Artificial Intelligence and Automation

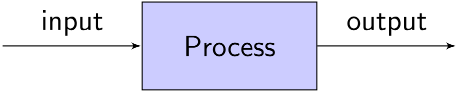
A Predicting Machine

Ph.D. Gerardo Marx Chvez-Campos

Instituto Tecnolgico de Morelia: Ing. Mecatrnica

A Basic Predicting Machine I

Let us start by proposing a basic **Machine** that can process information for us. We just input some data, and the machine " thinks" about the correct answer.



Now, let us introduce a specific **Machine** that can convert from Kilometers into Miles. In this example, multiply by a conversion factor θ.

$$\hat{y} = \theta \cdot x$$

A Basic Predicting Machine II

$$\hat{y} = \theta \cdot x \tag{1}$$

here:

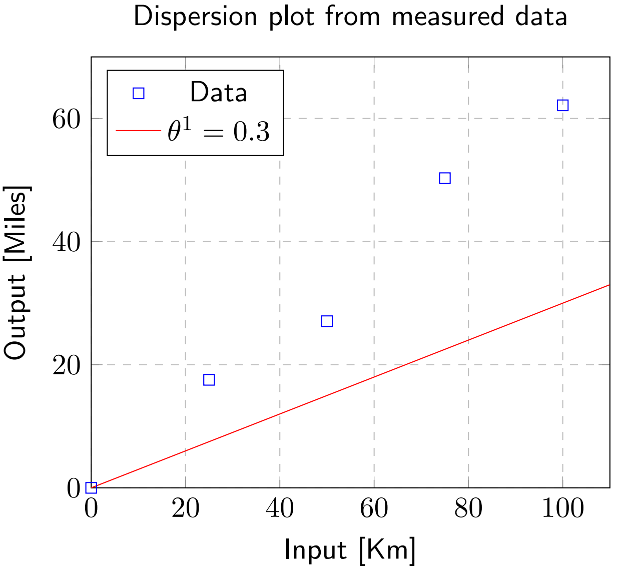
- x is the input data in kilometers
- θ is the unknown conversion factor, and
- $\hat{y}$ is the output converted into Miles

Adjusting the conversion factor I

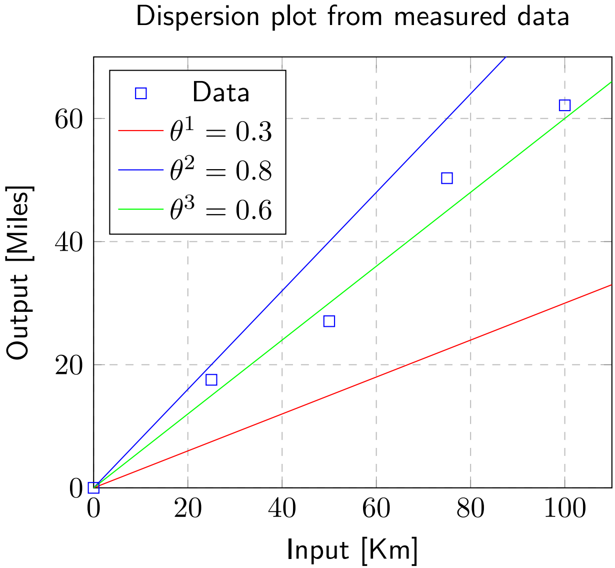
Now, we have to observe real data to **train** or **adjust** the conversion factor θ. The next data has been collected for some engineers by measuring the relationship between these two parameters:

