Regression I

Fundamentals of Artificial Intelligence

Instructor: Chenhui Chu

Email: chu@i.kyoto-u.ac.jp

Teaching Assistant: Youyuan Lin

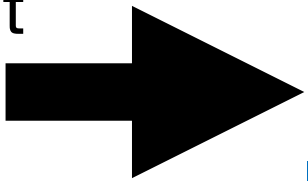
E-mail: youyuan@nlp.ist.i.kyoto-u.ac.jp

Gradient Descent Algorithm (1/4)

- This suggests some procedure for finding a minimum:
  - Start at any x (e.g., x = 0)
  - Compute f’(x)
    - If f’(x) > 0: Decrease x a bit
    - If f’(x) < 0: Increase x a bit
  - Repeat

Gradient Descent Algorithm (2/4)

- This suggests some procedure for finding a minimum:
  - Start at any x (eg. x = 0)
  - Compute f'(x)
    - If f'(x) > 0: Decrease x a bit
    - If f'(x) < 0: Increase x a bit
  - Repeat



In practice, we do this:

$$x := x - lr \cdot f'(x)$$

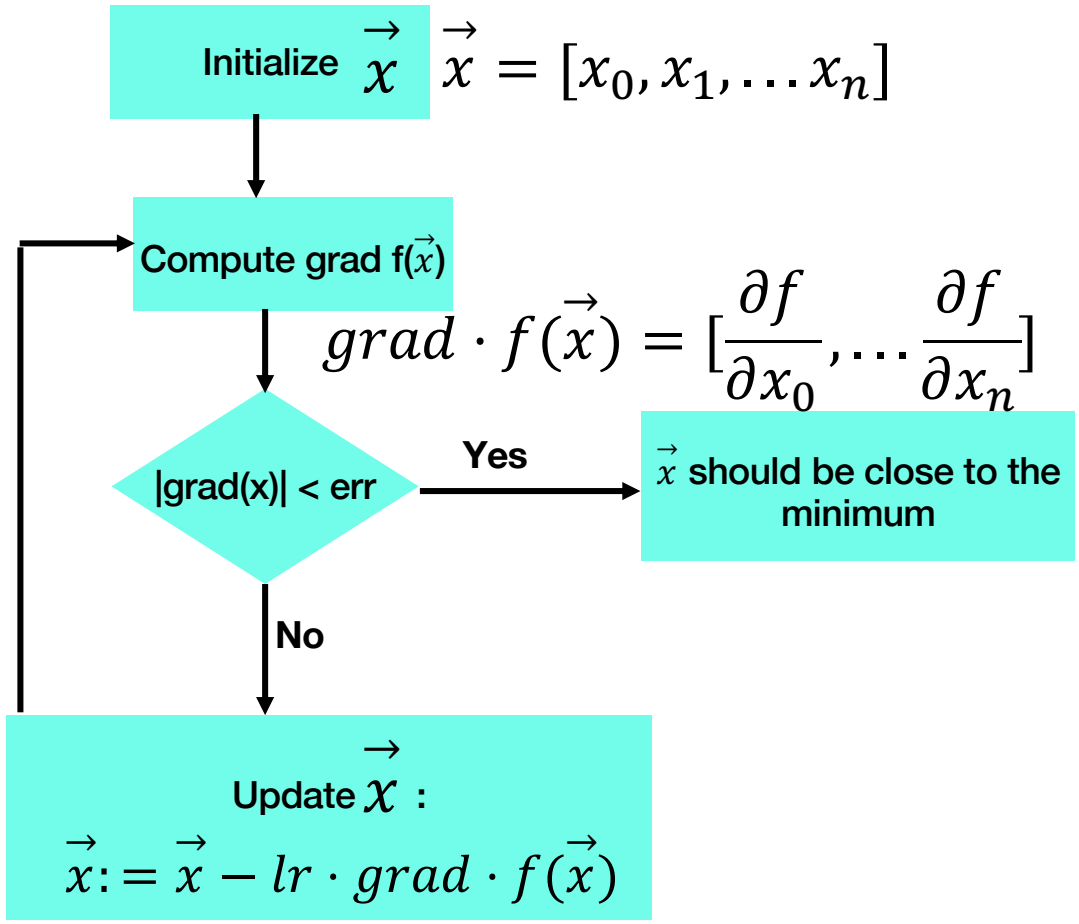
lr: learning rate

Should be a positive value

If too large: no convergence

If too small: very slow convergence

Gradient Descent Algorithm (3/4)



$$f(\vec{x}) = (x_0 - x_1)^2 + x_2^2 - x_2$$

$$grad \cdot f(\vec{x}) = [2(x_0 - x_1), 2(x_1 - x_0), 2x_2 - 1]$$

$$lr = 0.2$$

$$\vec{x} = [0,1,0]$$
$$grad \cdot f(\vec{x}) = [-2, 2, -1]$$

$$\vec{x} = [0.4,0.6,0.2]$$
$$grad \cdot f(\vec{x}) = [-0.4, 0.4, -0.6]$$

$$\vec{x} = [0.41,0.43,0.51]$$
$$grad \cdot f(\vec{x}) = [-0.04, 0.04, 0.01]$$

Gradient Descent Algorithm (4/4)

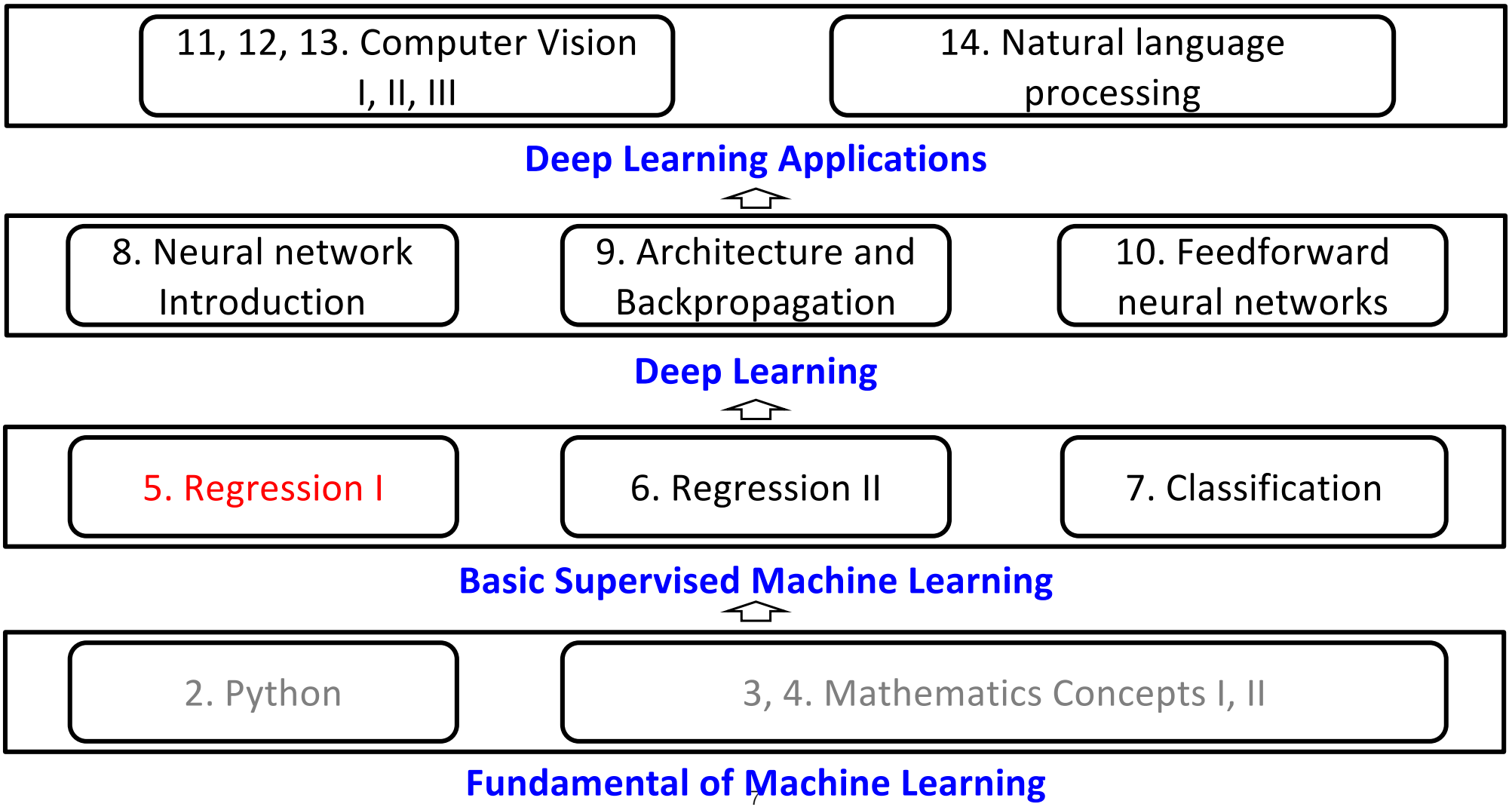
- Gradient descent works well **even** when we have functions of millions of variable
  - This is why it is so useful for Machine Learning and Neural Networks
  - Other methods will not be practical in such settings

