CMPE 258, Deep Learning

Introduction to Deep Learning, Linear regression

Jan 25, 2018

DMH 149A

Taehee Jeong

Ph.D., Data Scientist

SJSU SAN JOSÉ STATE UNIVERSITY

Survey

What is your experience or background for machine learning/deep learning?

What is your motivation to take deep learning class?

What do you expect after taking deep learning class?

tax·on·o·my

Artificial Intelligence

Machine Learning

Deep Neural Network (or Deep Learning)

Machine Learning

Machine Learning is the science (and art) of programming computers so they can *learn from data*

—Aurelien Geron, 2017

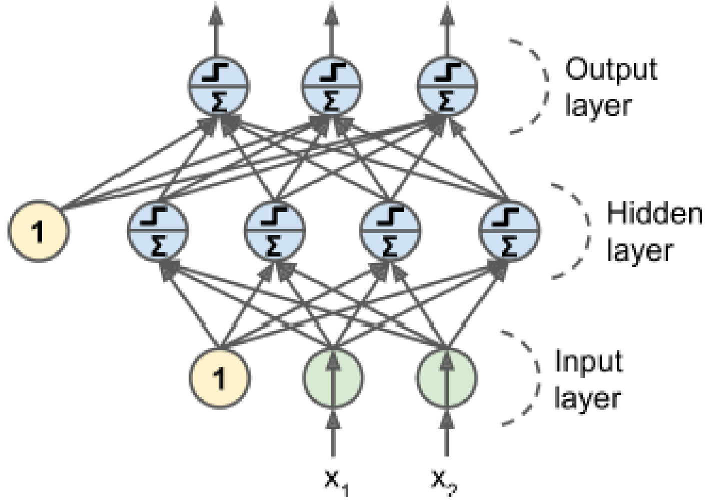
[Machine Learning is the] field of study that gives computers the ability to learn without being explicitly programmed.

—Arthur Samuel, *1959*

<Hands_On_ML>

Deep Neural Network

When an artificial neural network has two or more hidden layers, it is called a deep neural network.



<Hands_On_ML>

Deep Learning

Don’t jump into deep waters too hastily.

While Deep Learning is no doubt one of the most exciting areas in Machine Learning, you should master the fundamentals first.

Moreover, most problems can be solved quite well using Machine Learning techniques.

Deep Learning is best suited for complex problems such as image recognition, speech recognition, or natural language processing.

Also, you need enough amount of data, computing power, and patience.

<Hands_On_ML>

Prerequisites

CMPE 255 or CMPE 257

Please upload in Canvas a copy of your transcript, with the prerequisite class grade highlighted.

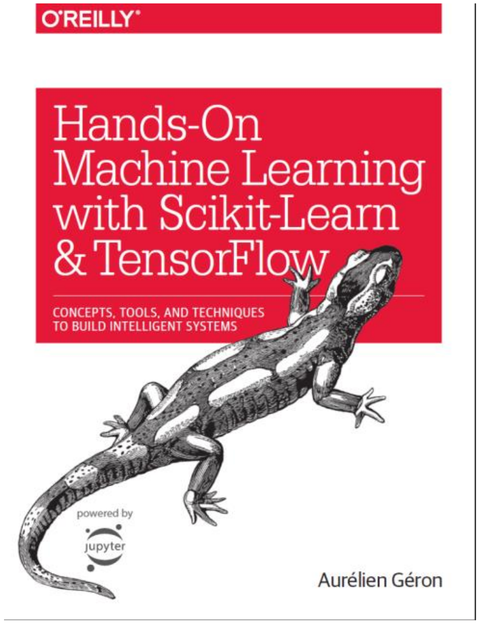
Please meet with me during office hour today if you did not complete the prerequisite classes.