Instance	**Data in**	**Converted**
1	0	0
2	25	15.5492
3	50	30.0684
4	100	62.1371

Adjusting the conversion factor II

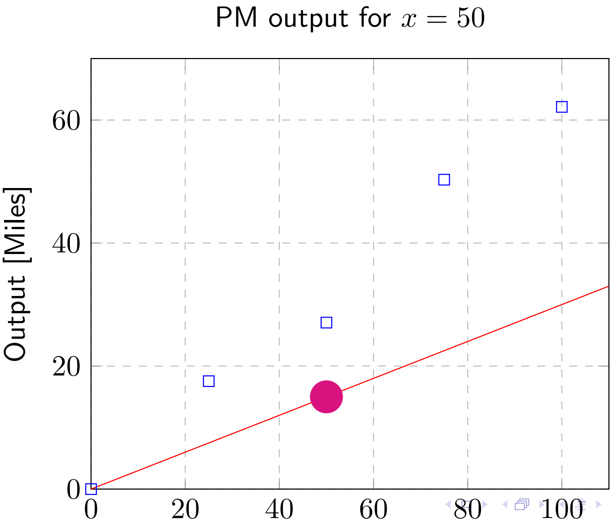


Adjusting the conversion factor III

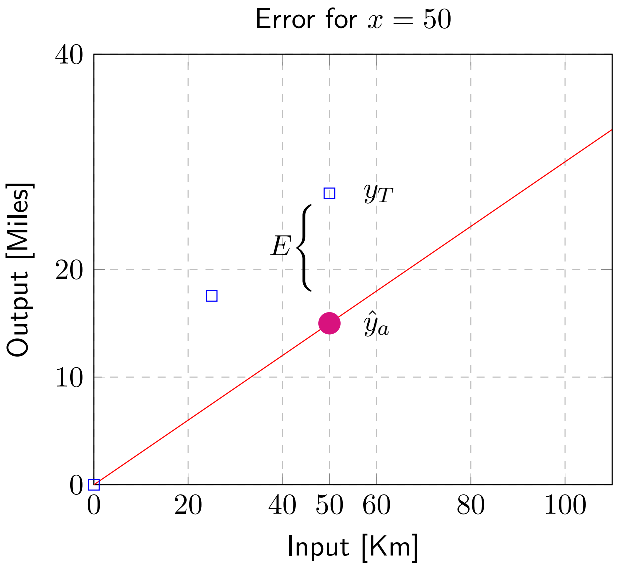


Example of a predicting machine

We have observed real data to **train** the conversion factor θ. Now try to understand how to train our **PM** with a first random approach $\theta^1 = 0.3$:

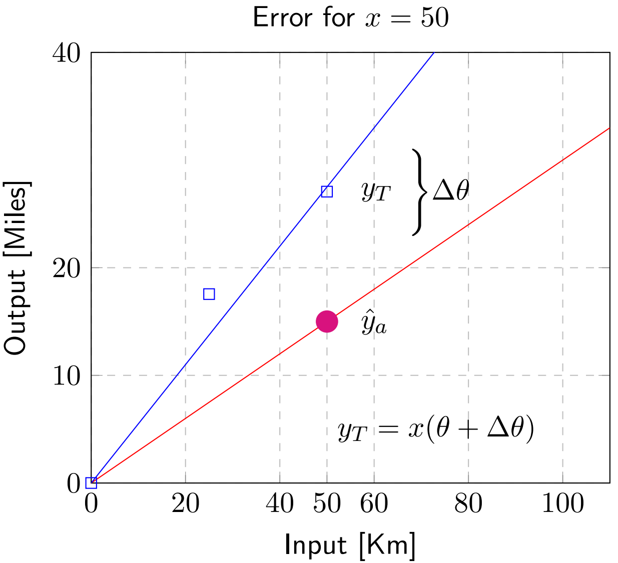


Example II



Error for $x = 50$
40
20
10
0
0
20
40
50
60
80
100
Output [Miles]
Input [Km]
y_T
E
$\hat{y}_a$

Example III



Error for $x = 50$
40
20
10
0
0
20
40
50
60
80
100
y_T
$\Delta\theta$
$\hat{y}_a$
$y_T = x(\theta + \Delta\theta)$
Output [Miles]
Input [Km]

Updating the parameter value I

Now, we can adjust the parameter value to a new one, therefore, a new model well be defined $\hat{y}$:

$$E = y_T - \hat{y}_a$$
$$E = \text{Desired output} - \text{Actual output}$$

Considering that target and actual output are computed with:

$$\hat{y}_a = \theta \cdot x$$
$$y_T = (\theta + \Delta\theta) \cdot x$$

Then, substituting on error formula:

$$E = (\theta + \Delta\theta) \cdot x - \theta \cdot x$$
$$E = \theta \cdot x + \Delta\theta \cdot x - \theta \cdot x$$
$$\Delta\theta = \frac{E}{x}$$

