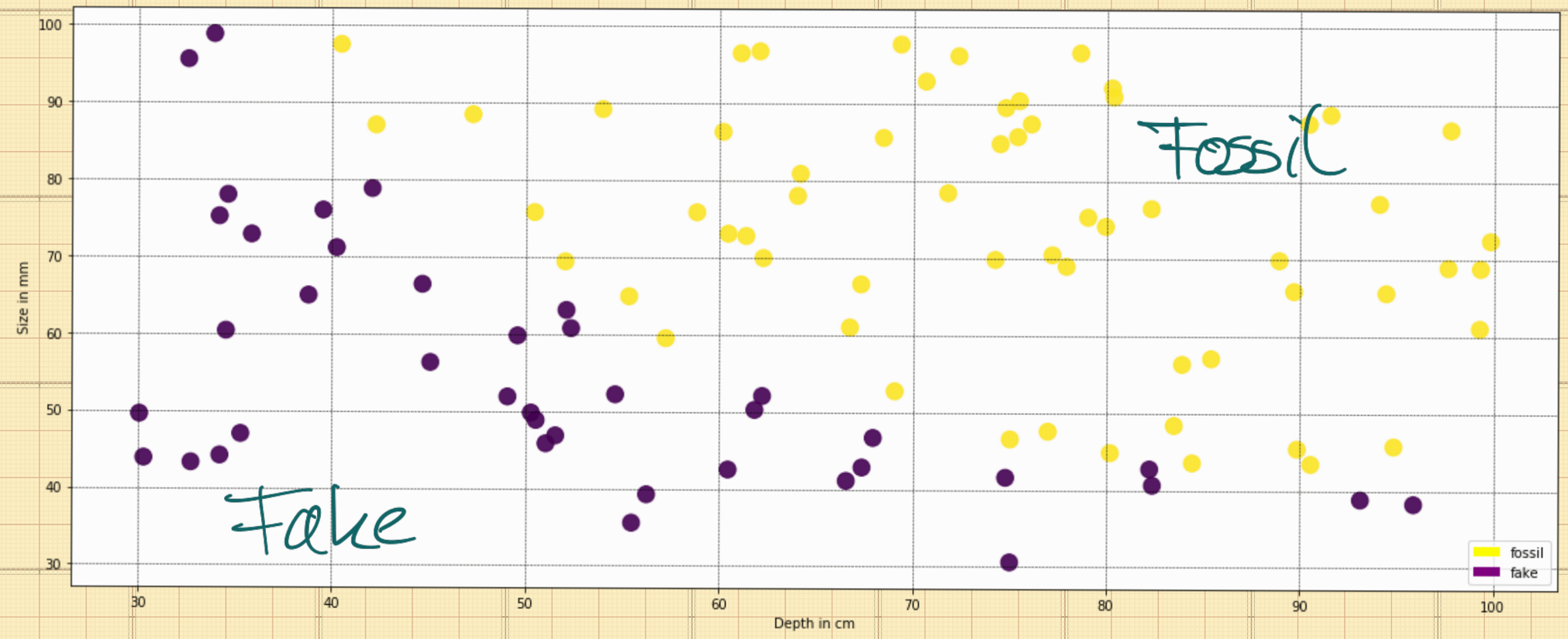
Logistic Regression (a Classifier!)