- Convergence will depend on the choice of a good **learning rate**
  - In experiments, a good deal of time is often spent finding an optimal learning rate
  - **Too large** learning rate: **no** convergence (ie. the system learn nothing)
  - **Too small** learning rate: **slow** convergence (ie. the system takes a long time to learn)

Schedule

- 1. Overview of AI and this Course (4/14)
- 2. Introduction to Python (4/21)
- 3, 4. Mathematics Concepts I, II (4/28, 5/12)
- 5, 6. Regression I, II (5/19, 5/26)
- 7. Classification (6/2)
- 8. Introduction to Neural Networks (6/9)
- 9. Neural Networks Architecture and Backpropagation (6/16)
- 10. Fully Connected Layers (6/23)
- 11, 12, 13. Computer Vision I, II, III (6/30, 7/7, 7/14)
- 14. Natural Language Processing (7/17)

Overview of This Course

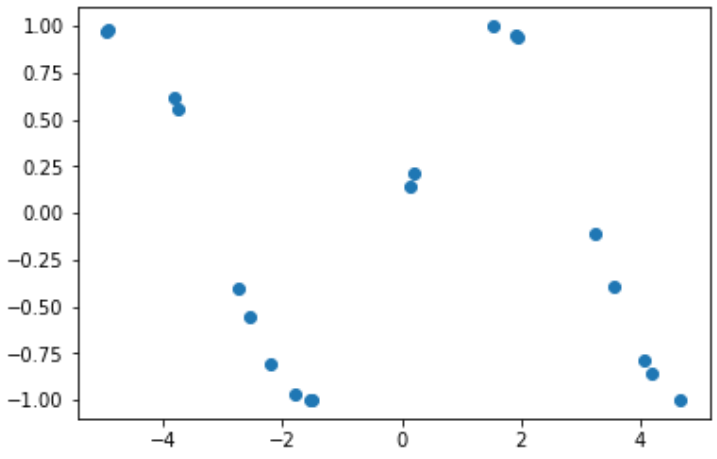


Learning a Function From Examples (1/2)

- Main idea today: Learning a function from examples

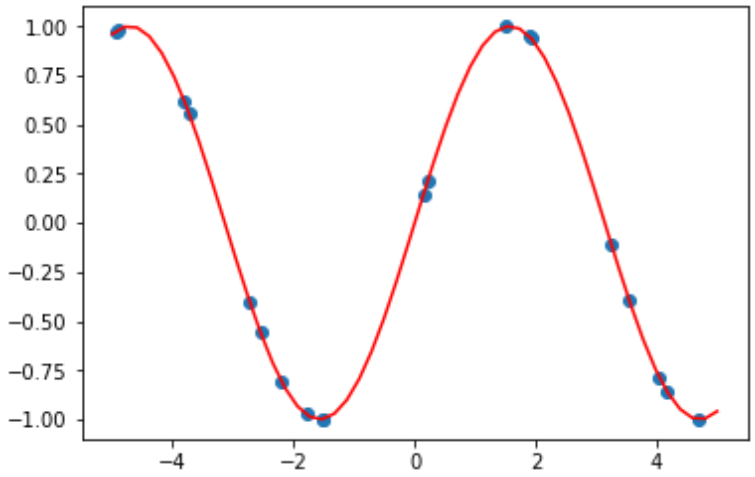
We are given example values of f(x) for some x

x	f(x)
-3.324820	0.182204
1.294681	0.962122
-4.141596	0.841473
2.928655	0.211333
-2.374387	-0.694126
4.555678	-0.987746
.....	.....



Learning

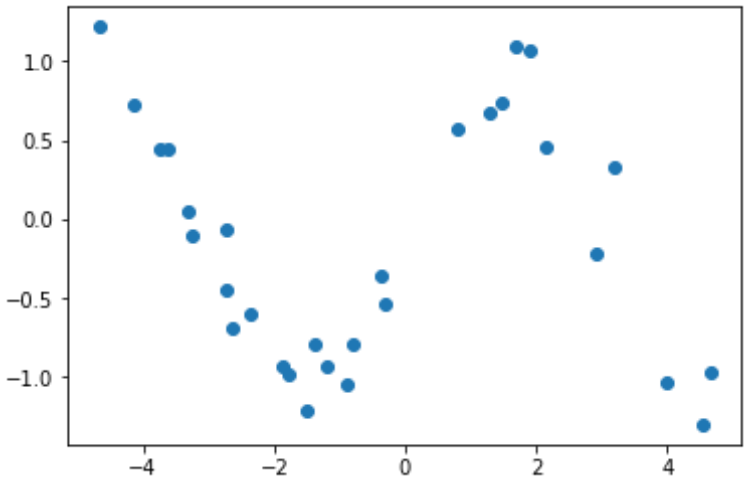
What would be the value of the function for all x?



Learning a Function from Examples (2/2)

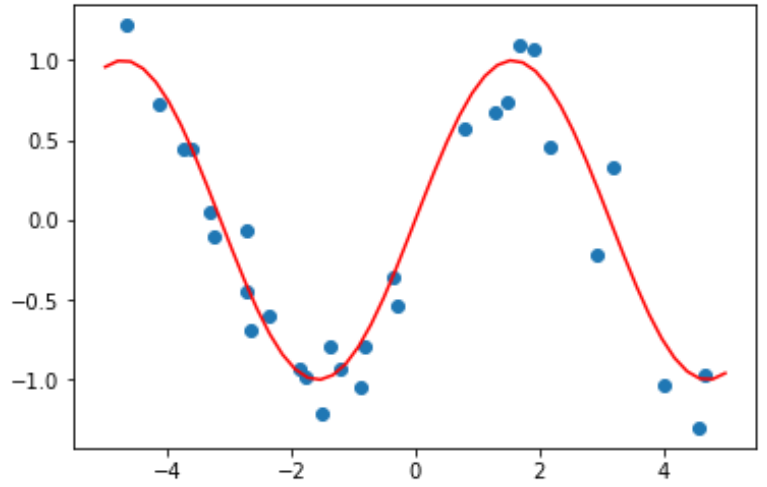
- We might not always want that the function we learn match exactly the examples. In practice, we might have to consider “imperfect” or “noisy” examples.

What we are given:



Learning

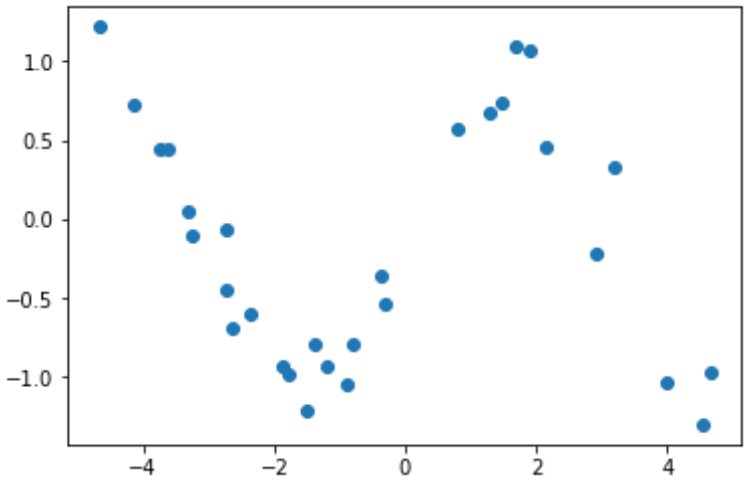
What we want to learn :



