ML lecture 1 - Introduction

$x_n \in R^{D_x}$, bold letter $\mathbf{x}$ = tensor

linear model = relationship between x & y

L2 loss = Euclidean = Mean Squared Err

$$L_2(w) = \frac{1}{N}\sum_{n=1}^{N}(y^T - y_n)^2$$
$$= \frac{1}{N}\sum(x_n \cdot w + b - y_n)^2$$
$$= L_2(w)\frac{1}{N}\|xw - y\|^2$$

Recall norm $\|.\|$ = measure size or length of a vector (magnitude)

$\|.\|_{L2}$ = Square, Sum, Square root

Why this sequence of actions?

- Pythag theorm
- Vectors hold direction, summing them plainly may cause (+) & (-) to cancel out
- A norm must be non-neg

Steps for $\frac{1}{N}||xw - y||^2$

① Diff of vectors: $xw - y$
e.g. $[1-2, 4-1, 3-5] = [-1, 3, -2]$
vector → vector

② Apply the L2 Norm: $||xw - y||$
- Square Comps $-1^2, 3^2, -2^2 = [1\ 9\ 4]$
- Sum $1 + 9 + 4 = 14$
- Sq Root $14 = \sqrt{14}$

③ Square Result: $\sqrt{14} \rightarrow 14$

④ Divide by N: $\frac{1}{N}\ 14$

Why Square at end?
- by Definition a Norm $||.||$ involves a Square root to represent the true distance (Pytag theorem)
- we reapply the Square to allow for better/easier maths in future
- To minimize loss we take derivatie of the function/equation
- Squared is much easer to work with (chain rule)

Recall optimal model parameters minimize the loss (w, b)

Training is the process of optimizing w & b

Minimize the loss function:

- find the min when its gradient (derivative) is set to 0
- gradient = Derivative

$$\left.\frac{d\,L^2(w)}{dw} = 0\right\}$$ Deriv of loss func set to 0

Expand & Sub in loss function:

$$\frac{1}{N}\frac{d(\|x w \cdot y\|^2)}{dw}$$

Recall $\frac{d(w)}{dw}$ is the notation that means derivative of a function w/ respect to w

D = infinite small

$d(w)$ = " change in function

dw = " change in var w

Keep expanding numerator:

$d((xw-y)^T(xw-y))/dw = 0$

$\hookrightarrow d(w^T x^T xw - 2w^T x^T y + y^T y)/dw = 0$

Apply derivative w/ respect to w

$2x^T xw - 2x^T y$

Rearrange to get w (goal)

$w = (x^T x)^{-1} x^T y$

w gives optimal linear model

L1 vs L2 Norm:

- when Differientated, L1 has no w term
- L1 gradient is a constant
- constants do show rate of change
- cant see how close a result is

Model Generalizability errors:
Bais & variance

high bias = less sensitive to train = less complex = risk of under fit

high Sens = More sensitive to train = More complex = risk of overfit

overfitting = Memorize training data. will not generalize well as has not learnt from data

Regularization:
- To tackle overfitting
- Penalty to loss function
-

L1 reg = Lasso

$$\mathcal{L}_{lasso}(w) = \frac{1}{N}\|xw - y\|^2 + \underbrace{a|w|}_{\text{reg term}}$$

- Penalizes complexity
- Complexity = WS

- L1 Penalize non-zero weights
- encourages $w=0$ to reduce complex

L2 Regularization:
+ $\alpha \|w\|^2$
- encourages small weights to reduce comp

Classification Models:

$\{(x_n, y_n)\}_{n=1}^{N}, x_n \in \mathbb{R}^{Dx}, y_n \in \mathbb{Z}^{Dy}$

Vector → category

$f(x) = \sigma(xw+b) = \hat{y}$ — $\sigma(\cdot)$ - activation f - discrete values

Classification sign loss: or 0-1 loss

$L(w) = \frac{1}{N}\sum 1 \cdot \text{Sign}(x_n w + b) \neq y_n$

Hinge loss: $L(w) = \frac{1}{N}\sum \max(0, -y_n(x_n w + b))$

- hinge allow how far yes or NO

linear model

Pos or Neg