Lin. Reg: regression; Log. Reg: Classification

→ we consider binary classification: pass/fail; spam/no spam; $y = \{0, 1\}$



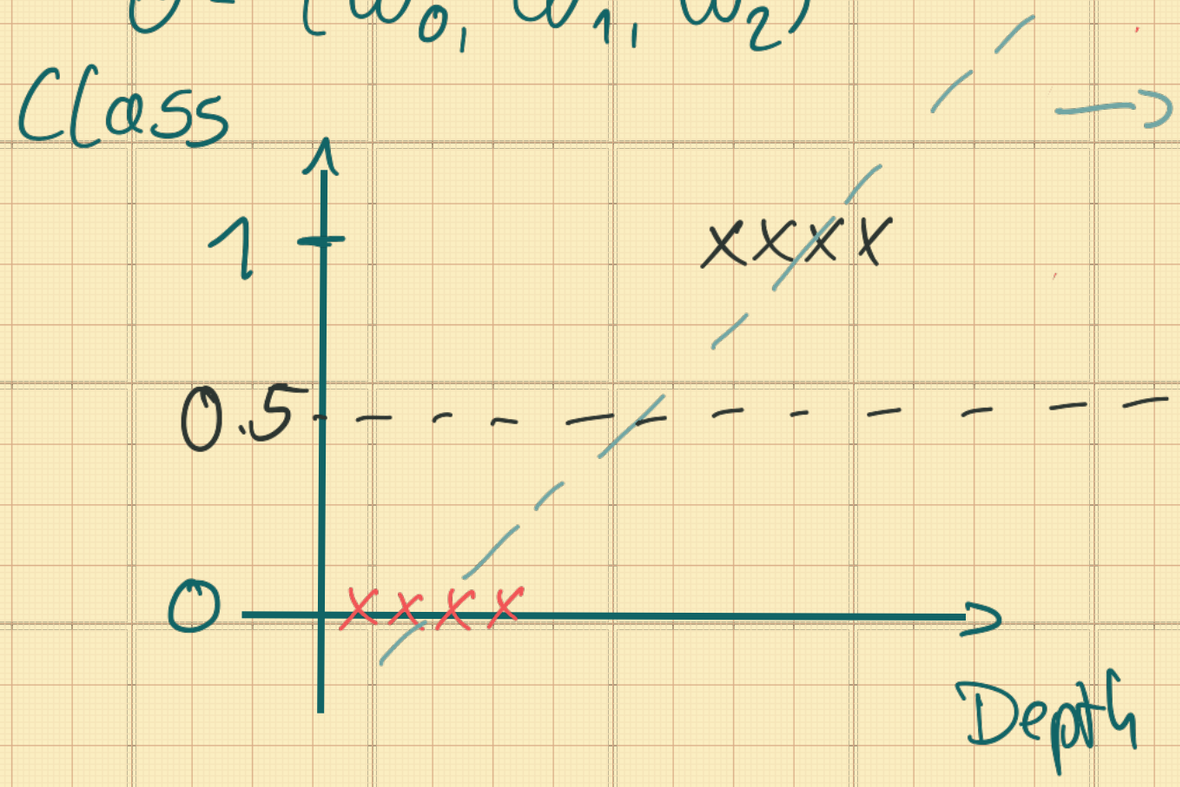
→ train an ML method with inputs x_1: depth, x_2: size to predict 0/1 : fake / fossil

→ let us try the linear regression:

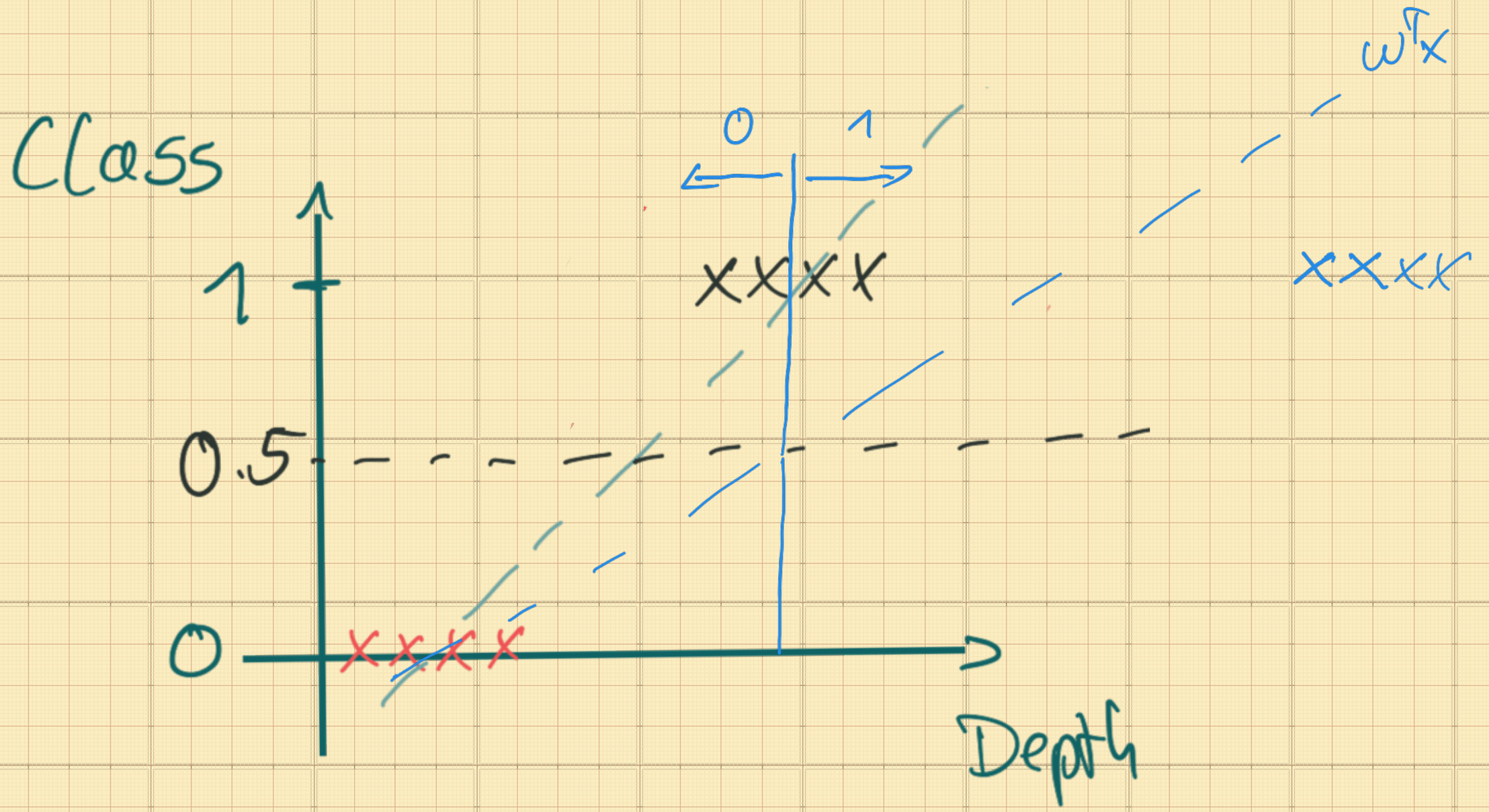
$$h_\theta = w_0 + w_1 x + w_2 x_2$$

$$\theta = (w_0, w_1, w_2)^T$$

if $\hat{y} > 0.5 \rightarrow 1$
if $\hat{y} \leq 0.5 \rightarrow 0$

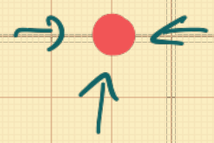


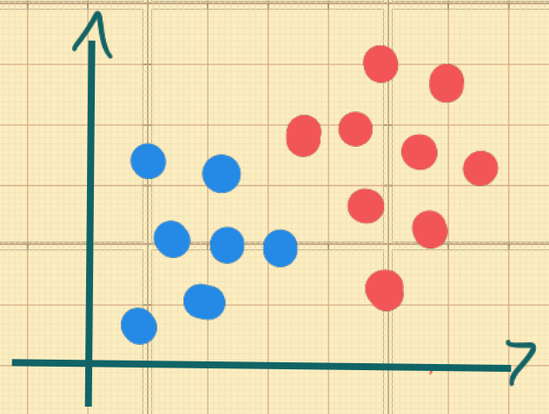
→ Linreg: $h = w^T x$, minimizes MSE


Class
1
0.5
0
Depth
$w^T x$
0
1

→ linear model not well suited for classification!

→ outlies move the optimum strongly





→ output of $x^T w$ is unbounded, not $0 \leq x^T w \leq 1$

→ Probabilistic p.o.v: $p(y \mid x, \theta) = \text{Ber}(y \mid \mu(x))$

→ MLE → cost function: "cross entropy"