Predicting from Examples

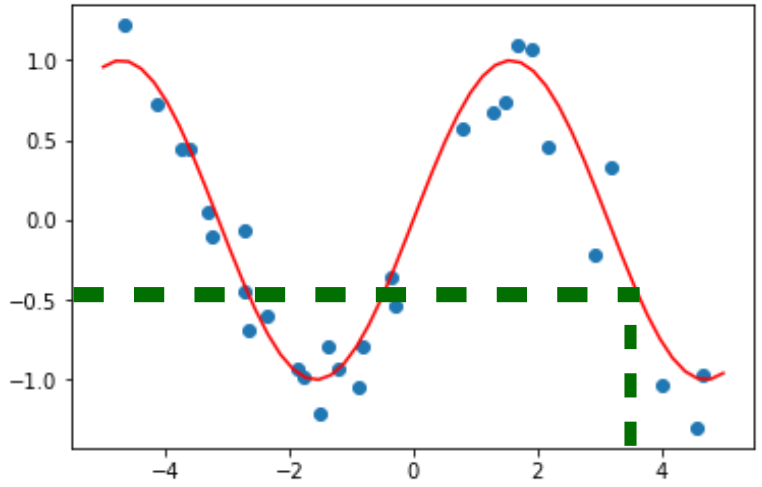
- This can also be seen as a prediction task
- Given examples of the value of the function, can we predict its value for points not given in the example?

What we are given:



Predict

We predict the value at 3.5 should be -0.45



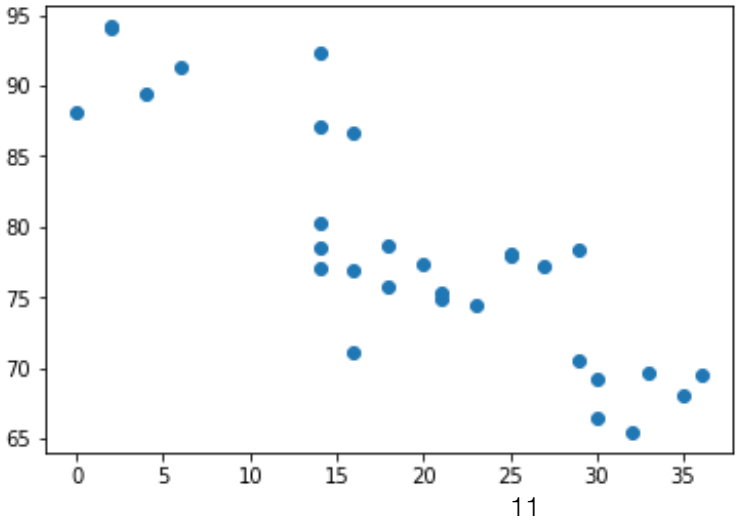
Example: Influence of Smoking on Life Expectancy

- For a more practical context, let us consider some Health-related situation. Let us suppose we gathered the data from many persons about:
  - **How many cigarettes** they were smoking per day; **How old** they died

- Now, knowing that somebody smokes x cigarettes per day, we would like to **predict** what age he is most likely to die

daily cigarets	age of death
32.0	73.471399
7.0	88.237207
30.0	82.077261
17.0	85.576741
27.0	76.190373
15.0	84.899030
20.0	72.598501
28.0	77.018773
....	.....

Age



Cigarettes

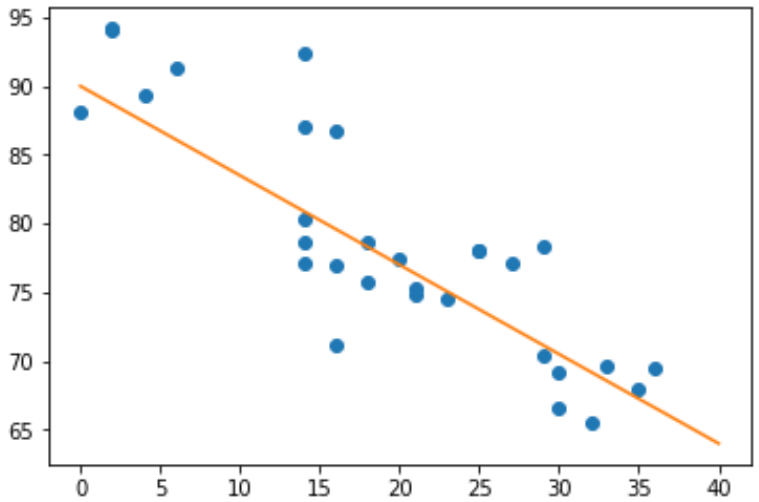
What age am I most likely to die if I smoke 10 cigarets per days?

Linear Regression (1/6)

- One possible answer is to do a ***linear regression***. We assume there is a *linear relation* between loss in life expectancy and number of cigarettes smoked
- Noting the life expectancy (or age of death) as *age* and the number of cigarettes smoked as *cig*, we suppose:

$$age = \theta_0 + \theta_1 \times cig$$

Unknown Parameters



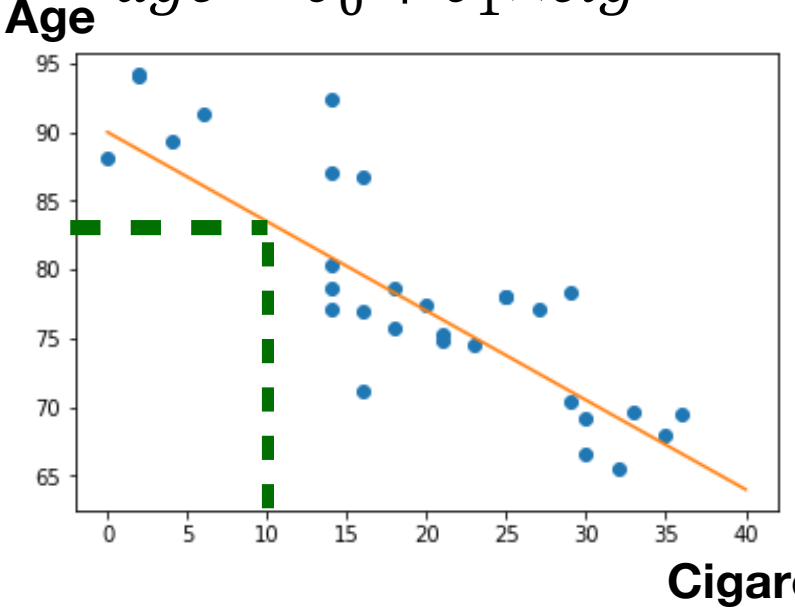
$\theta_0 \approx 90$ $\theta_1 \approx -0.7$

- The examples points are not exactly on a line: this is because cigarettes are just one of many things that influence life expectancy
- Two persons smoking the same number of cigarettes might have different health condition, eating habits, do different amount of sport, etc… But we do not have this information in our data.
- This is why we said earlier we expect examples to be noisy: in general, we do not know all of the hidden factors explaining the output.

Linear Regression (2/6)

- The linear regression consists in finding the good values for the parameters
- In our Machine Learning terminology, we could also say we are “learning” the function that compute the life expectancy from the number of cigarettes smoked

$$age = \theta_0 + \theta_1 \times cig \qquad \theta_0 \approx 90 \qquad \theta_1 \approx -0.7$$



- Now, we can make our prediction:
- What age am I most likely to die if I smoke 10 cigarettes per day?
  - 90-0.7x10 = 83 year old

Linear Regression (3/6)

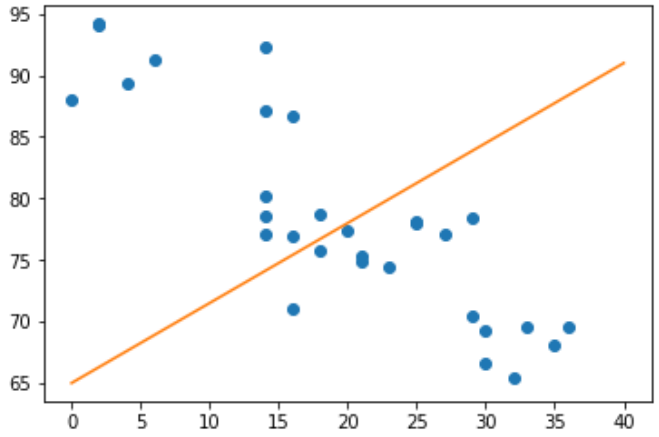
Age is a function of 3 variables:

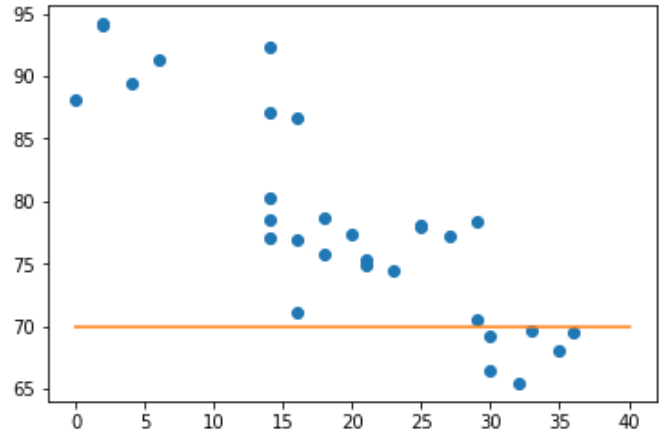
$$age(\theta_0, \theta_1, cig) = \theta_0 + \theta_1 \times cig$$

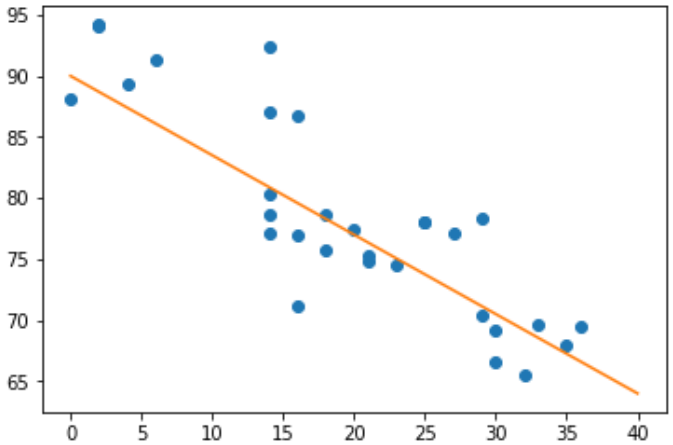
Age can also be seen as a parameterized function of one variable:

$$age_{\theta_0,\theta_1}(cig) = \theta_0 + \theta_1 \times cig$$

Age







Cigarettes

$\theta_0 \approx 65 \quad \theta_1 \approx 0.7$ $\theta_0 \approx 70 \quad \theta_1 \approx 0$ $\theta_0 \approx 90 \quad \theta_1 \approx -0.7$

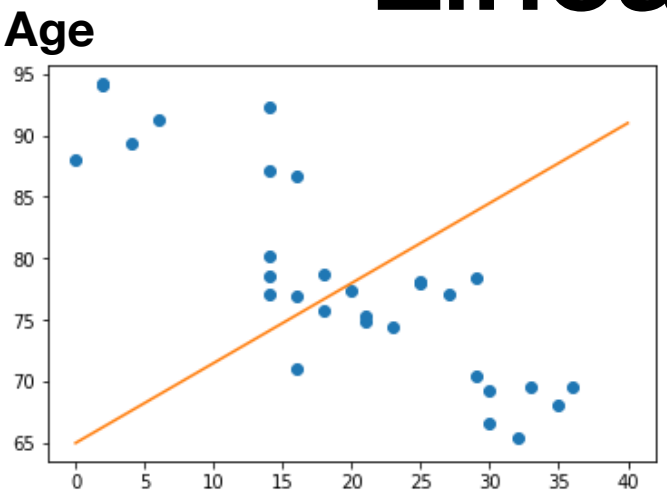
What age am I most likely to die if I smoke 10 cigarettes per day?

72 y.o. **70 y.o.** **83 y.o.**

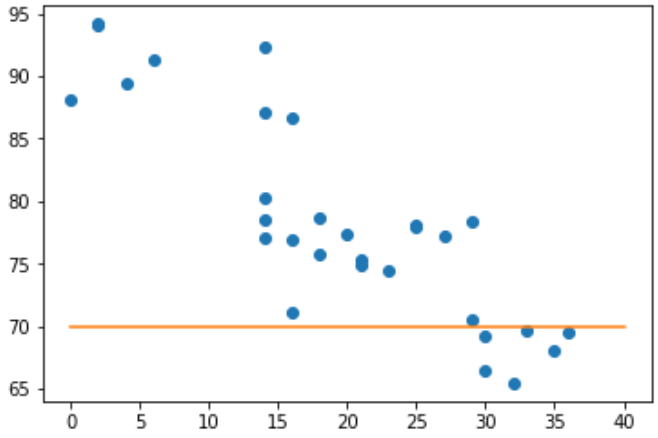
Linear Regression (4/6)

- Linear regression is a common tool in **statistics** and can also be seen as a simple **Machine Learning** task
- Point of view is a bit different (exaggerating a bit):
- In statistics, we are also interested in interpreting the parameters, like the slope of the linear approximation
  - Smoking one additional pack of cigarette is associated with losing 2.5 years of life expectancy
- In Machine Learning, we do not care so much about the parameters. We care about the **"predictive" power**
  - If I know how many pack of cigarettes a person smoke, can I predict what age she will die?

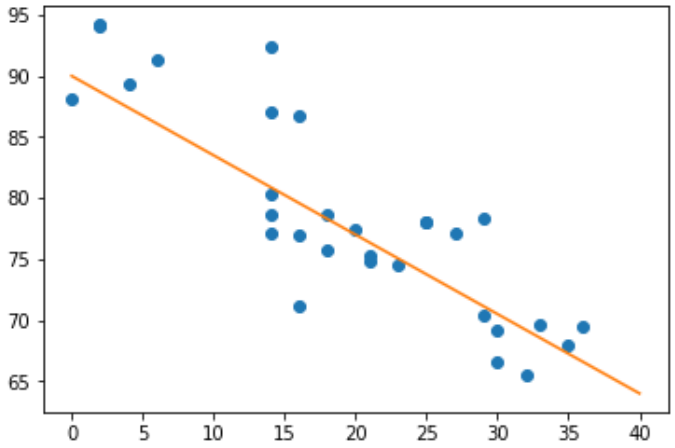
Linear Regression (5/6)



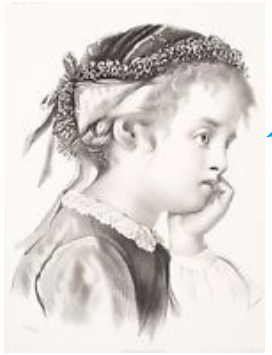
$\theta_0 \approx 65 \quad \theta_1 \approx 0.7$



$\theta_0 \approx 70 \quad \theta_1 \approx 0$



$\theta_0 \approx 90 \quad \theta_1 \approx -0.7$



So, which one is best?

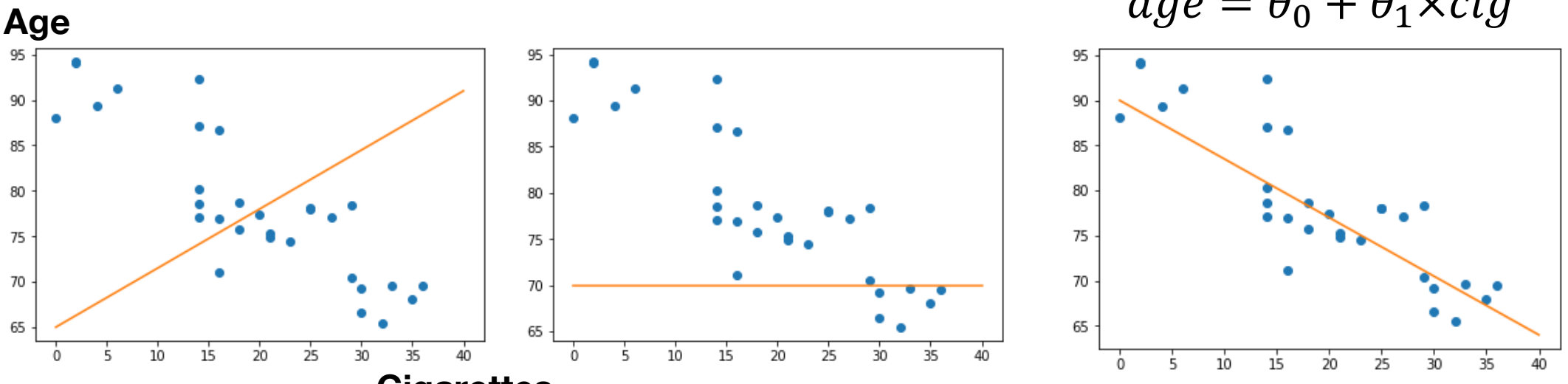
$$age = \theta_0 + \theta_1 \times cig$$

How to know which values of the parameters θ_0, θ_1 we should use?