Syllabus

textbook

Hands-On Machine Learning with Scikit-Learn & TensorFlow

(by Aurelien Geron)



Syllabus

Grading Information

(A+) >= 98,
(A) >= 94 and <98
(A-) >= 90 and <94
(B+) >= 85 and <90
(B) >= 75 and <85
(B-) >= 70 and <75
(C+) >= 68 and <70
(C) >= 64 and <78
(C-) >= 60 and <64
(D) >= 50 and <60
(F) < 50

Participation	10%
Homework assignments	20%
Group Project	20%
Midterm exam I	20%
Midterm exam II	20%
Final exam	20%

Grade distribution guideline for graduate courses

- 40% of the students receive A-level grades
- 50% of the students receive B-level grades
- 5-10% of the students receive C-level grades
- Up to 2% of the students received D or F-level grades

Course Schedule

Week	Date	Topics, Readings, Assignments, Deadlines
1	1/25	Introduction to Deep Learning, Linear regression
2	1/30	Cost function, Gradient Decent, Learning rate, Normal Equation
2	2/1	Overfitting, Underfitting, Training/Validating/Testing
3	2/6	Regularization, Bias-variance Trade-off
3	2/8	Logistic Regression, Binary classification
4	2/13	Softmax, Multiclass classification
4	2/15	Neural Networks , Hidden layers
5	2/20	Shallow Neural Networks, Backpropagation
5	2/22	Deep Neural Networks
6	2/27	Initialization: Xavier and He
6	3/1	Batch Normalization
7	3/6	Midterm exam1
7	3/8	Optimizers
8	3/13	Regularization
8	3/15	Convolutional Neural Networks

Course Schedule

Week	Date	Topics, Readings, Assignments, Deadlines
9	3/20	Convolutional Layer
9	3/22	Pooling Layer
10	3/27	Spring Recess
10	3/29	Spring Recess
11	4/3	CNN architectures
11	4/5	Midterm exam2
12	4/10	Recurrent Neural Networks
12	4/12	Memory cells, Input and Output Sequences
13	4/17	No class. Make up: (4/12, 5:45 – 7:00pm) Group Project proposals
13	4/19	No class. Make up: (4/24, 5:45 – 7:00pm) Group Project proposals
14	4/24	Training RNNs
14	4/26	LSTM, GRU
15	5/1	Autoencoders
15	5/3	Autoencoder application
16	5/8	Group Project Presentations
16	5/10	Group Project Presentations
Final Exam	5/16	Final Exam 2:45 pm – 5:00 pm

Group project

- Group Project proposals, 4/12 & 4/24
- Group Project Presentations, 5/8 & 5/10
- Groups of 3 students will be formed to work on a term-long group project related to deep learning.
- Each group member is expected to participate in every phase of the project. The final grade of each member will be proportional to his/her participation in the group.
- The term project will be graded on the basis of the following three components: a) project implementation, b) project report, c) project demonstration.

Prerequisites

Python Programming experience

- Pandas, Numpy, Matplotlib
- Python, Jupyter notebook
  - Pandas, Numpy
  - Load/Save data
  - Visualize data
  - Call a function from open-source library such as scikit-learn
  - Make a function
  - Loop
  - Matrix calculation

Prerequisites

College-level math

Calculus, linear algebra, probabilities, statistics

- Linear Algebra
  - Vectors
  - Matrices
  - Matrix multiply
  - Derivatives
  - …
- Probability
  - Normalization

Prerequisites

Machine Learning

- Linear Regression
  - Cost (or Loss) function
  - Gradient
  - Gradient descent
- Logistic Regression
  - Sigmoid function
  - Cost function
  - Gradient
  - Gradient descent
- Classification
  - Binary
  - Multiclass
- Regularization
  - Train/ Validation/ Test data set distribution
  - Bias, Variance
  - Underfitting, Overfitting
  - L1, L2 loss function

Course outline

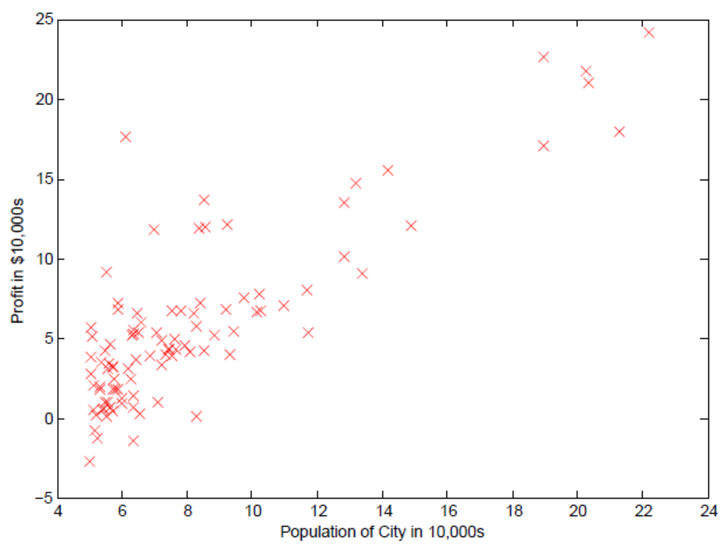
- Linear Regression, Gradient Decent, Regularization
- Logistic Regression, Classification
- Neural Network, Hidden Layers
- Deep Neural Network, Initialization, Batch Normalization
- Convolutional Neural Network, Convolution, Pooling
- Recurrent Neural Network, LSTM
- Autoencoders

