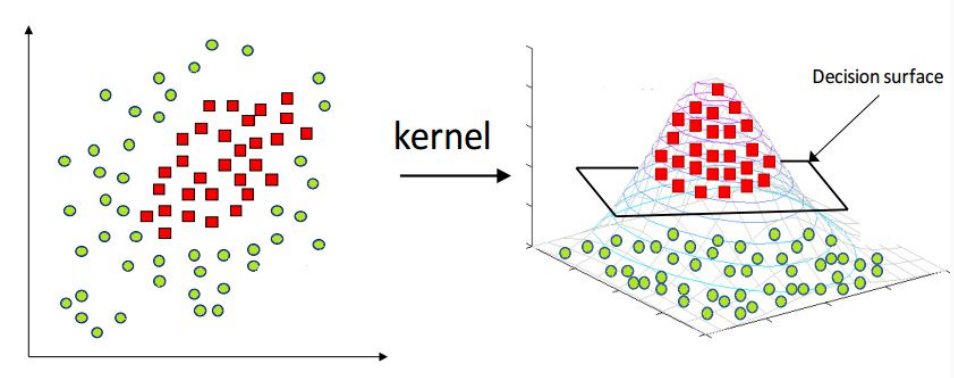
The system was used later on used by companies to read millions of checks and zip codes.

MNIST Handwritten Digits

1995 - Support Vector Machines

SVM's have existed since the 1960's but Cortes and Vapnik presented the standard model used today.

Widespread success and use in tasks from text categorization to image classification.

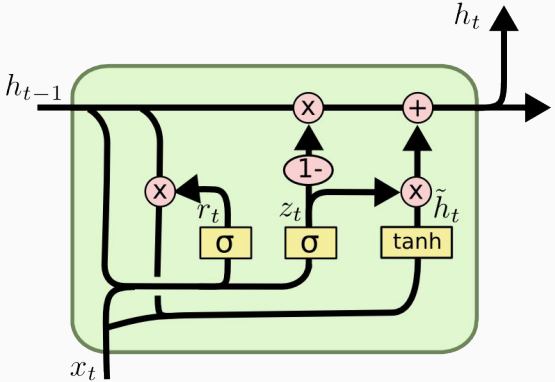


SVM Kernel Transformations

1997 - Long Short-Term Memory

Schmidhuber and Hochreiter proposed a type of recurrent neural network (RNN) titled Long short-term memory (LSTMs).

Solved the long-term dependency problem of RNN's. Still used today especially for natural language problems.



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

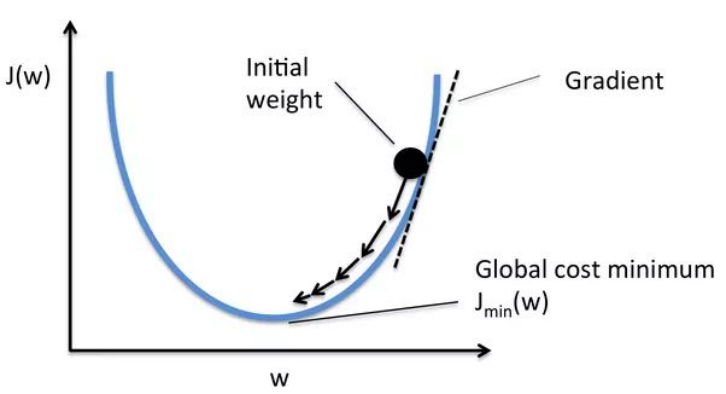
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Cell in LSTM Model Architecture

1998 - Gradient Based Learning

Yann LeCun and a group of researched published the paper ***Gradient-Based Learning Applied to Document Recognition*** which introduced gradient based learning.

Talked about the stochastic gradient descent algorithm (SGD) in combination with back-propagation to train neural networks.

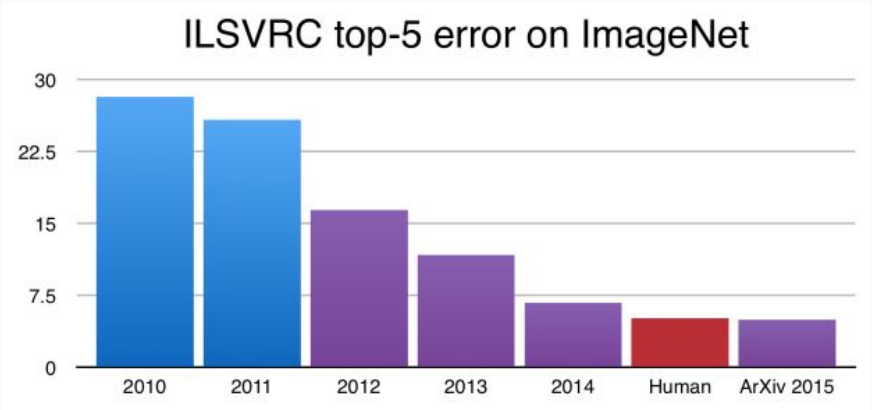


Gradient Descent Intuition

2012 - AlexNet

Alex Krizhevsky, Geoffrey Hinton, and Ilya Sutskever used Convolutional Neural Networks (CNNs) to compete in the ImageNet challenge and outperformed the second best model by over 10%.