Linear Regression (6/6)

- How are we going to find the values of the parameters θ_0, θ_1 ?
- We need a criterion to evaluate the **quality** of θ_0, θ_1.

$$age = \theta_0 + \theta_1 \times cig$$

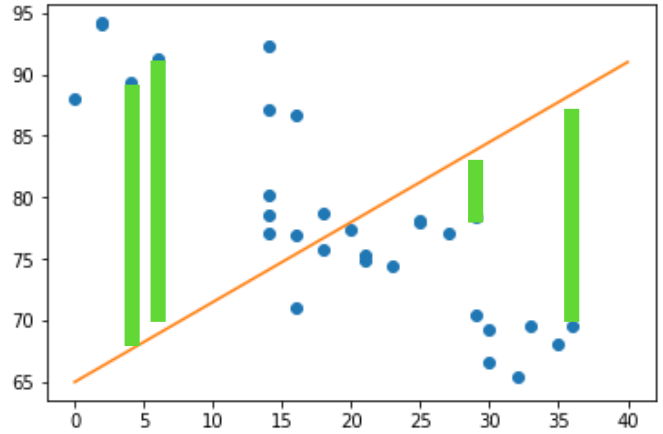


$\theta_0 \approx 65 \quad \theta_1 \approx 0.7$ $\theta_0 \approx 70 \quad \theta_1 \approx 0$ $\theta_0 \approx 90 \quad \theta_1 \approx -0.7$

Mean Squared Error (1/5)

- For each example, we consider the ***error:*** the distance between the example and the model predictions

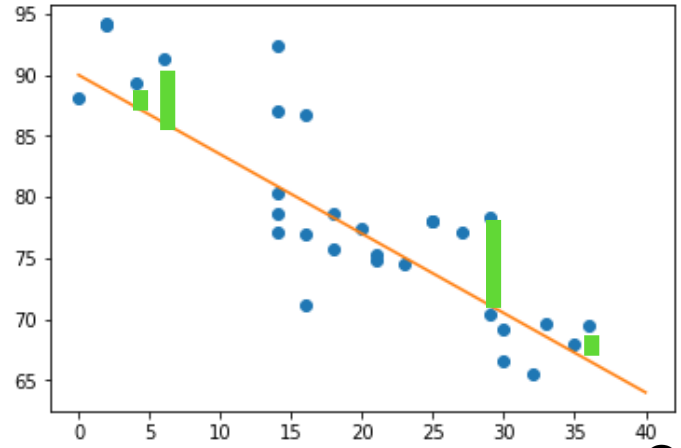
Age



Cigarettes

$\theta_0 \approx 65 \quad \theta_1 \approx 0.7$

Age

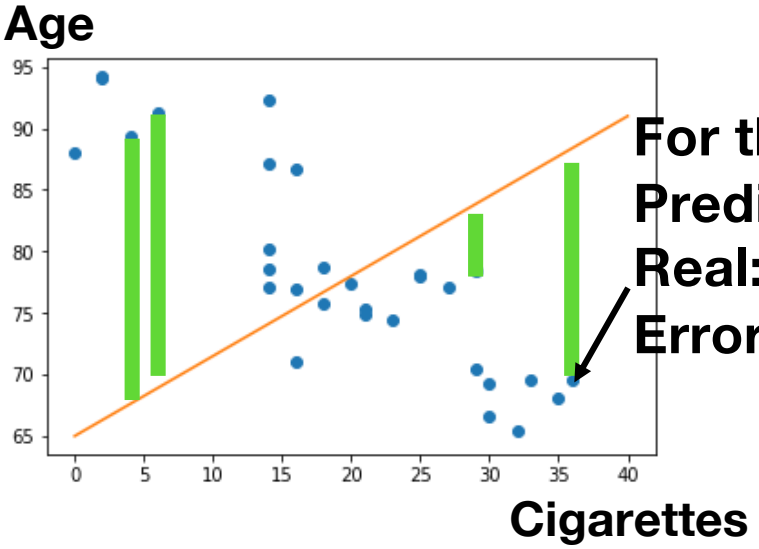


Cigarettes

$\theta_0 \approx 90 \quad \theta_1 \approx -0.7$

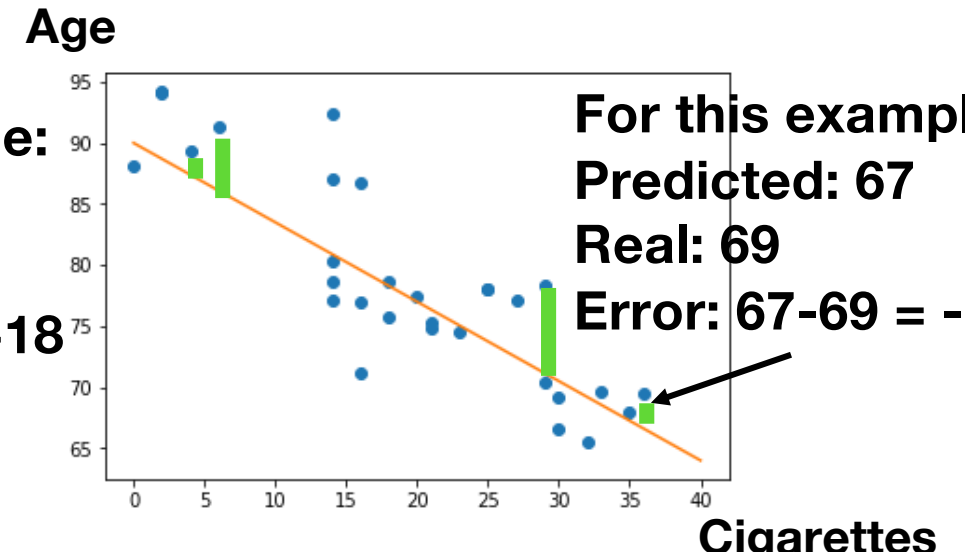
Mean Squared Error (2/5)

- For each example, we consider the ***error:*** the distance between the example and the model predictions



For this example:
Predicted: 87
Real: 69
Error: 87-69 = +18

$\theta_0 \approx 65 \quad \theta_1 \approx 0.7$

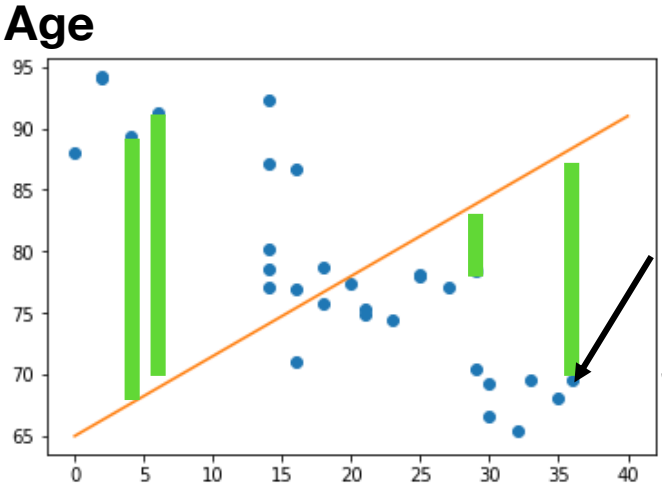


For this example:
Predicted: 67
Real: 69
Error: 67-69 = -2

$\theta_0 \approx 90 \quad \theta_1 \approx -0.7$

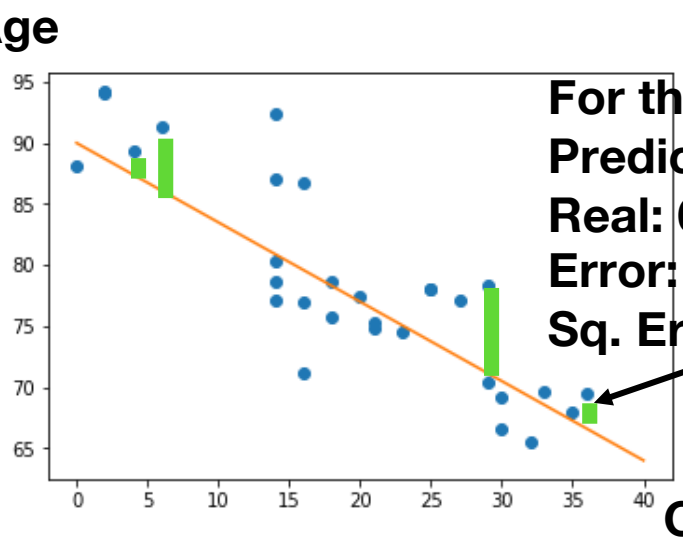
Mean Squared Error (3/5)

