A first course in Machine Learning
Chapter 1 - Linear moddelling

Defining a model

input, x output, t

$t = f(x)$ $t = ax$

where a is a paremeter that needs to be defined somehow

$t = f(x; a)$

Linear relationship = $y = mx + c$

Avoid 0 outputs

- if x = year & $y = 0$ output will always be 0 $t = a \cdot 0 = 0$
- This is illogical hence new para acts as start point / intercept

$$t = f(x; w_0, w_1) = \underline{\underline{w_0}} + w_1 x$$

w_0 = intercept
w_1 = gradient

1.1.3 Define a good model

- we want to find best values of w_0, w_1
- to do this use a loss function

$$(t_n - f(x_n; w_0, w_1))^2$$

- Above known as squared loss func
- Describes accuracy of w_0, w_1 against actual outcomes
- $L_n()$ = Loss function

$$L_n(t_n, f(x_n; w_0, w_1)) = (t_n - f(x_n; w_0, w_1))^2$$

Loss function should be averaged against against all ~~row~~ entries (n) and minimised

$$L = \frac{1}{N} \sum_{n=1}^{N} L_n(t_n, f(x_n; w_0, w_1))$$

$\underset{w_0, w_1}{\text{Argmin}}$ is maths way to express optimized w_0, w_1 vals
= "find args that minimise ..."

1.1.4 Least Squares Solution

Historically the minimisation of the squared loss function comes from least squares error estimation. With modern tech other loss functions are computationally viable & may be more suited

Worked Example:

find relationship = $f(x; w_0, w_1) = w_0 + w_1 x$

$$L = \frac{1}{N} \sum_{n=1}^{N} L_n(t_n, f(x_n; w_0, w_1))$$

$$\frac{1}{N} \sum \cancel{L_n} (t_n - f(x_n : w_0, w_1))^2$$

$$\frac{1}{N} \sum (t_n - (w_0 + w_1 x_n))^2$$

$$\frac{1}{N} \sum (w_1^2 x_n^2 + \underbrace{2w_1 x_n w_0 - 2w_1 x_n t_n}_{2w_1 x_2 (w_0 - t_n)} + w_0^2 + 2w_0 t_n + t_n^2)$$

Above = sub in loss function & multi & factorize turning Points: ← 1st order Deriv

- points where gradient of function = 0
- Could be min, max or saddle point
- To determin take second order Deriv
  - Min = Pos, Max = Neg, Sad = 0

Differentiating the loss functions

- at a min, the partial derivs w/ respect to w_0, w_1 should be 0
- Compute, set to 0 & solve
- Partial Derivative w/ respect to w_1 removes any that doesnt include w_1 (turns to 0) as is constant)
- Advised to rearrange to take n indexing out of sum (Σ) before Partial derivative

$$\frac{1}{N}\sum(w_1^2 x_n^2) \longrightarrow w_1^2 \frac{1}{N}\left(\sum x_n^2\right)$$

- Set Partialled equation and set to 0

$$2w_0 + 2w_1\frac{1}{N}\left(\sum x_n\right) - \frac{2}{N}\left(\sum t_n\right) = 0$$

$$w_0 = \frac{1}{N}\left(\sum t_n\right) - w_1\frac{1}{N}\left(\sum x_n\right)$$

note

- $\frac{1}{N}\left(\sum t_n\right)$ = Average t $= \bar{t}$
- $\frac{1}{N}\left(\sum x_n\right)$ = Averge x $= \bar{x}$

So, $\hat{w}_0 = \bar{t} - w_1\bar{x}$

this $\hat{w}_0$ depends on w_1 and can now be subbed to the ~~loss min function~~ w_1 Partial deriuate, Set to 0 and solve for w_1

$$\hat{w}_1 = \frac{\overline{xt} - \bar{x}\,\bar{t}}{\overline{x^2} - (\bar{x})^2}$$

$\overline{x^2}$ = Average square valu of the data

$\overline{xt}$ =

Simple worked example of LM

n	x_n	t_n	$x_n t_n$	x_n^2
1	1	4.8	4.8	1
2	3	11.3	33.9	9
3	5	17.2	86	25
$\frac{1}{N}\sum$	3	11.1	41.57	11.67

$$\hat{w}_1 = \frac{\overline{xt} - \bar{x}\bar{t}}{\overline{x^2} - (\bar{x})^2} = \frac{41.57 - (3 * 11.1)}{11.67 - (3 * 3)} = 3.1$$

$$\hat{w}_0 = \bar{t} - w_1 \bar{x} = 11.1 - (3.1 * 3) = 1.8$$

best function $= f(x : w_0, w_1) = 1.8 + 3.1x$

Plug any x value to get prediction