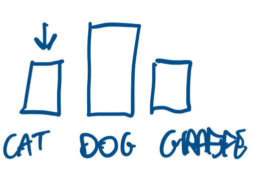
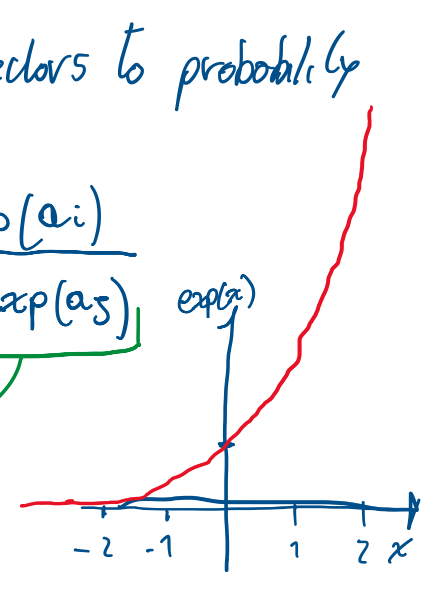
Monday, 28 October 2024 10:04

Generalization for $C > 2$
Multiclass - Classification



Softmax maps vectors to probability simplex.

$$\text{softmax}(a_i) = \frac{\exp(a_i)}{\sum_j \exp(a_j)}$$

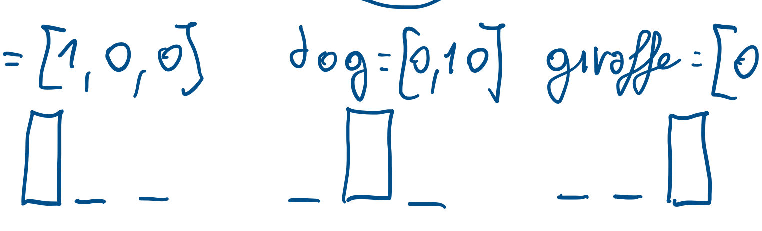


$\exp(a_i)$: Higher a_i get large probability
Lower a_i get scaled down

→ ensure everything sum up to 1

$f_\theta(x_i) = \text{softmax}(\theta^T x_i)$ → Logits

cat $= [1, 0, 0]$ dog $= [0, 1, 0]$ giraffe $= [0, 0, 1]$



Probability distribution with all the mass on a single class.
$f(x)$ encodes the probability distribution over all the classes
$f(x)_i$ represent the probability for the single class i

$$p(Y \mid f(x)) = \prod_i^c [f(x)_i]^{y_i}$$

since Y is one-hot vector only $f(x)_i$ is going to contribute in the product

$Y = [0, 1, 0]$

$f(x)_1^0 \cdot f(x)_2^1 \cdot f(x)_3^0 \cdot$

$1 \cdot f(x)_2 \cdot 1$

$f(x)_2$ → true class

$0 f(x)_1 + 1 f(x)_2 + 0 f(x)_3$

apices : refer to sample in data
pedices : refer to single class in class vector

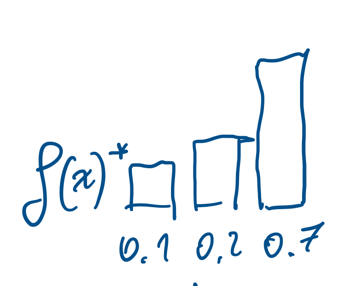
$$p(y^i \mid f(x^i)) = f(x^i)_y$$

$$L = \prod_i^n p(y^i \mid f(x^i))$$

$$\log L = \sum^n \log p(y^i \mid f(x^i)) = \sum^n \log f(x^i)_{y^i}$$

$$NLL = \underbrace{-\sum^n \log f(x^i)_{y^i}}_{\text{Cross Entropy}}$$

$$= -\sum_i^n \sum_j^c y_j^i \log f(x^i)_j$$



$Y =$ cat
$Y =$ giraffe

$Y =$ cat

$$= -\Big(\underbrace{1(-1)}_{\text{cat}} + \underbrace{0(-0.698)}_{\text{dog}} + \underbrace{0(-0.15)}_{\text{giraffe}}\Big)$$

