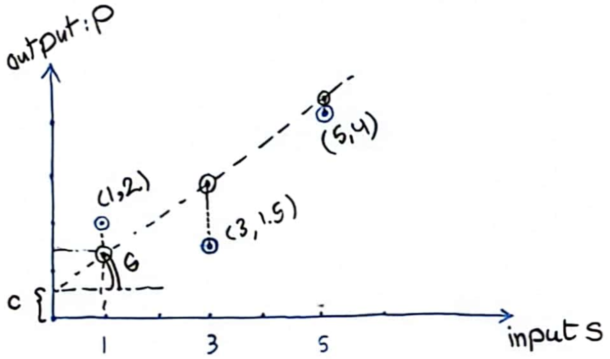
Linear Regression

using Gradient descent

starting with a simple problem.



Q: what is the best fiting line?

let's assume slope is already known and we want to get the intercept.

$$\boxed{\text{so } y = W_0 + X \cdot W_1 \quad W_1 = 1}$$

so to get the best value for the intercept we will construct a loss Function (least square) and then use Gradient-descent to optimize it

---

$$\text{Loss Function} = \sum (\underbrace{P}_{\text{real}} - \underbrace{P(s)}_{\text{pridicted}})^2$$

$$\text{so } L.F = \sum_{1}^{n} \Big[P - (\overbrace{W_0}^{\text{intercept}} + \overbrace{W_1}^{\text{slop}=1} \overbrace{S_1}^{1})\Big]^2 \qquad S_1, S_2, S_3, \dots, n$$

$$= [2-(W_0+1)]^2 + [3-(W_0+1.5)]^2 + [4-(W_0+5)]^2$$

---

so now we take the $\frac{dL.F}{dW_0}$ to get the $\overset{\text{gradient}}{\text{slop}}$ of the loss Function with respect to W_0

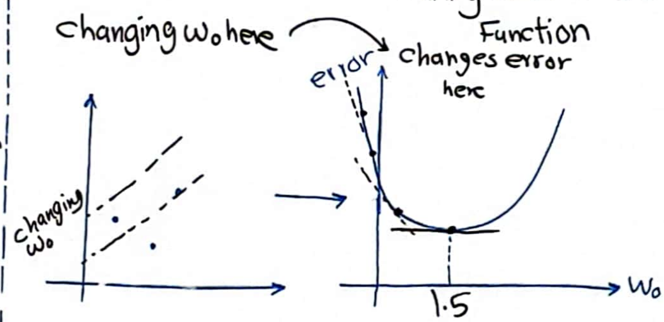
$$\text{so } \frac{d\,L.F}{dW_0} = -2[2-(W_0+1)] -2[3-(W_0+1.5)] -2[4-(W_0+4)]$$

---

now using this equation

$$\boxed{W_0\_New = W_0\_old - slope * (\text{learning rate})}$$

using code or calculator



so some iteration the answer

the answer For $W_0 = 1.5$

---

this was for a single parameter W_0

let's take a look at W_0 & W_1

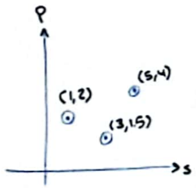
Linear Regression

using Gradient descent

so now working with two Parameters

$$y = w_0 + w_1 x$$

Notice that
Number of parameters
equals N.O Features + 1



using the same input points

n = N.O examples "points"

$$\text{LOSS Function} = \sum_{1}^{n} (P - P(S_n))^2$$

$$= (2-(w_0+w_1 1))^2 + (1.5-(w_0+w_1 3))^2 + (4-(w_0+w_1 5))^2$$

Now we have a Function of 2 Variables so we should take the drivative with respect to each one of them

$$\frac{\partial L.F}{\partial w_0} = -2[2-(w_0+w_1)] - 2[1.5-(w_0+3w_1)] - 2[4-(w_0+w_1 5)]$$

$$\frac{\partial L.F}{\partial w_1} = -2[2-(w_0+w_1)] - 2\times 3[1.5-(w_0+3w_1)] - 2\times 5[4-(w_0+w_1\times 5)]$$

so now we need to Calculate w_0_New & w_1_new

$$w_0\text{-new} = w_0\text{-old} - \frac{\partial L.F}{\partial w_0} \times \text{learning Rate}$$

