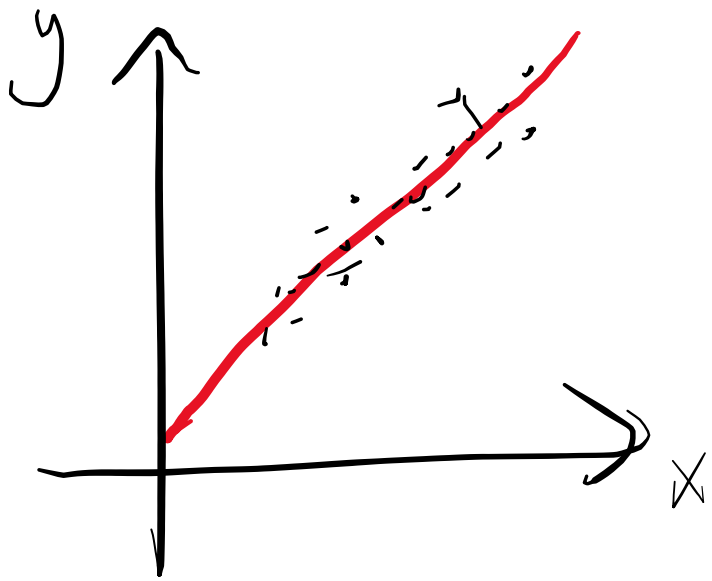
- Please join Slack workspace and channel
- Please leave your video on if you can.

A closer look at regression

in particular linear least squares regression.



$$L = \sum_{i=1}^{N} \left(y_i - f(x_i; w) \right)^2$$

$$f(x) = ax + b$$

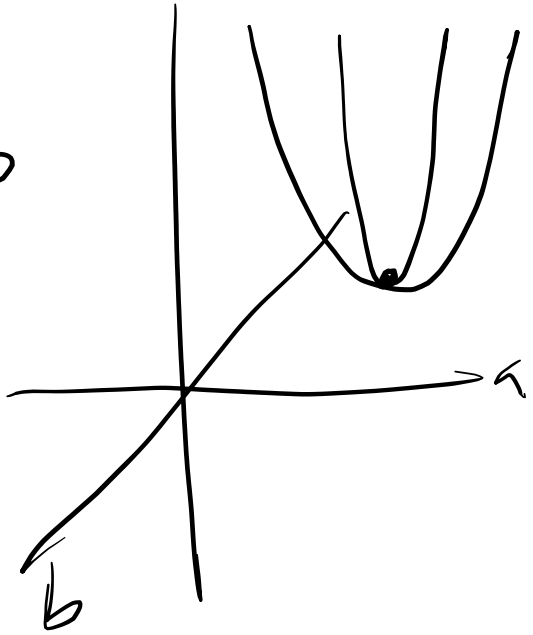
Mean squared error (MSE)