Updating the parameter value II

$$\theta_{New} = \theta_{old} + \Delta\theta \tag{2}$$

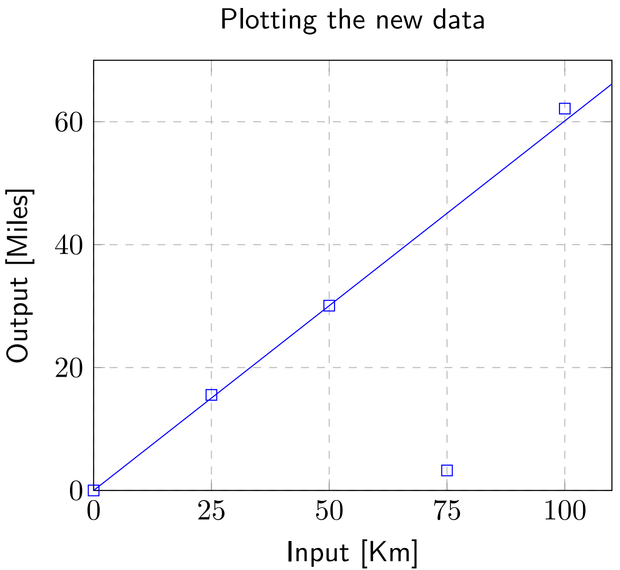
with $\theta = 0.3$ and $x^3 = 50$:

$$E = 30.0684 - 15$$
$$E = 15.0684$$
$$\theta_{New} = 0.3 + \frac{15.0684}{50}$$
$$\theta_{New} = 0.3 + 0.3013$$
$$\theta_{New} = 0.6013$$

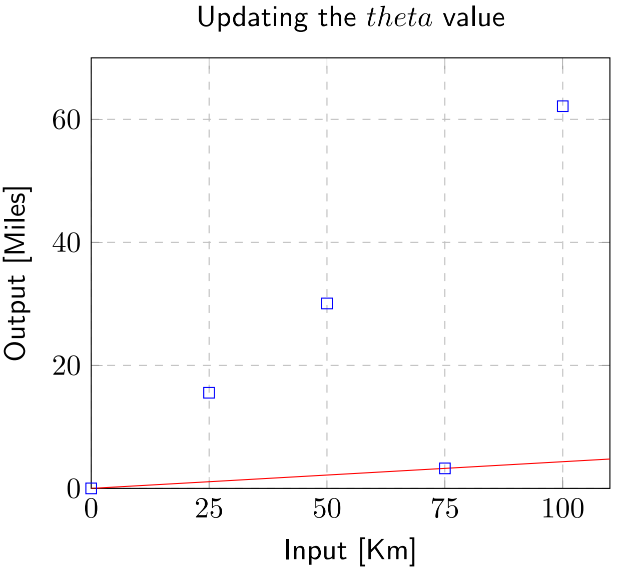
Outliers and learning rate

Instance	Data in	Converted
1	0	0
2	25	15.5492
3	50	30.0684
4	100	62.1371
5	75	3.2548

Example: Outlier



Example: Outlier II



The learning rate

- We updated the $theta$ value considering the new instance
- Now $\hat{y}$ gives the desired output
- However, the other values are forgiven
- What is wrong with this method?

Updating for each training data example, all we get is that the final update simply matches the last training example closely. **In effect we are throwing away any learning that previous training examples might gives us** and just learning from the last one.

How can we fix it?

Learning rate

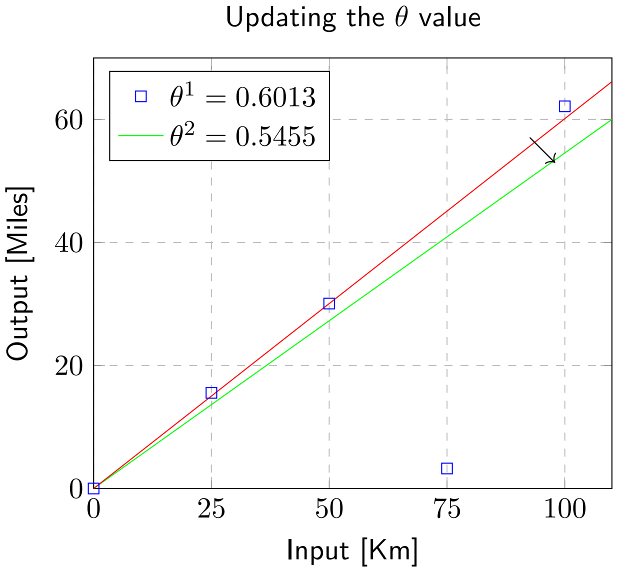
Thus, a way to moderate or update the model parameters, is by just moving a little bit in the new instance direction, but not completely:

$$\theta_{new} = lr \cdot \Delta\theta + \theta_{old} \tag{3}$$

here lr is knowing as the learning rate, and can help us to obtain a better model that will consider in equality all the instances.