$= 1$

$Y =$ giraffe

$$= -\big(0(-1) + 0(-.69) + 1(-0.15)\big)$$

$= 0.15$

x_1 $\quad Y_1 =$ cat $\quad [1, 0, 0]$ $\quad f(x_1) = [0.4, 0.2, 0.4]$
x_2 $\quad Y_2 =$ dog $\quad [0, 1, 0]$ $\quad f(x_2) = [0.01, 0.98, 0.01]$

$$CE(\theta) = -\sum_i^2 \sum_j^3 y_j^i \log f(x^i)_j$$

$$= x_1 \; -\Big(y_1^1 \log(f(x^1))_1 + y_2^1 \log(f(x^1))_2 + y_3^1 \log(f(x^1))_3 +$$
$$x_2 \; \big(y_1^2 \log(f(x^2))_1 + y_2^2 \log(f(x^2))_2 + y_3^2 \log(f(x^1))_3$$

$$= -\Big(1(-0.91) + 0(-1.6) + 0(-0.91) +$$
$$0(-4.6) + 1(-0.02) + 0(-4.6)\Big)$$

$\log(f(x^1)) = [-0.91, -1.6, -0.91]$
$\log(f(x^2)) = [-4.6, -0.02, -4.6]$

$$= -(-0.91 - 0.02) = 0.93$$

Maximise the probability of the true class at the expense of the other outputs.

Softmax : unstable : $-\log\left(\frac{\exp p_i}{\sum_j \exp p_j}\right)$

$-p_i + \underline{\log \text{sumexp}(p)}$ → $\log\left(\sum_i \exp p_i\right)$

$\log\left(\sum \exp(p_i - c)\right) + c$

avoid 0s messing with our optimization by adding a small constant c
if you set $c = \max(p)$ ensure numerical instabilities will never occur.

$-\frac{1}{2n}(\text{Loss})$

→ used for differentiation of square differences
→ Normalize by number of samples in the dataset

$f(x_i) = \text{softmax}(\theta^T x_i)$ → Multiclass Logistic Regression

best set of parameters obtained by minimizing $CE(\theta)$

Entropy $H(p) = -\sum_i p_i \log p_i$ — measure uncertainty of distribution p

Cross Entropy $CE(p, q) = -\sum_i p_i \log q_i$ — measure uncertainty of distribution p when we encode it with q

KL divergence $KL(p \| q) = +\sum_i p_i \log \frac{p_i}{q_i}$ → how much information is lost when you use q istead of p

$$CE(p, q) = H(p) + KL(p \| q)$$

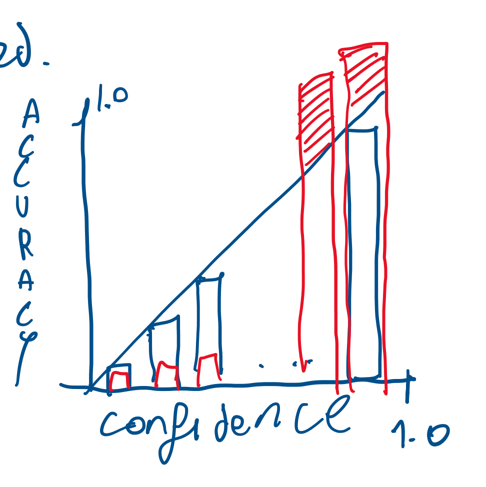
$H(p)$ → uncertainty of true distribution
$KL(p \| q)$ → additional cost of using q instead of p

$f(x)_i$ logits cannot be considered as real probabilities.

$p(Y = i \mid x) = f(x)_i$ unless this happens!

How to measure if a model is calibrated.

1 pick a validation set
2° split $[0, 1]$ into m bins $[1/m]$
B_m : number of sample whose confidence falls into bin m
p_m : average confidence for each bin
a_m : average accuracy for each bin