$$\frac{\partial L}{\partial a} = \frac{\partial L}{\partial b} = 0.$$

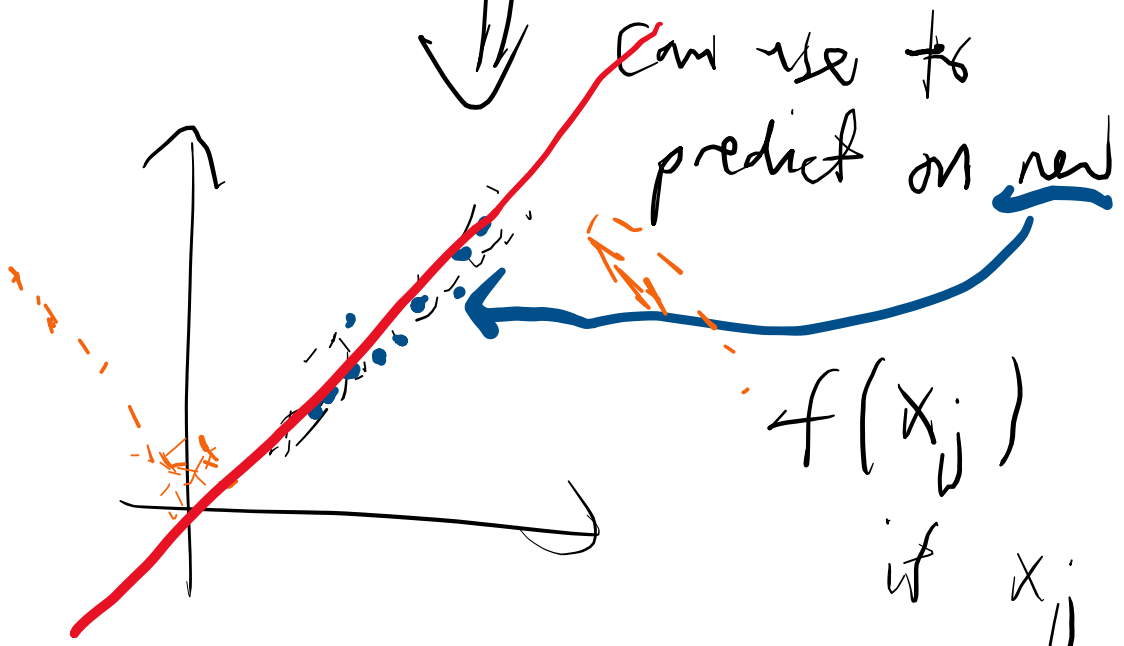
L is convex wrt a, b

$\exists$ a global min wrt a, b

$$\frac{\partial L}{\partial a} = \frac{\partial L}{\partial b} = 0 \quad \text{trivial to solve.}$$



$\Downarrow$



can use to predict on new data

$f(x_j)$ $x_j \in$ new data

if x_j in previous domain $\rightarrow$ interpolation ✓

if new data is outside range ⟶ extrapolation ✗

DANGEROUS

In ML:

loss or objective function

usually trying to minimize wrt parameters

in general, L is not convex, minima cannot be found analytically, f is not linear, x or y are multivariate

In regression, what is special about MSE loss?

General, formal reason: Maximum Likelihood Estimation

maximize $L = P(\text{data} \mid \text{model})$

model: $P(y \mid x, a, b) = N e^{-\frac{(y-(ax+b))^2}{2\sigma^2}}$

each data pt is iid

$$P(\text{data} \mid \text{model}) = \prod_{i=1}^{N} P(y_i \mid x_i, a, b)$$

$$= \exp(-MSE)$$

usually

$-\log P$

"log likelihood"

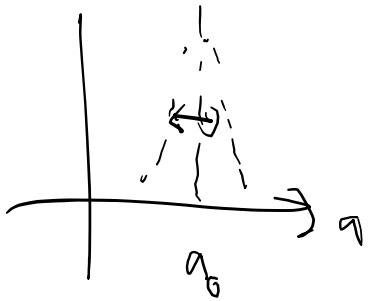
MLE:

1. "consistency" as data $\to \infty$

estimated parameters $\longrightarrow$ true parameters

(a, b)

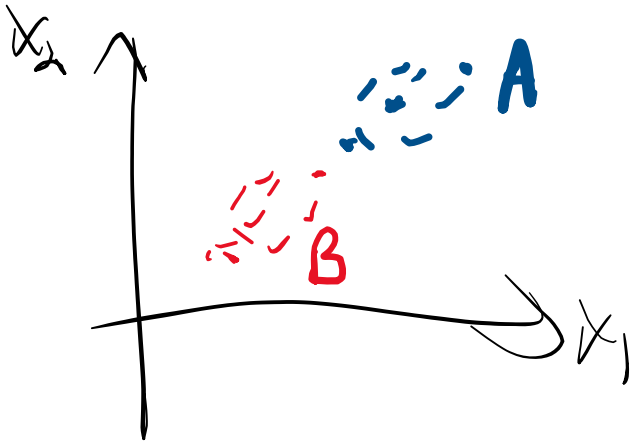
2. "efficiency" as data $\to \infty$ min variance estimator



- (Binary) Classification

iid = independent & identically distributed

- What should loss be under MLE?



$f(x;w)$ = prob. that x is type A.