Sparked a huge interest in CNNs for Image related tasks. Since then new models have been proposed like Microsoft's ResNet which achieved superhuman accuracy on ImageNet.



Present

Over the last 5-10 years, Deep Learning has become incredibly mainstream.

Upended a lot of traditional machine learning techniques for tasks.

Interest in Deep Learning research conferences such as NIPS have more than tripled.

Even Kristen Stewart has published a Deep Learning paper.

So why now?

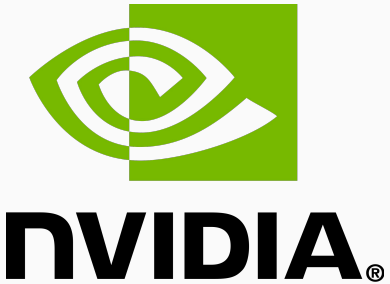
Why now

Computing power has finally caught up.

Increase in data collection and labeling.

New deep learning platforms and frameworks.

Research Advancements (new optimizers, batch normalization, dropout, ReLU)



Logistic Regression

Logistic Regression Overview

Developed by statistician David Cox in 1958.

Very similar to linear regression but used for classification.

Tries to find a linear separator between classes.

Models the probability that an example belongs to a certain class.

$$P(Y = 1 \mid data)$$

Predicted probability is used to classify the example.

Utilizes the logistic (sigmoid) function.

Example

Given information about a patient can we decide if they have lung cancer?

Given set of features ***x*** such as age, gender, weight, tumor mass, etc.

Want to predict a probability that either ***Y=1*** they do, or ***Y=0***, they don't.

Basics

Features are represented as $[x_0, x_1, \dots, x_{|F|}]$ where $|X|$ = ***# of Features***.

Weights represented as a list $[w_0, w_1, \dots, w_{|B|}]$ where $|B| = |X|$.

Additional parameter ***b*** is the bias.

Our weights and bias are estimated from our training data.

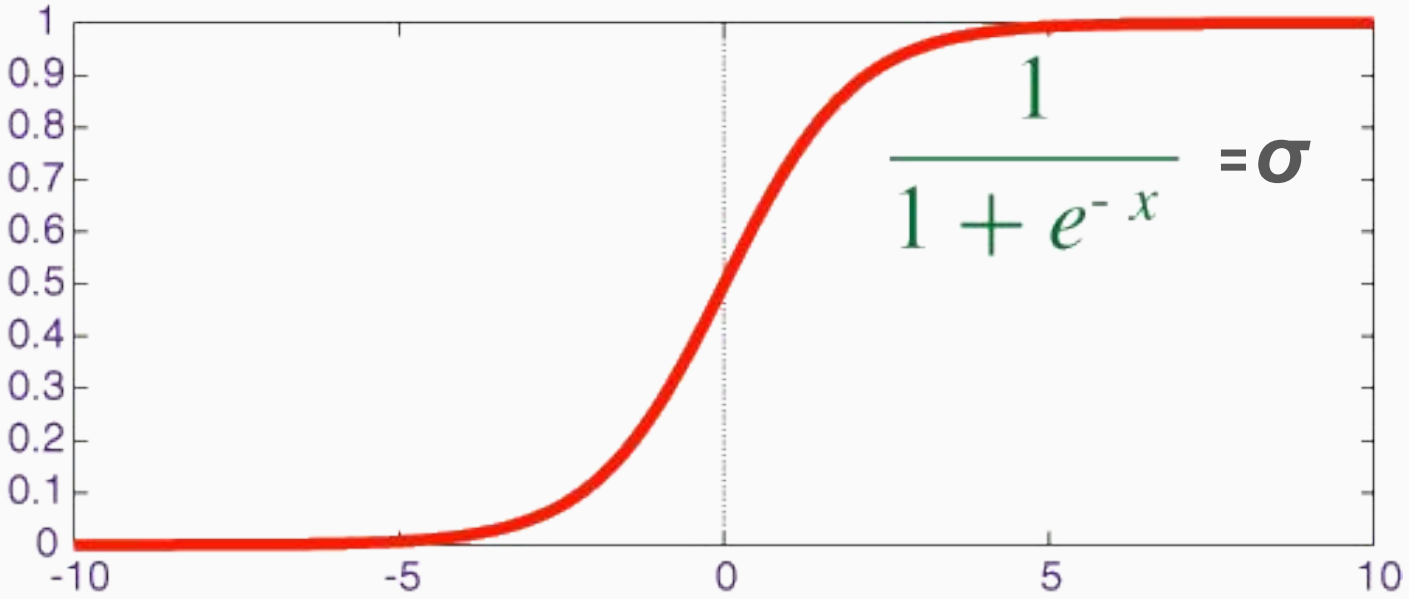
Ideally, the learned weights for the model predict a value near 1 if it belongs to the class and 0 if it doesn't.