→ non-linear w.r.t. θ

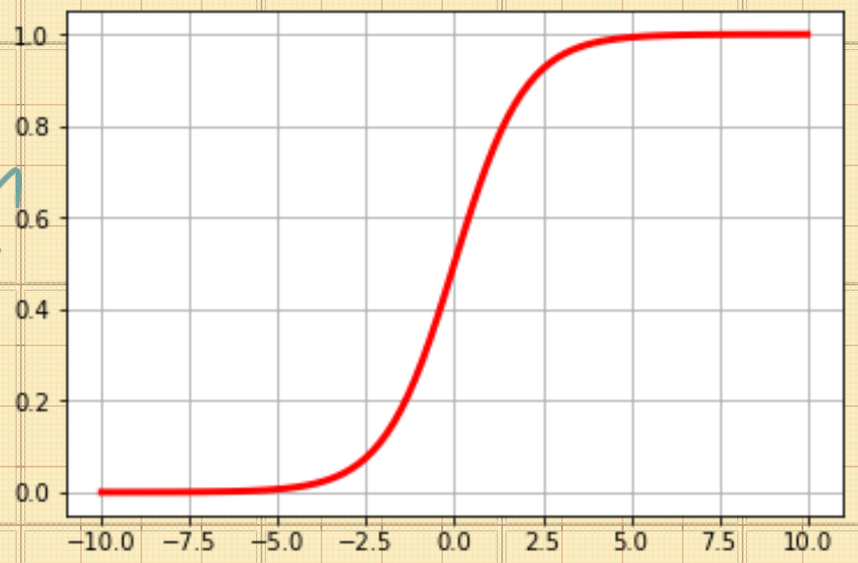
→ optimization more difficult

→ practical approach!

→ extend linear hypothesis by

non-linear activation function

→ "sigmoid" function:



↓ induced by

Bernoulli distr.

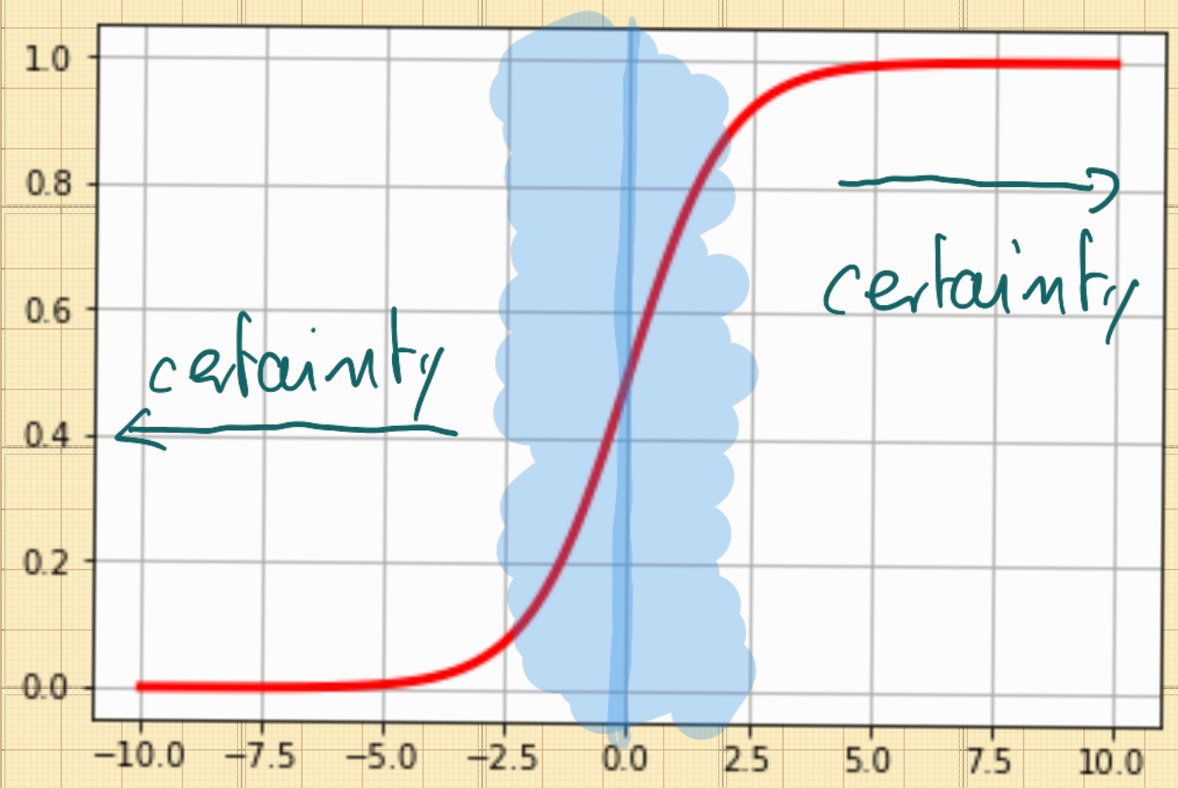
$$G(z) = \frac{1}{1+\exp(-z)}$$

$z \to \infty : G(z) \to 1$

$z \to -\infty : G(z) \to 0$

$$\to \overset{\log}{h_\theta} = G(\overset{\text{lin}}{h_\theta}(x)) = G(w_0 + w_1 x_1 + w_2 x_2) \in (0,1)$$

→ We can now interpret $h_\theta^{\log}(x)$ as the probabilities $p(y=1|x)$: "How likely is class 1?"