$= P(A|x) = 1 - P(B|x)$

$$P(\text{data}|\text{model}) = \prod_{i \in A} P(A|x_i) \prod_{j \in B} P(B|x_j)$$

$$= \prod_{i \in A} f(x_i;w) \prod_{j \in B} (1 - f(x_j;w))$$

$$L = -\log P = -\sum_{i \in A} \log f(x_i; w) - \sum_{j \in B} \log \left(1 - f(x_j; w)\right)$$

"log loss"

$$= -\sum_{i \in \text{data}} \left(y_i \log f_i + (1 - y_i) \log (1 - f_i) \right)$$

"binary cross entropy"

$$y_i = \begin{cases} 1 & (A) \\ 0 & (B) \end{cases}$$

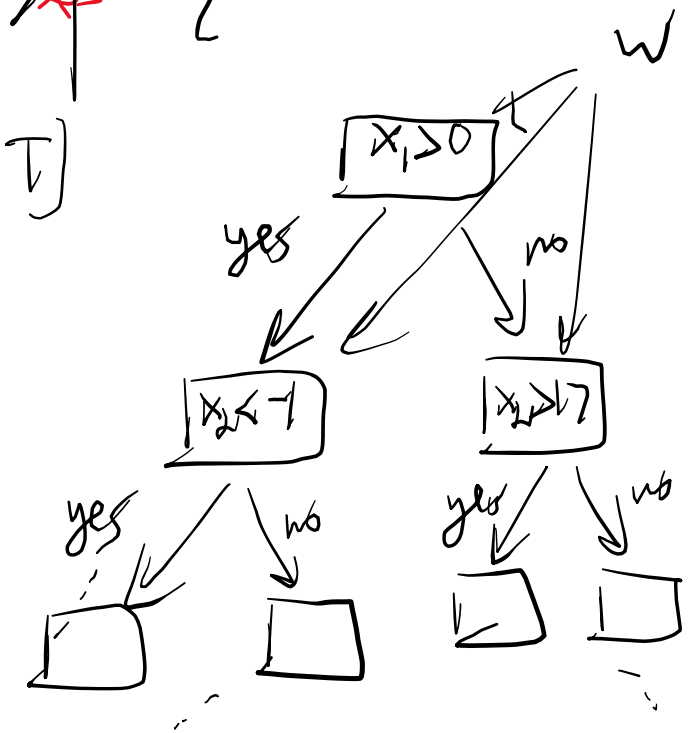
• What family of fns to use? $f(x; w)$

popular options in HEP:

1. $f(x; w) = f_0(x)$ "high level feature"

$f_0(x) = r$

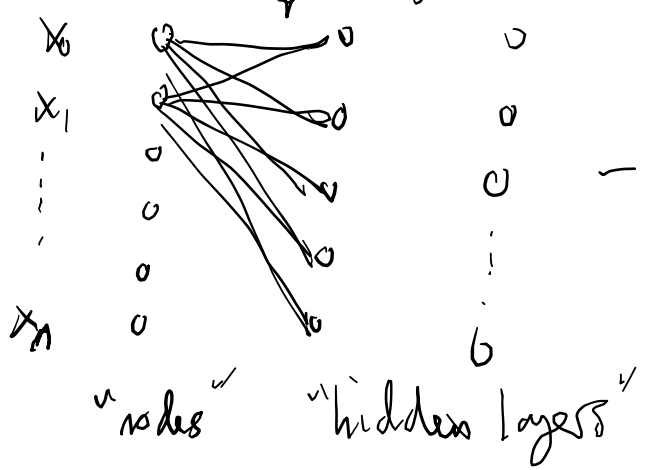
2. (Boosted) Decision Tree (BDT)



3. Neural Networks

- Biologically inspired
- Scale well to higher dimensions
- Expressive ("Universal Approx Thm")
- Generalize unreasonably well.