- For each example, we consider the ***error:*** the distance between the example and the model predictions
- We take the **squared error**



For this example:
Predicted: 87
Real: 69
Error: 87-69 = +18
Sq. Error = 164

$\theta_0 \approx 65 \quad \theta_1 \approx 0.7$

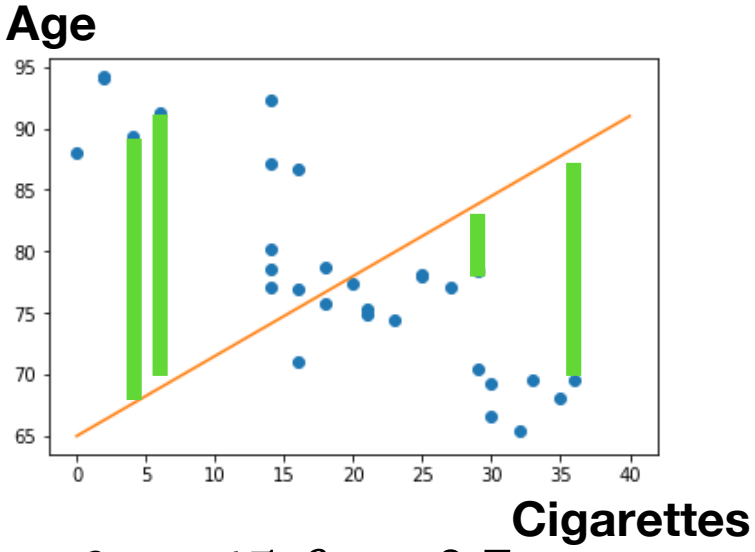


For this example:
Predicted: 67
Real: 69
Error: 67-69 = -2
Sq. Error = 4

$\theta_0 \approx 90 \quad \theta_1 \approx -0.7$

Mean Squared Error (4/5)

- Finally, we take **<u>average</u>** of the squared error for **all examples**



$\theta_0 \approx 65\ \theta_1 \approx 0.7$

$$MeanSquaredError = \frac{1}{N} \cdot \sum_i (error_i)^2$$

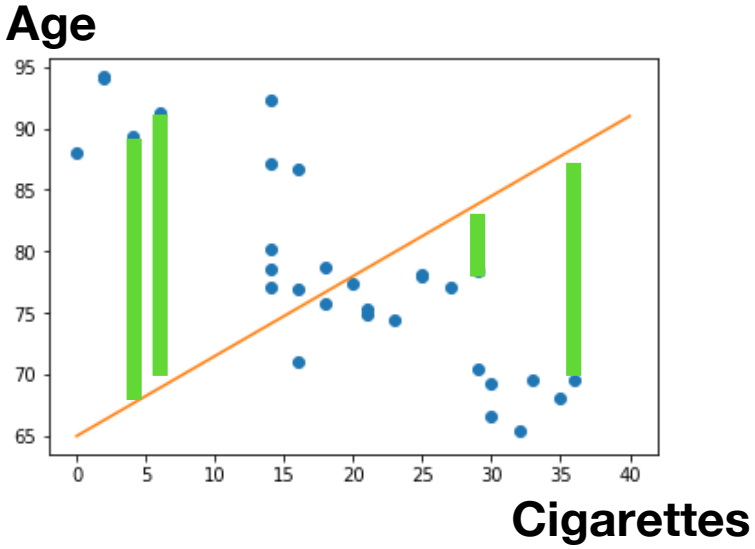
$$MeanSquaredError = \frac{1}{N} \cdot \sum_i (f(x_i) - y_i)^2$$

N: total number of examples
x_i: number of cigarettes smoked by person i
y_i: age person i died
$f(x_i)$: prediction of our model

In our case: $f(x_i) \sim age_{\theta_0,\theta_1}(cig)$

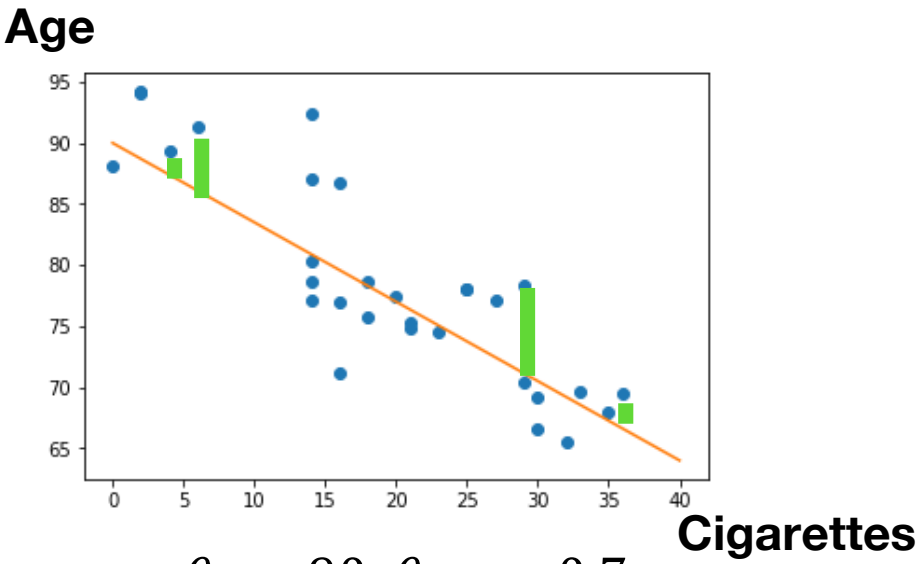
Mean Squared Error (5/5)

- The mean squared error now gives us a criterion for finding which parameters are best



$\theta_0 \approx 65 \quad \theta_1 \approx 0.7$

$MeanSquaredError = 294.7$



$\theta_0 \approx 90 \quad \theta_1 \approx -0.7$

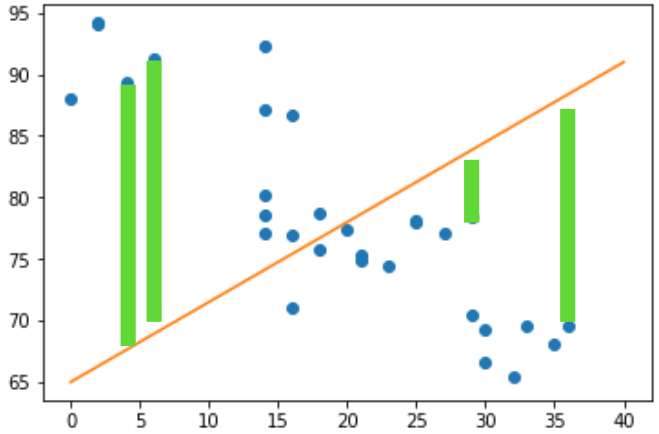
$MeanSquaredError = 18.3$

Minimizing the Mean Squared Error

$$age = \theta_0 + \theta_1 \times cig$$

- Now, we know what we want: we want to find θ_0, θ_1 such that the Mean Squared error is the smallest