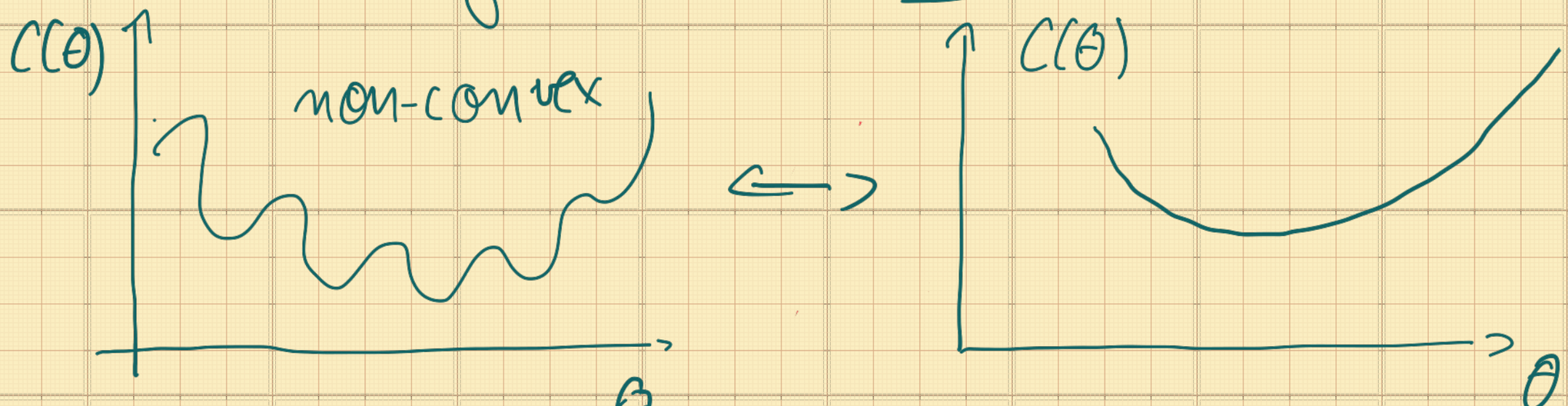


→ note that now all points are equally important in the cost function (outliers do not skew the cost)

$h_\theta^{log}(x) = 0.5$: decision boundary

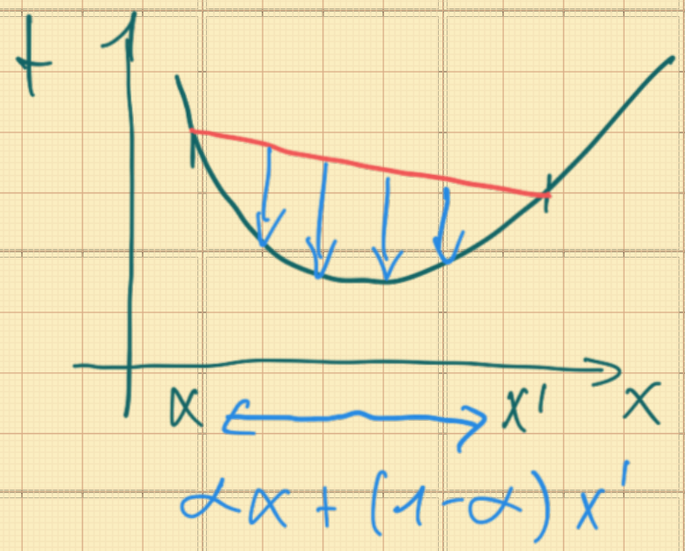
$$C = \frac{-1}{N} \sum_{n=1}^{N} \left[y_i \log(h(x_i)) + (1-y_i) \log(1-h(x_i)) \right]$$

(a MSE cost function is not convex w.r.t. θ)



Def.: $0 \leq \alpha \leq 1$: if $f(\alpha x + (1-\alpha)x') \leq \alpha f(x) + (1-\alpha) f(x')$