$$\text{Expected Calibration Error (ECE)} = \sum_m \frac{B_m}{n} |a_m - p_m|$$

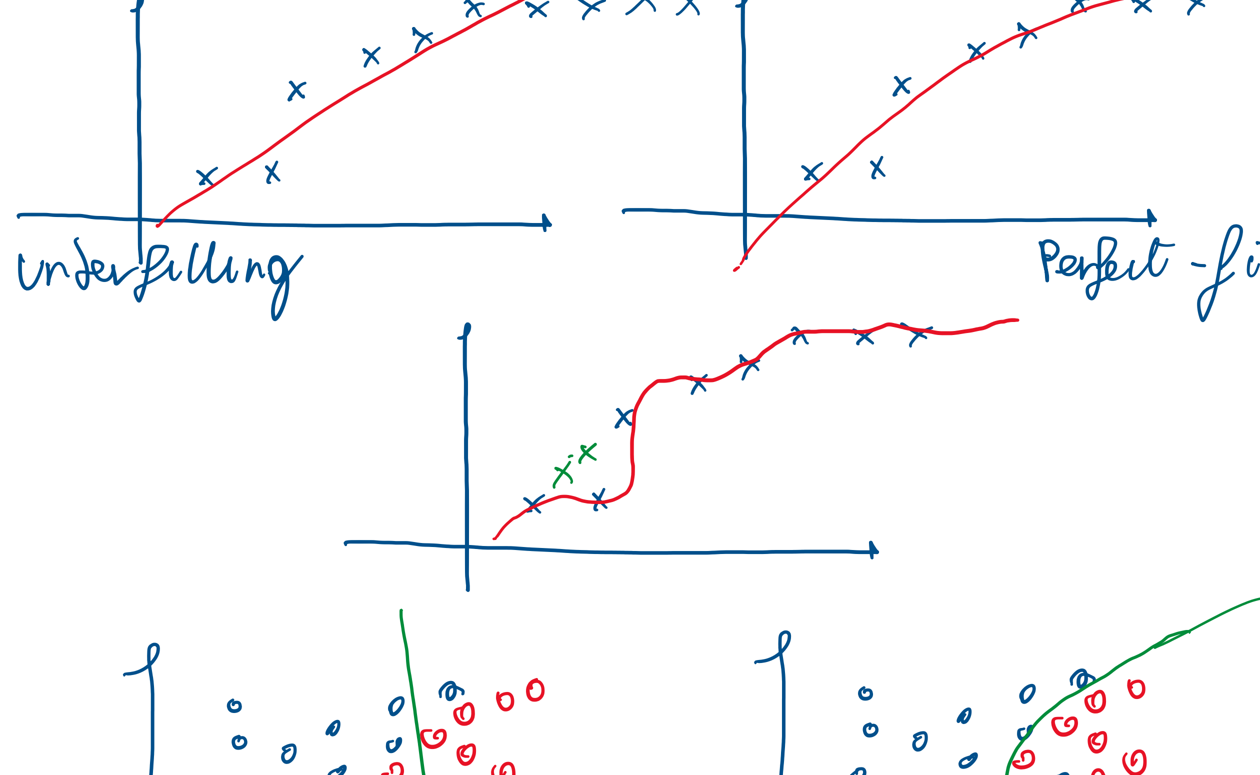
Hyperplanes

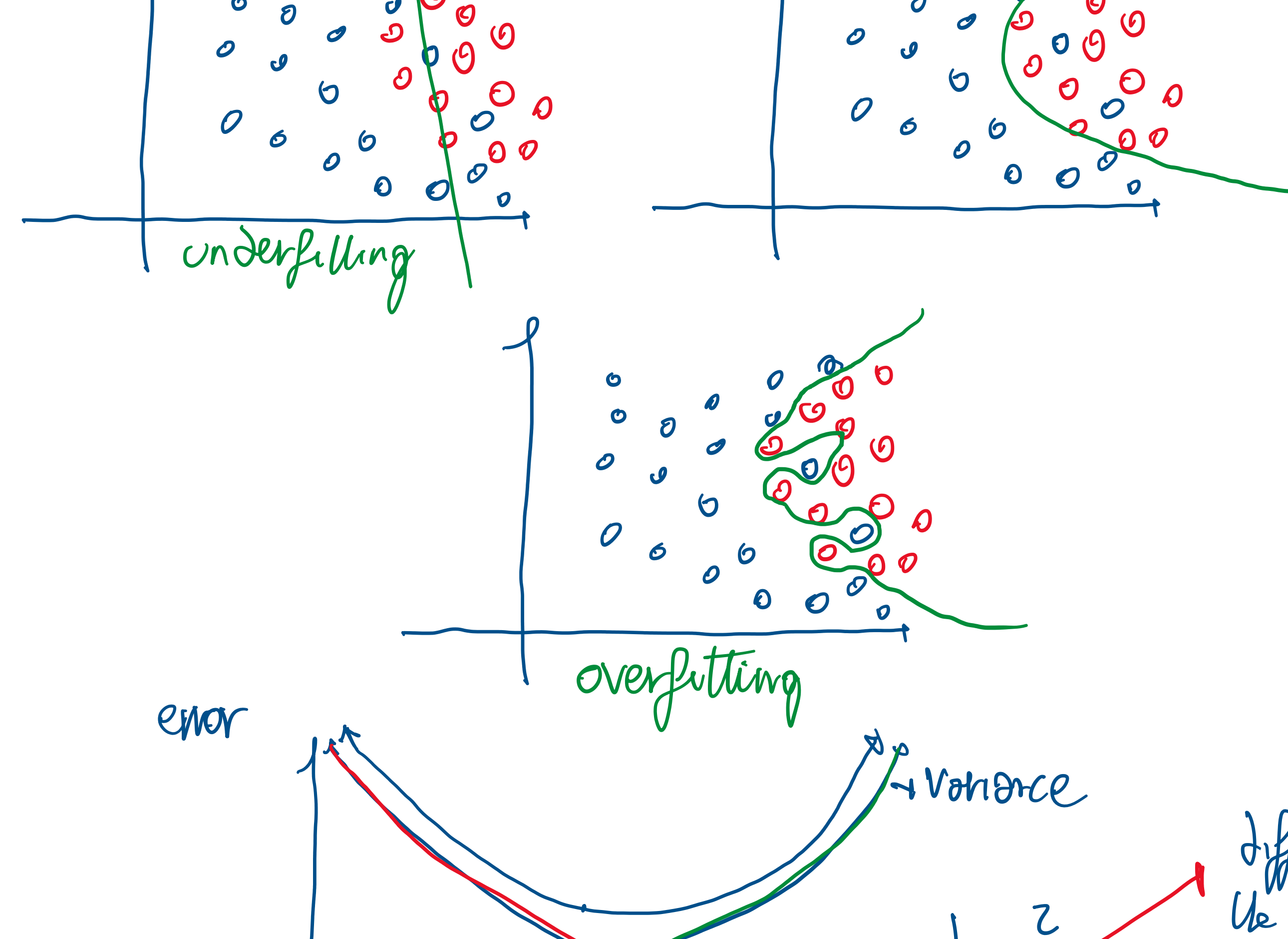
N dimensional space → $N-1$ dimensional flat subspace