Feed forward NNs



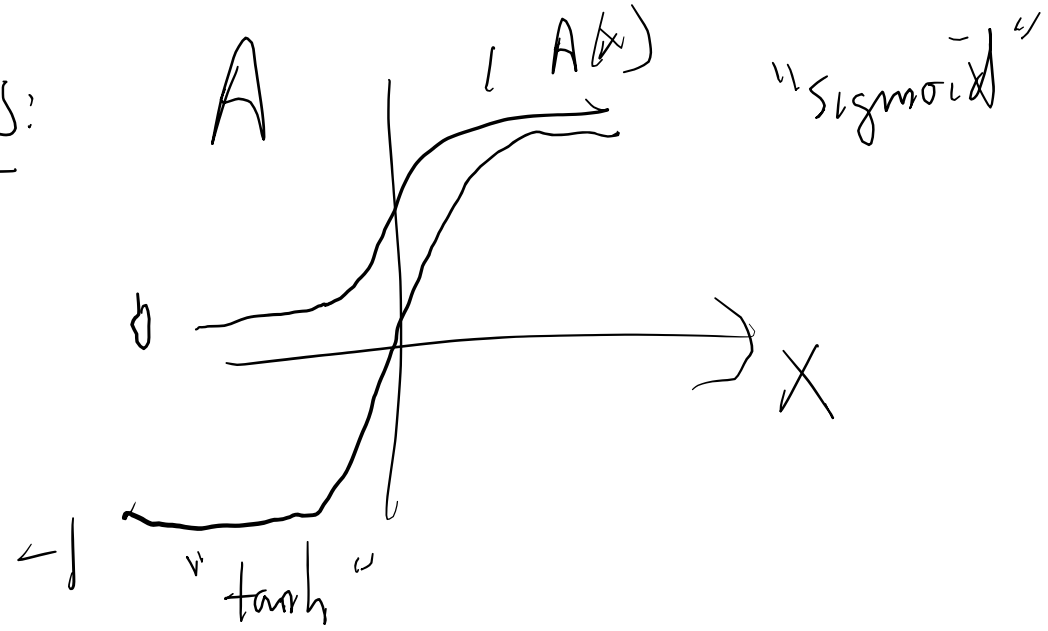
"fully connected"
"dense" NN (DNN)

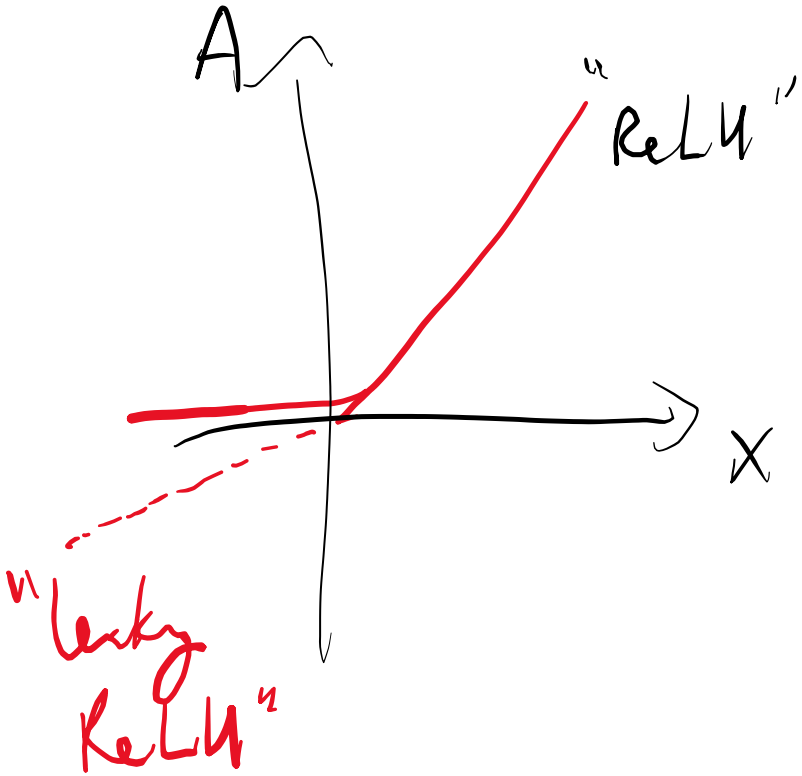
weights biases

$$\vec{x}^{(j)} = A\left([W^{(j)}] \cdot \vec{x}^{(j-1)} + \vec{b}^{(j)}\right)$$

elementwise activation fn.

Examples:





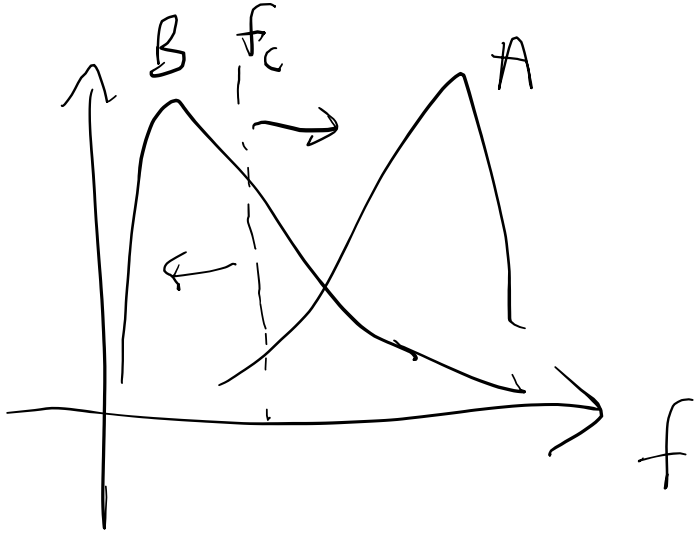
$$A\begin{pmatrix} s \\ x \end{pmatrix} = \begin{pmatrix} A(x_0) \\ A(x_1) \\ \vdots \\ A(x_{n-1}) \end{pmatrix}$$