Example: Outlier III

Using a $lr = 0.1$ the model update results:

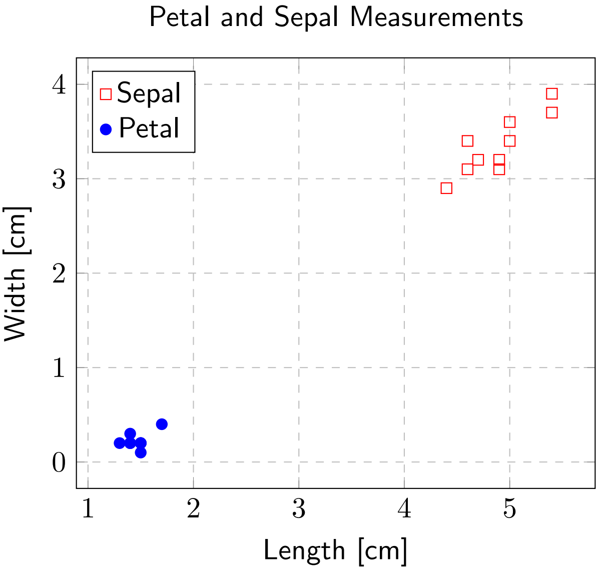


Classifying Machine I

Now, we have to observe real data to **train** or **adjust** the conversion factor c. The next data has been collected for some scientists by measuring the relationship between these two parameters:

Instance	Bug Length	Bug Width	Class
1	4.9	3.	a
2	4.7	3.2	a
3	4.6	3.1	a
4	5.	3.6	a
5	5.4	3.9	a
6	4.6	3.4	a
7	5.	3.4	a
8	4.4	2.9	a
9	4.9	3.1	a
10	5.4	3.7	a
11	1.4	0.2	b
12	1.3	0.2	b
13	1.5	0.2	b
14	1.4	0.2	b
15	1.7	0.4	b
16	1.4	0.3	b
17	1.5	0.2	b
18	1.4	0.2	b
19	1.5	0.1	b
20	1.5	0.2	b

Classifying Machine II



Classifying Machine III

Todo Modify example to make a 45o classifier

Ordinary Least Squares

Instance	Data in	Converted
1	0	0
2	25	15.5492
3	50	30.0684
4	100	62.1371

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x \tag{4}$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{N} x_i(y_i - \bar{y})}{\sum_{i=1}^{N} x_i(x_i - \bar{x})} \tag{5}$$

$$\hat{\beta}_0 = \bar{y} - \beta_1 \bar{x} \tag{6}$$

Ordinary Least Squares

Instance	Data in	Converted
1	0	0
2	25	15.5492
3	50	30.0684
4	100	62.1371

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{N} x_i(y_i - \bar{y})}{\sum_{i=1}^{N} x_i(x_i - \bar{x})} \tag{7}$$

$$\hat{\beta}_1 = \frac{0(0-26.93)+25(15.54-26.93)+50(30.06-26.93)+100(62.13-26.93)}{0(0-43.75)+25(25-43.75)+50(50-43.75)+100(100-43.75)} \tag{8}$$

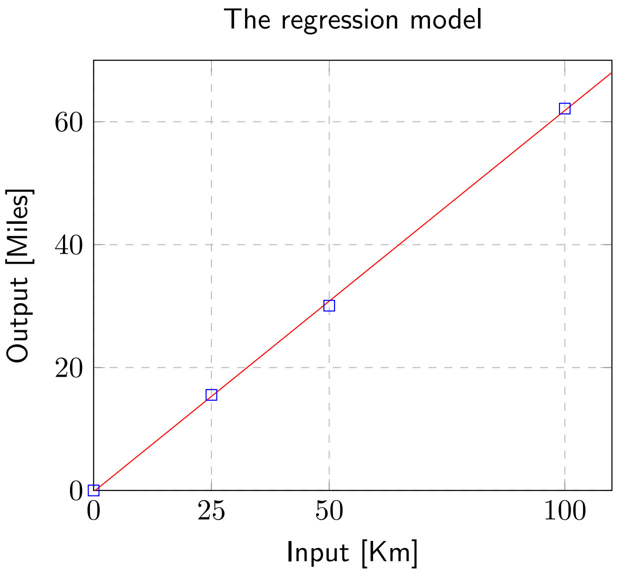
$$\beta_1 = \frac{3391.75}{5468.75} = 0.6202 \tag{9}$$

Ordinary Least Squares

$$\beta_1 = \frac{3391.75}{5468.75} = 0.6202 \tag{10}$$

$$\beta_0 = 43.75 - 0.6202(26.93) = -0.2037 \tag{11}$$

Example: OLS III



References

- https://www.mygreatlearning.com/blog/what-is-artificial-intelligence/
- https://www.formica.ai/blog/which-ai-is-learn-by-its-own