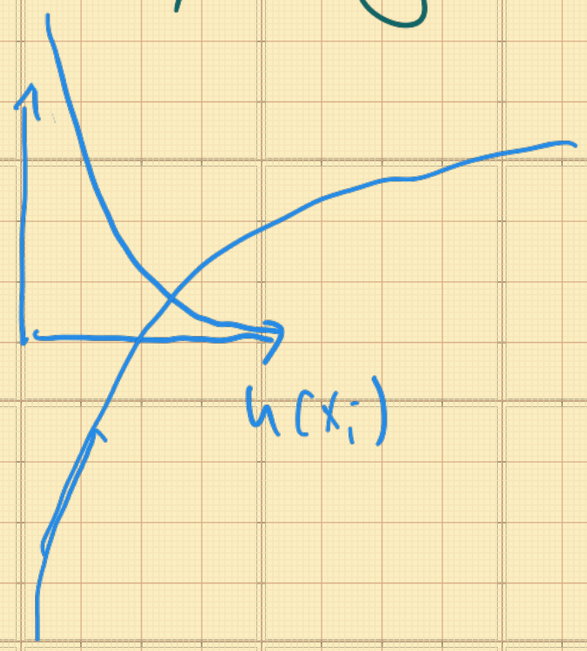
linear blending

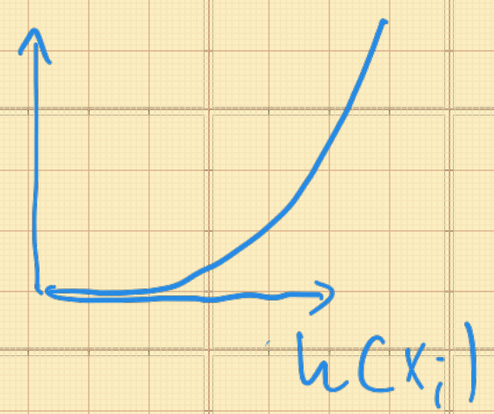


convex

non-convex

$$C' = -y_i \log(h_i) - (1-y_i)\log(1-h_i)$$

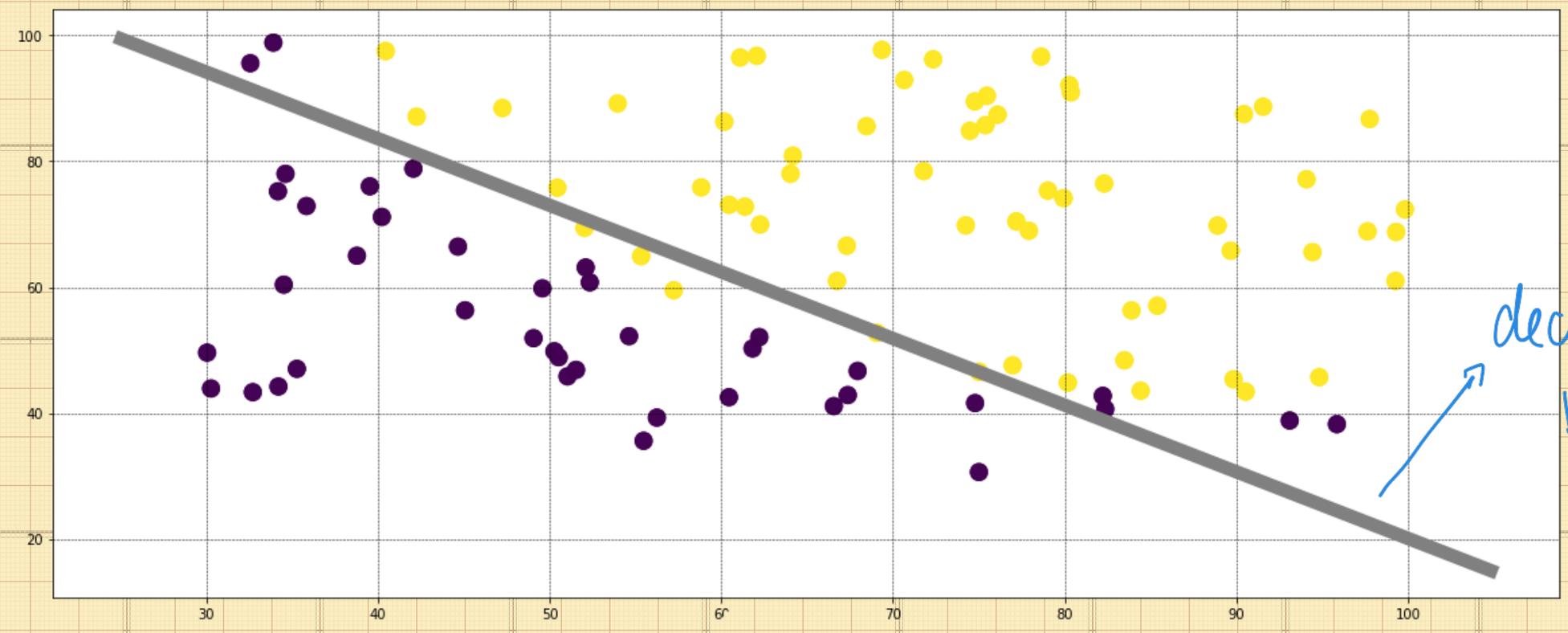




$$\nabla_\theta \mathcal{L}^i = \frac{\partial \mathcal{L}^i}{\partial w_j} = (h_i - y_i)\, x_{ij}$$

$$\text{e.g. } w_1^{new} = w_1^{old} - \frac{LR}{N} \sum_{n=1}^{N} \frac{\partial \mathcal{L}^n}{\partial w_1}$$