Age

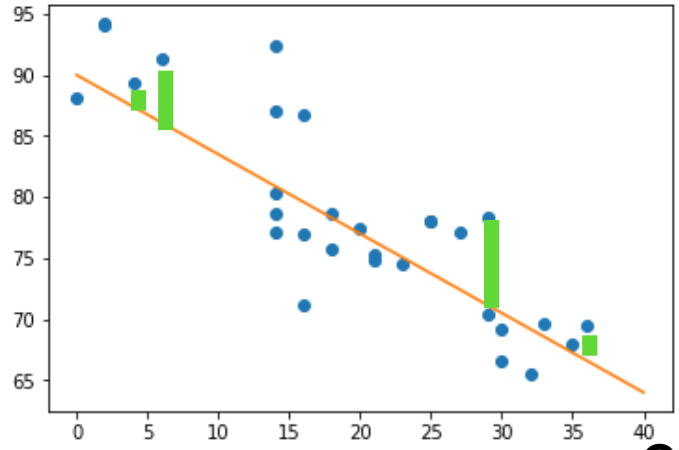


Cigarettes

$\theta_0 \approx 65 \quad \theta_1 \approx 0.7$

$MeanSquaredDistance = 294.7$

Age



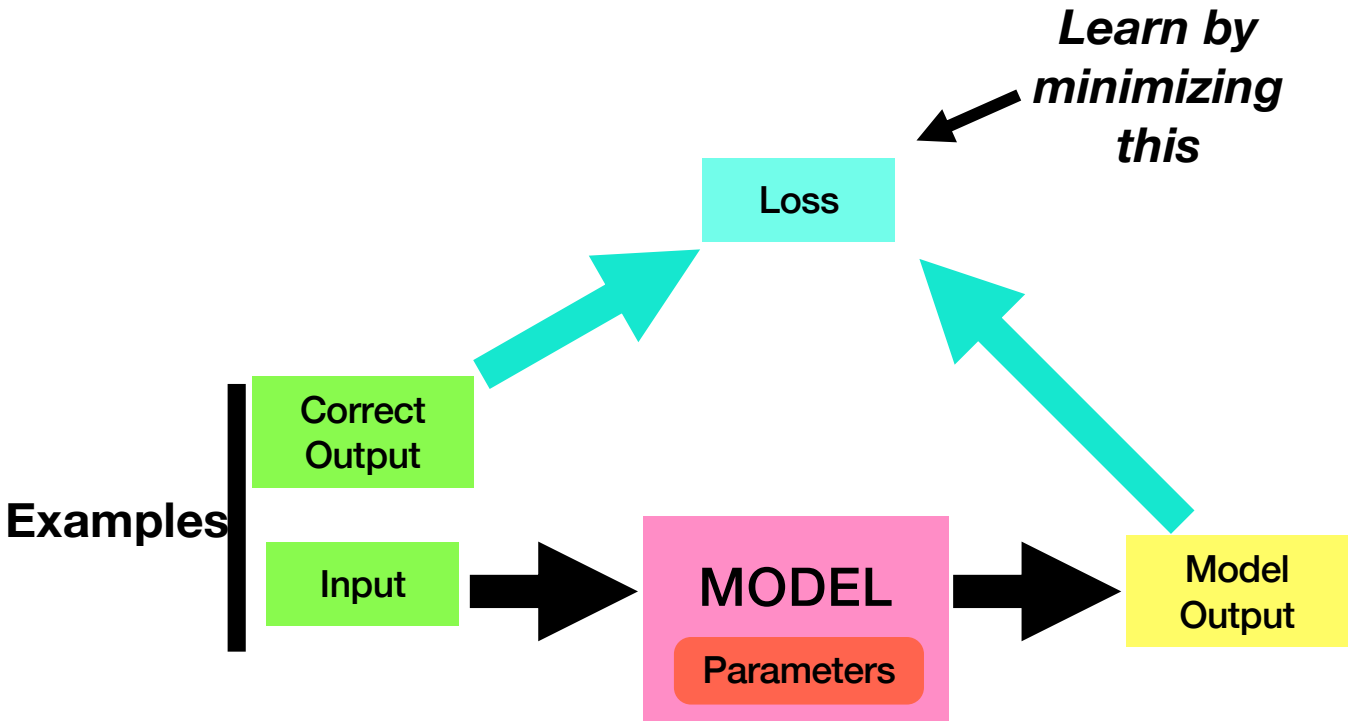
Cigarettes

$\theta_0 \approx 90 \quad \theta_1 \approx -0.7$

$MeanSquaredDistance = 18.3$

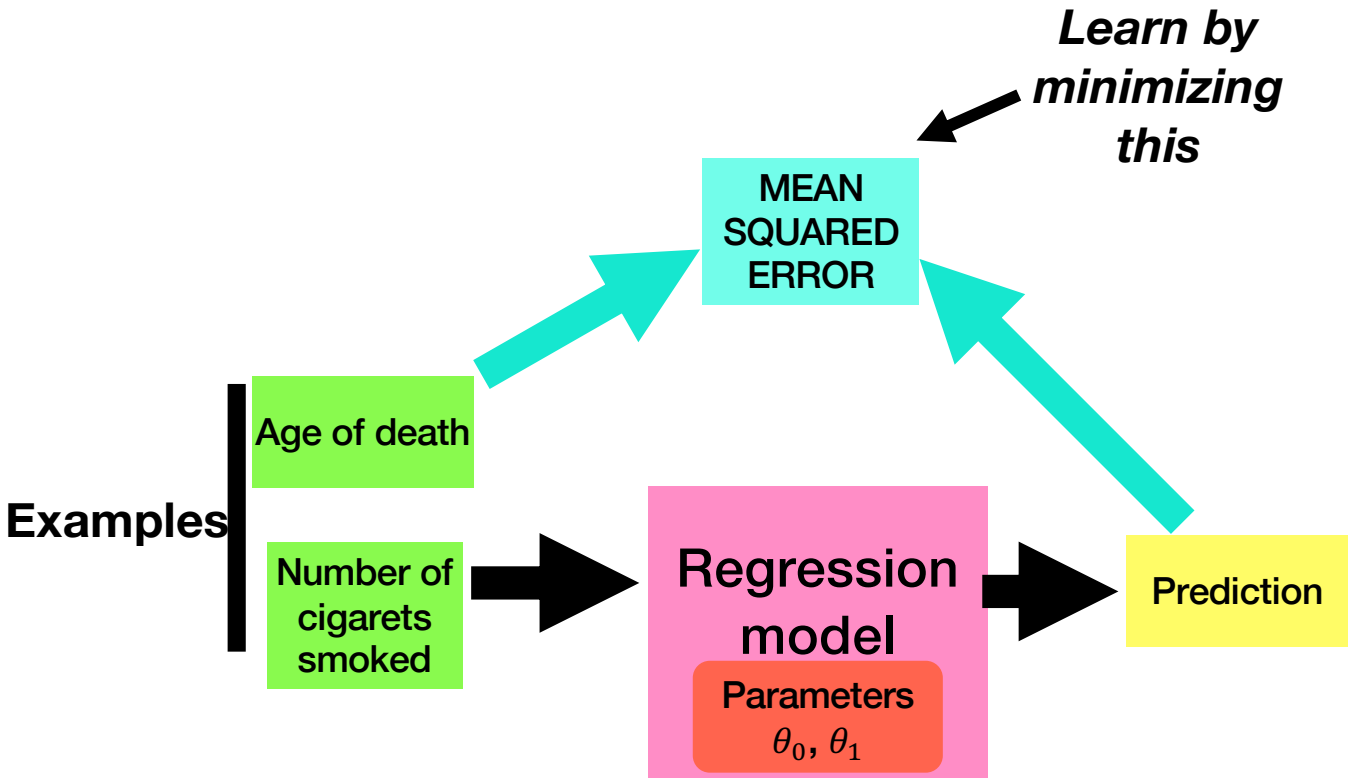
Supervised Learning (1/2)

- In supervised learning, we usually have:
  - A **MODEL**: a "parameterized" function that takes input and produces output
  - A ***Loss***: A function that computes how different the model output is from the correct output
  - ***Examples*** of input and correct output



Supervised Learning (2/2)

- In supervised learning, we usually have:
  - A **MODEL**: a "parameterized" function that takes input and produces output
  - A ***Loss***: A function that computes how different the model output is from the correct output
  - ***Examples*** of input and correct output



$$age_{\theta_0,\theta_1}(cig) = \theta_0 + \theta_1 \times cig$$

Minimizing the Mean Squared Error (1/4)

- We saw one method for “easily” minimizing a function: **gradient descent**
- We can apply it here
- We need to express the Mean Squared Distance as a function of θ_0, θ_1

$$MeanSquaredError = \frac{1}{N} \cdot \sum_i (f(x_i) - y_i)^2$$

x_i: number of cigarettes smoked by person i
y_i: age person i died
$f(x_i)$: prediction of our model
N: total number of persons in our data

With $f(x_i) = \theta_0 + \theta_1 \times x_i$

(just a rewriting of our model:)

$$age = \theta_0 + \theta_1 \times cig$$

Minimizing the Mean Squared Error (2/4)

- Therefore, the Mean Squared Distance as a function of θ_0, θ_1 is :

$$MeanSquaredError(\theta_0, \theta_1) = \frac{1}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i)^2$$

What is the gradient?