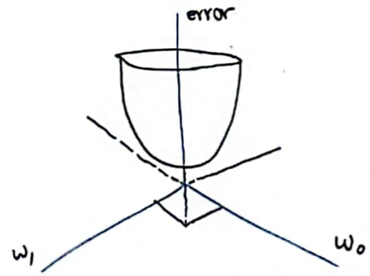
$$w_1\text{-new} = w_1\text{-old} - \frac{\partial L.F}{\partial w_1} \times \text{learning Rate}$$

we can Turn it into Vector operations

$$\begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = \begin{bmatrix} w_0\text{-old} \\ w_1\text{-old} \end{bmatrix} - \begin{bmatrix} \frac{\partial L.F}{\partial w_0} \\ \frac{\partial L.F}{\partial w_1} \end{bmatrix} * \text{Learning Rate}$$

using code for iterating & updating we will reach the minimum error

Looking at the loss Function in 3d and BTW this is the limit of Visualizing



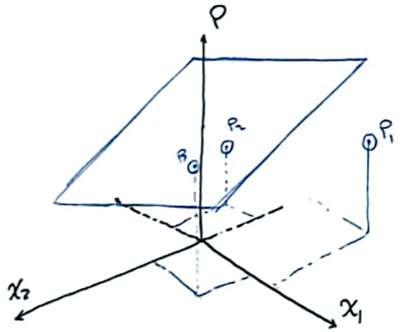
Linear Regression

using Gradient descent

let's take a look if we increased the

Number of the inputs to 2:

we will have something like this



now the problem will be Fiting this plane to these points

$$y = W_0 X_0^n + W_1 X_1^n + W_2 X_2^n$$

the equation of the plane

where $\boxed{X_0^n = 1}$

n is number of point where each point $(x_1, x_2, \overset{\text{target}}{P})$

we will do the same as previous

but now we will try to Vectorize these operations

working with multi variate problems:

$$y = \sum_{i=0}^{D} (W_i X_i^{(n)})$$

D is the Number of Features so D+1 param

$\overbrace{0 \longrightarrow D}$ 0'indexed

and as Known

$$\underset{1 \times D+1}{[W_0\, W_1\, W_2\, W_3]} \odot \begin{bmatrix} 1 & 1 & 1 \\ x_1^0 & X_1^1 & X_1^2 \\ x_2^0 & X_2^1 & X_2^2 \\ x_3^0 & X_3^1 & X_3^2 \\ \vdots & & \end{bmatrix} = y(W_i, X_i^n)$$

$$W^T \cdot X^n \quad D+1, n$$

$$y = W^T . \underset{\boxed{1 \times n}}{X^n} \longleftarrow$$ So this is the hyper-plane Function after Vectorizing it

so the new loss Function

$$cost(W_i) = \frac{1}{2N} \sum_{n=1}^{N} \left(y(x_i^n, W_i) - t^n\right)^2$$

$$\frac{\partial cost}{\partial W_i} = \frac{1}{N} \sum_{n=1}^{N} (y(x_i^n, w_i) - t^n) . X_i^n$$

N = number of points

Vectorizing the $\frac{\partial cost}{\partial W_i}$

$$\frac{\partial cost}{\partial W_i} = \sum_{n=1}^{N} \left(y(X_i^n, w_i) - t^n \right) \cdot X_i^n$$

$$y(X_i^n, W_i) - t^n$$

$$\begin{bmatrix} \text{sum of errors}_1 & \text{sum of errors}_2 & \cdots N \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 & \\ X_1^0 & X_1^1 & \\ X_2^0 & X_2^1 & \cdots N \\ X_3^0 & X_3^1 & \end{bmatrix}$$

$1 \times \underset{\sim}{N}$ $\qquad$ $D+1 \times N$

$$\underset{D+1 \times \boxed{N}}{X} \odot \underset{\boxed{N} \times 1}{\left[y(X_i^n, W_i^n) - t^n \right]^T}$$

let's call it E

so now

$$\frac{\partial cost}{\partial W_i} = \begin{bmatrix} \frac{\partial cost}{\partial w_0} \\ \frac{\partial cost}{\partial w_1} \\ \vdots \\ D+1 \end{bmatrix} = X \odot E^{T}$$

Transpose

الحمد لله