Algorithm outline

Initialization (zero, random)
Activation functions
Softmax
One-hot matrix
Forward propagation / Backward propagation
Regularization
Optimization
Mini-batch gradient descent
Hyper-parameters tuning
Transfer learning

Linear regression

With One variable



- Regression: predict continuous valued output
- Supervised learning

<Machine Learning, Andrew Ng>

Linear regression

Cost function, Gradient descent

The objective of linear regression is to minimize the cost function

$$J = \frac{1}{m}\sum_{i=1}^{m}(\widehat{y^i} - y^i)^2$$

$$\hat{y} = b + Wx = W_0 + W_1 x$$

- *J : cost function*
- $\hat{y}$: *predicted target value*
- y: *actual target value*
- *x: input value*
- *b: bias*
- *W: weight*

Linear regression

(Batch) Gradient Descent

$$W_j := W_j - \alpha \frac{\partial J}{\partial W}$$

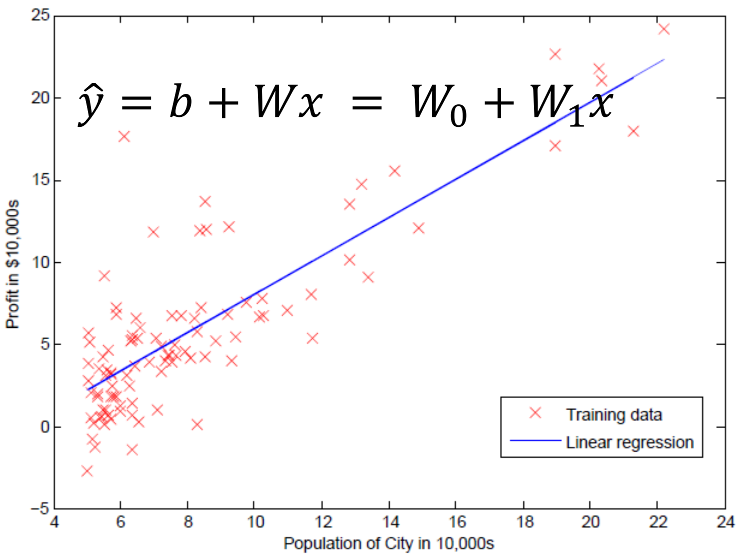
- α: *learning rate*
- $\frac{\partial J}{\partial W}$: gradient

$$\frac{\partial J}{\partial W_0} = \frac{2}{m} \sum_{i=1}^{m} \left(\widehat{y^i} - y^i \right)$$

$$\frac{\partial J}{\partial W_1} = \frac{2}{m} \sum_{i=1}^{m} \left(\widehat{y^i} - y^i \right) x$$

Linear regression

Linear regression fit



<Machine Learning, Andrew Ng>

- Weights
  $W_0, W_1, \ldots$

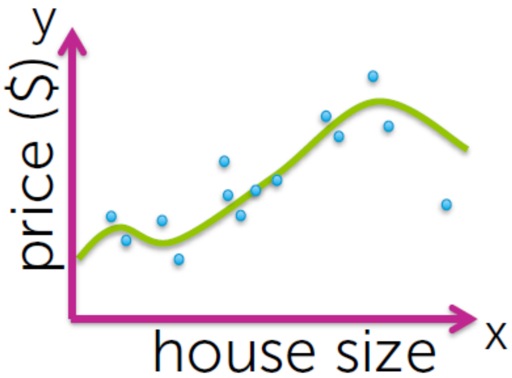
- Performance measure
  - Cost

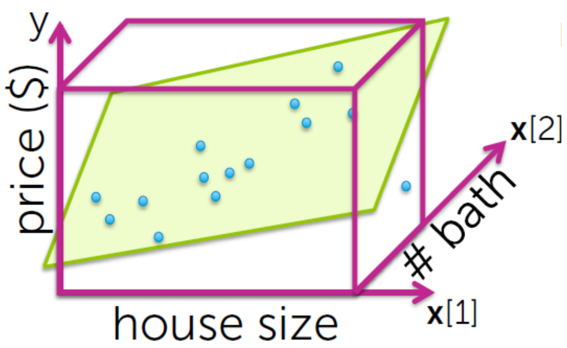
$$J = \frac{1}{m}\sum_{i=1}^{m}(\widehat{y^i} - y^i)^2$$

  - Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(\widehat{y^i} - y^i)^2}$$

Linear regression





More complex relationships than just a line.

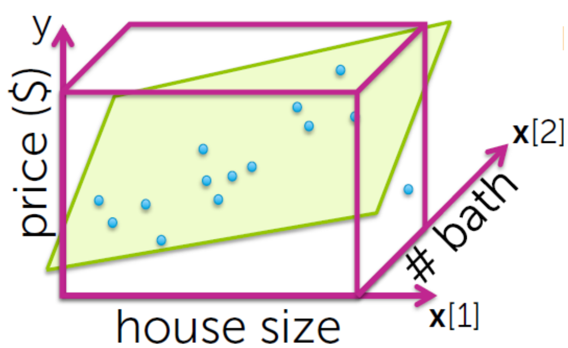
Incorporate more inputs

- Square feet
- # bathrooms
- # bedrooms
- Lot size
- Year built
- …

<Machine Learning, Emily Fox & Carlos Guestrin>

Linear regression

With multiple variables



<Machine Learning, Emily Fox & Carlos Guestrin>

$$\hat{y} = W_0 + W_1 x_1 + W_2 x_2 + \ldots$$

$$\frac{\partial J}{\partial W_0} = \frac{2}{m} \sum_{i=1}^{m} (\widehat{y^i} - y^i)$$

$$\frac{\partial J}{\partial W_1} = \frac{2}{m} \sum_{i=1}^{m} (\widehat{y^i} - y^i) x_1$$

$$\frac{\partial J}{\partial W_2} = \frac{2}{m} \sum_{i=1}^{m} (\widehat{y^i} - y^i) x_2$$

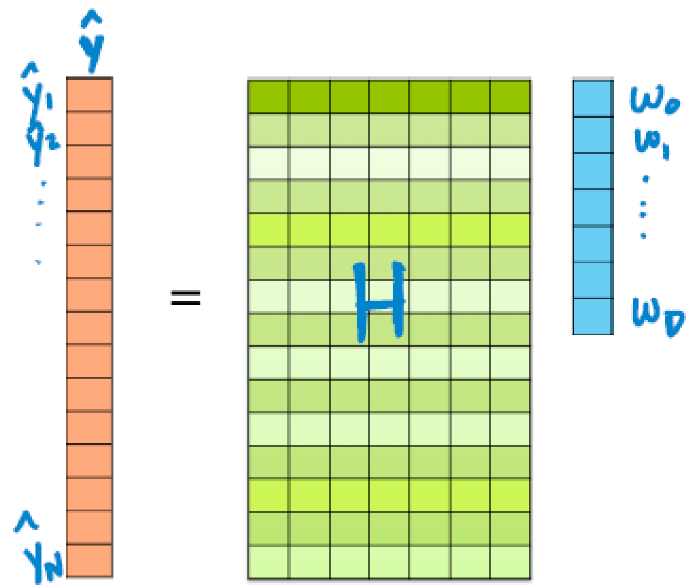
Linear regression

Matrix calculation

$$J = \frac{1}{m}\sum_{i=1}^{m}(\widehat{y^i} - y^i)^2$$

$$J = \frac{1}{m}(\hat{Y} - Y)^T(\hat{Y} - Y)$$

$$J = \frac{1}{m}(W \cdot X - Y)^T(W \cdot X - Y)$$



<Machine Learning, Emily Fox & Carlos Guestrin>

Linear regression

Normal equation

To find the value of θ that minimizes the cost function, there is a *closed-form solution*

—in other words, a mathematical equation that gives the result directly. This is called the *Normal Equation.*

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

Summary

Linear regression

- Cost function
- Gradient descent
- Matrix calculation
- Normal equation