Therefore probability of belonging to default class is $P(Y=1 \mid x) = \sigma(w^T x + b)$.

Sigmoid Function

S-shaped curve that maps real numbers to a value between 0 and 1.

The output is the probability. If output is > 0.5, it belongs to the class.



Making Predictions

Let's say we have two people **A** and **B**. We want to predict if they are female (1) or male (0).

A is 5.5 ft tall and weighs 125 lbs. **B** on the other hand is 6 ft tall and weighs 180 lbs.

Our parameters are $w_0 = 12.29$ (coef. of height), $w_1 = -0.43$ (coef. of weight), and b.

Based on our model:

$$P(Y=1 \mid A) = \sigma(a_0 * w_0 + a_1 * w_1 + b) = \sigma(5.5*12.29 + 125*-0.43 + b) = \sigma(14.3) = 0.9 >= 0.5$$

$$P(Y=1 \mid B) = \sigma(b_0 * w_0 + b_1 * w_1 + b) = \sigma(6.0*12.29 + 180*-0.43 + b) = \sigma(-3) = 0.2 < 0.5$$

Our model predicts **A** is female, and **B** is male.

* This is a made up example.

Training the model

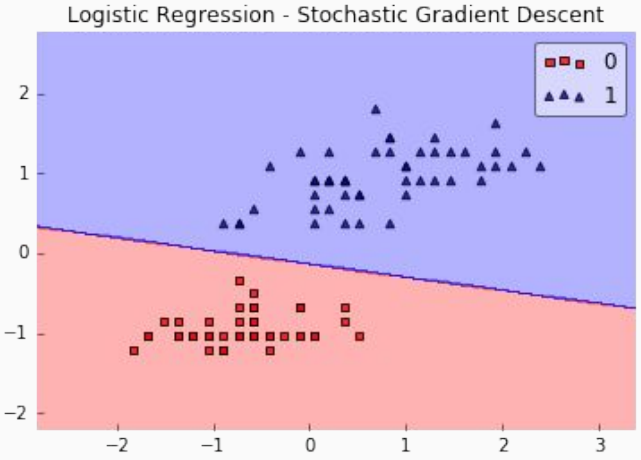
Update using stochastic gradient descent (SGD).

True label = $\bar{y}$, Features = x, Bias = b, Weights = w

Prediction = $\hat{y} = \sigma(w^T x + b)$

Error $e = \bar{y} - \hat{y}$

Updates bias: $b \leftarrow b + \alpha * e * \hat{y} * (1 - \hat{y})$



Update weights: For every coefficient, set $w_i \leftarrow w_i + \alpha * e * \hat{y} * (1 - \hat{y}) * x_i$

Where α is the learning rate.

Repeat for every training example.

Deriving the Derivative

Let's denote the sigmoid function as $\sigma(x) = \dfrac{1}{1+e^{-x}}$.

The derivative of the sigmoid is $\dfrac{d}{dx}\sigma(x) = \sigma(x)(1-\sigma(x))$.

Here's a detailed derivation:

$$
\begin{aligned}
\frac{d}{dx}\sigma(x) &= \frac{d}{dx}\left[\frac{1}{1+e^{-x}}\right] \\
&= \frac{d}{dx}\left(1+\mathrm{e}^{-x}\right)^{-1} \\
&= -(1+e^{-x})^{-2}(-e^{-x}) \\
&= \frac{e^{-x}}{(1+e^{-x})^2} \\
&= \frac{1}{1+e^{-x}} \cdot \frac{e^{-x}}{1+e^{-x}} \\
&= \frac{1}{1+e^{-x}} \cdot \frac{(1+e^{-x})-1}{1+e^{-x}} \\
&= \frac{1}{1+e^{-x}} \cdot \left(\frac{1+e^{-x}}{1+e^{-x}} - \frac{1}{1+e^{-x}}\right) \\
&= \frac{1}{1+e^{-x}} \cdot \left(1 - \frac{1}{1+e^{-x}}\right) \\
&= \sigma(x) \cdot (1-\sigma(x))
\end{aligned}
$$

* Taken from Math StackExchange here

Announcements

Announcements

Join Piazza for class questions and discussions.

Complete weekly feedback and questionnaire.

Check out the ***pre_class.md*** file for each week.

Practical 1 is posted on the course website due **February 16th** at **11:59 p.m**.

Questions?