• Metrics for binary classification

- Suppose we've learned $f(x; w)$. How do we measure/characterize its performance?

1. $\mathcal{L}[f(x; w)]$ loss fn is ultimate metric
   however its not practically useful / interpretable

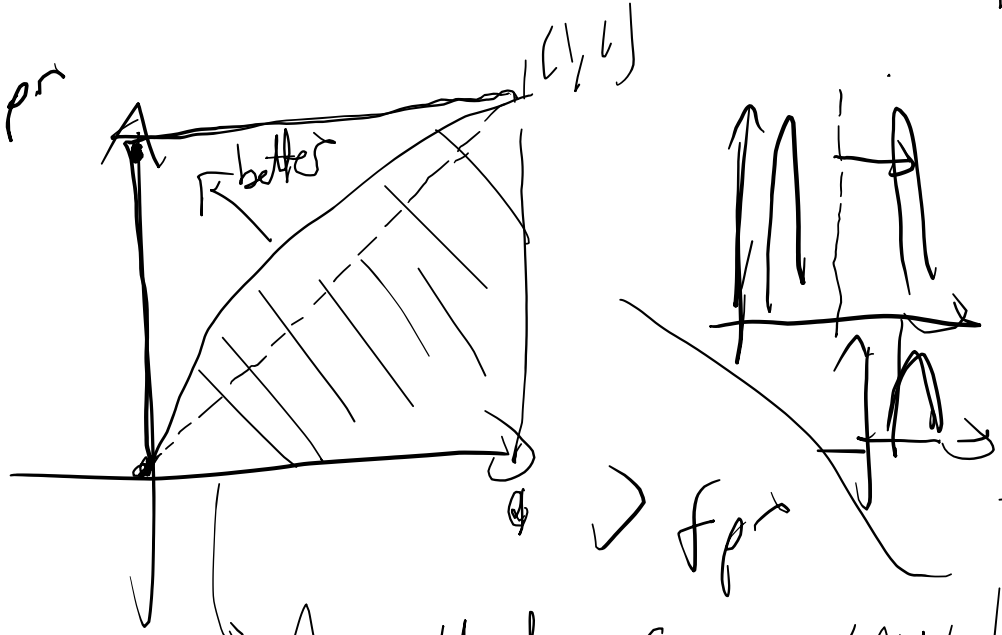
2. $f(x; w) > f_c$ A
   $< f_c$ B
   "working point"

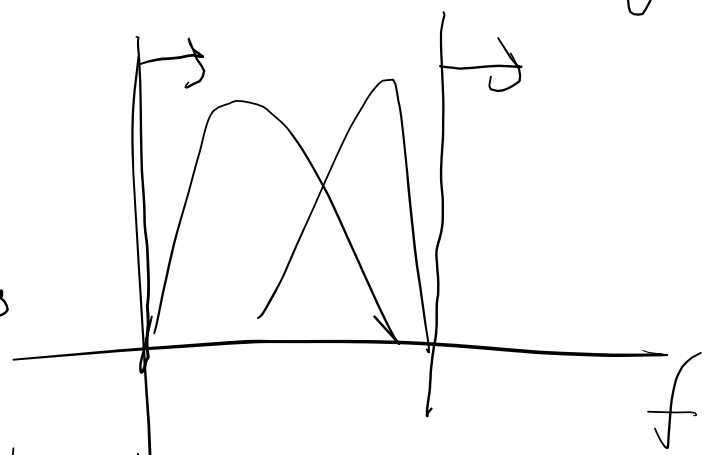


fraction of class A passing cut : "true positive rate" tpr
"signal efficiency"

" " B " " : "false positive rate" fpr
"background efficiency"

"ROC curve"





→ Area Under Curve (AUC)

(0.5 = random guessing)