Minimizing the Mean Squared Error (3/4)

$$MeanSquaredError(\theta_0, \theta_1) = \frac{1}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i)^2$$

What is the gradient?

Minimizing the Mean Squared Error (4/4)

$$MeanSquaredError(\theta_0, \theta_1) = \frac{1}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i)^2$$

What is the gradient?

$$\begin{vmatrix} \frac{2}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i) \\ \\ \frac{2}{N} \cdot \sum_i x_i \times (\theta_0 + \theta_1 \times x_i - y_i) \end{vmatrix}$$

Gradient Descent for Linear Regression (1/7)

- Knowing the gradient, what would be the formula for the gradient descent update?

$$\theta_0 := ? \qquad \left| \quad \frac{2}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i) \right.$$

$$\theta_1 := ? \qquad \left| \quad \frac{2}{N} \cdot \sum_i x_i \times (\theta_0 + \theta_1 \times x_i - y_i) \right.$$

Gradient Descent for Linear Regression (2/7)

- Knowing the gradient, what would be the formula for the gradient descent update?

$$\theta_0 := \theta_0 - lr \times \frac{2}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i) \quad \left| \quad \frac{2}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i) \right.$$

$$\theta_1 := \theta_1 - lr \times \frac{2}{N} \cdot \sum_i x_i \times (\theta_0 + \theta_1 \times x_i - y_i) \quad \left| \quad \frac{2}{N} \cdot \sum_i x_i \times (\theta_0 + \theta_1 \times x_i - y_i) \right.$$

Gradient Descent for Linear Regression (3/7)

$$MeanSquaredError(\theta_0, \theta_1) = \frac{1}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i)^2$$

$$\theta_0 := \theta_0 - lr \times \frac{2}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i)$$

$$\theta_1 := \theta_1 - lr \times \frac{2}{N} \cdot \sum_i x_i \times (\theta_0 + \theta_1 \times x_i - y_i)$$

Let us start with θ_0, θ_1 =0, lr=0.1

$MSE(0,0) = ?$

$\theta_0 := \theta_0 - ?$

$\theta_1 := \theta_1 - ?$

Our data:

daily cigarettes	**age of death**
32.0	73
7.0	88
17.0	85

Gradient Descent for Linear Regression (4/7)

daily cigarettes	**age of death**
32.0	73
7.0	88
17.0	85

$$MeanSquaredError(\theta_0, \theta_1) = \frac{1}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i)^2$$

$$\theta_0 := \theta_0 - lr \times \frac{2}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i)$$

Let us start with θ_0, θ_1 =0, lr=0.1

$$\theta_1 := \theta_1 - lr \times \frac{2}{N} \cdot \sum_i x_i \times (\theta_0 + \theta_1 \times x_i - y_i)$$

$$MSD(0,0) = \frac{1}{3} \cdot ((0 + 0 \times 32 - 73)^2 + (0 + 0 \times 7 - 88)^2 + (0 + 0 \times 17 - 85)^2) = 6766$$

$$\theta_0 := \theta_0 - lr \times \frac{2}{3} \cdot ((0 + 0 \times 32 - 73) + (0 + 0 \times 7 - 88) + (0 + 0 \times 17 - 85)) = 16.4$$

$$\theta_1 := \theta_1 - lr \times \frac{2}{3} \cdot (32 \times (0 + 0 \times 32 - 73) + 7 \times (0 + 0 \times 7 - 88) + 17 \times (0 + 0 \times 17 - 85)) = 293.1$$

Gradient Descent for Linear Regression (5/7)

$$MeanSquaredError(\theta_0, \theta_1) = \frac{1}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i)^2$$

$$\theta_0 := \theta_0 - lr \times \frac{2}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i)$$

$$\theta_1 := \theta_1 - lr \times \frac{2}{N} \cdot \sum_i x_i \times (\theta_0 + \theta_1 \times x_i - y_i)$$

Our data:

daily cigarettes	**age of death**
32.0	73
7.0	88
17.0	85

Interesting question: can we get rid of the sigma?

The mean squared distance and its gradient are computed as an average over the data examples

If instead, we just compute the gradient for a random example, does it work?

Gradient Descent for Linear Regression (6/7)

daily cigarettes	**age of death**
32.0	73
7.0	88
17.0	85

$$\theta_0 := \theta_0 - lr \times \frac{2}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i)$$

$$\theta_1 := \theta_1 - lr \times \frac{2}{N} \cdot \sum_i x_i \times (\theta_0 + \theta_1 \times x_i - y_i)$$

Let us start with θ_0, θ_1 =0, lr=0.1

$$\theta_0 := \theta_0 - lr \times \frac{2}{3} \cdot ((0 + 0 \times 32 - 73) + (0 + 0 \times 7 - 88) + (0 + 0 \times 17 - 85)) = 16.4$$

$$\theta_1 := \theta_1 - lr \times \frac{2}{3} \cdot (32 \times (0 + 0 \times 32 - 73) + 7 \times (0 + 0 \times 7 - 88) + 17 \times (0 + 0 \times 17 - 85)) = 293.1$$

Instead, choose an example at random (e.g., Person 2)

$$\theta_0 := \theta_0 - lr \times 2 \cdot ((0 + 0 \times 7 - 88)) = 17.6 \qquad \theta_1 := \theta_1 - lr \times 2 \cdot (7 \times (0 + 0 \times 7 - 88)) = 123$$

Gradient Descent for Linear Regression (7/7)

$$MeanSquaredError(\theta_0, \theta_1) = \frac{1}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i)^2$$

$$\theta_0 := \theta_0 - lr \times \frac{2}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i)$$

$$\theta_1 := \theta_1 - lr \times \frac{2}{N} \cdot \sum_i x_i \times (\theta_0 + \theta_1 \times x_i - y_i)$$

Our data:

daily cigarettes	**age of death**
32.0	73
7.0	88
17.0	85

Interesting question: can we get rid of the sigma?

The mean squared distance and its gradient are computed as an average over the data examples

If instead, we just compute the gradient for a random example, does it work?

YES! (provided the learning rate is decreased over time: it is Stochastic Gradient Descent)

Stochastic Gradient Descent

- Stochastic gradient descent says that we can replace the average of the gradient over all examples by the gradient given by a randomly chosen example
- Slower Convergence
- But if we have one million examples: one million times faster to compute!

$$\theta_0 := \theta_0 - lr \times \frac{2}{N} \cdot \sum_i (\theta_0 + \theta_1 \times x_i - y_i)$$

Choose example i randomly

$$\theta_0 := \theta_0 - lr \times 2 \cdot (\theta_0 + \theta_1 \times x_i - y_i)$$

In practice, we often average over a few examples (instead of just one). This is called mini-batch gradient descent

Linear Regression: Other Methods

- Because Linear Regression is a very simple form of Machine Learning (we model data with a simple function), there are methods more direct to minimize the Mean Squared Distance (e.g., Normal Equations)
- However, for many equations and many variables, Stochastic Gradient Descent can still be the most efficient solution
- There are many existing implementations of Linear Regression, so in practice you would not need to do the gradient descent by yourself anyway
- For example, in python, we can use the function ***linregress*** in the package ***stats*** of the library ***scipy***:

```
In [1061]: from scipy import stats

In [1062]: stats.linregress(daily_cig_smoked, life_expectancy)

Out[1062]: LinregressResult(slope=-0.710478500965002, intercept=92.29875345880524, rvalue=-0.8609731
           085193286, pvalue=1.0335137971251332e-09, stderr=0.07932348624135113)
```

Next

- What if I have more than one variable I want to use?
  - E.g., Using Daily number of cigarettes, weight, BMI index and sport activity to predict age of death
- What if I want to learn a function more complex than a linear function?
- We will see that next time. But we will see that in practice the process is always the same:
  - Define the model
  - Define the loss
  - Do a gradient descent on the loss

Google Colab Notebook

https://shorturl.at/Yml7q

Report

- Submit the report google colab notebook **in pdf** via PandA
  - Submission due: **next lecture**
  - Name the pdf file as **student id_name**.