j: feature index
$(w_0 x_0 + w_1 x_1 + w_2 x_2)$
i: which sample

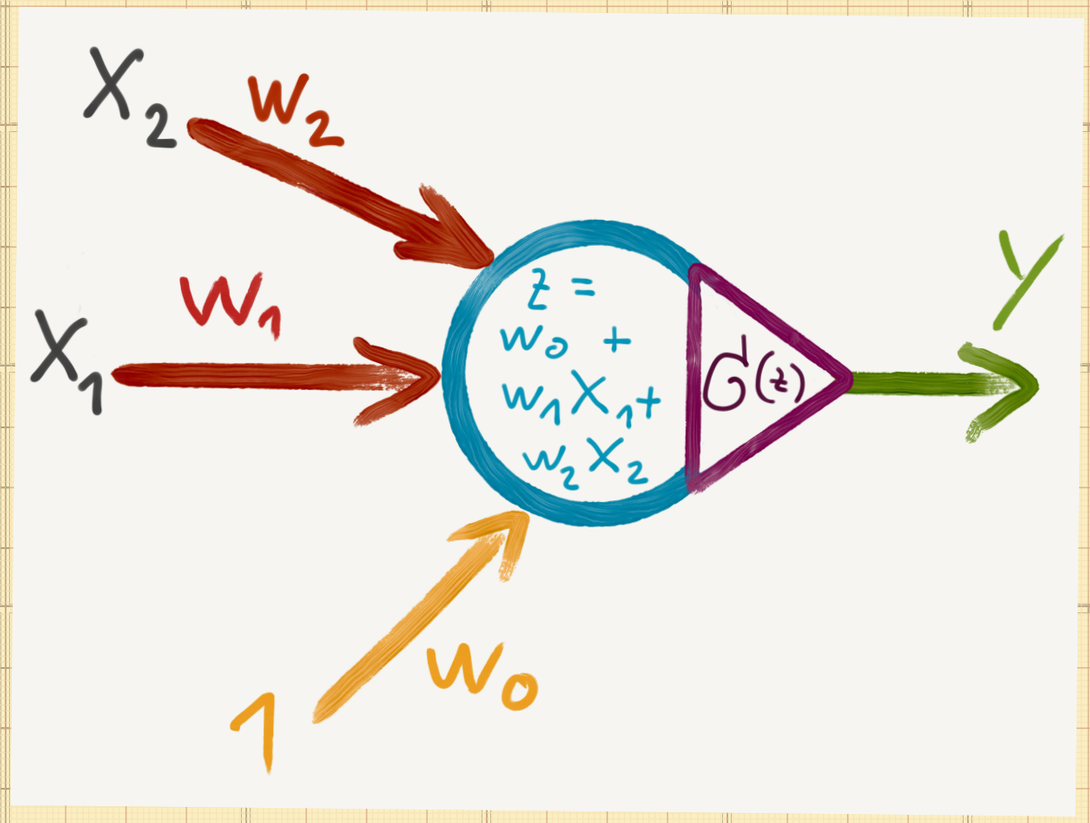


Problem: linearly seperable data

Logistic neuron:

linear neuron

→ activation function



Sigmoid, tanh, ReLU, ...

"Perceptron": Rosenblatt ~~1926~~ 1962 : $\sigma(z) = \begin{cases} 1, & z \geq 0 \\ -1, & z < 0 \end{cases}$

→ Book recommendation:

Pattern recognition & machine learning, Ch. Bishop

Springer