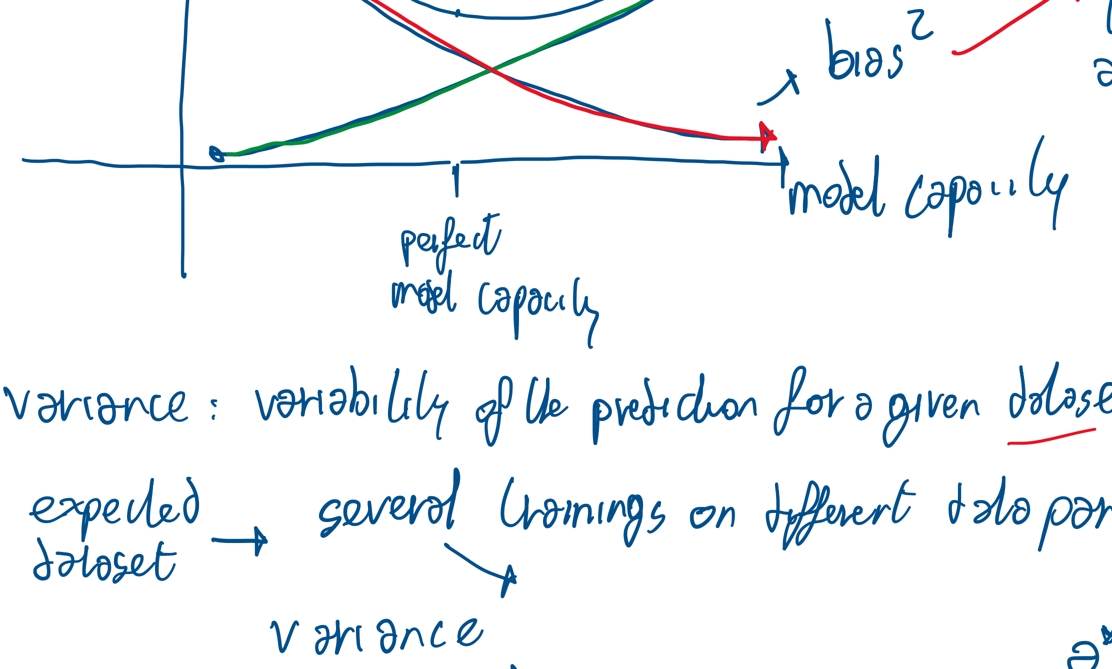
2D → 1D line
3D → 2D plane

1 can only separate classes that are linearly separable
2 distance from closest data sample and hyperplane is called margin

Bias - Variance



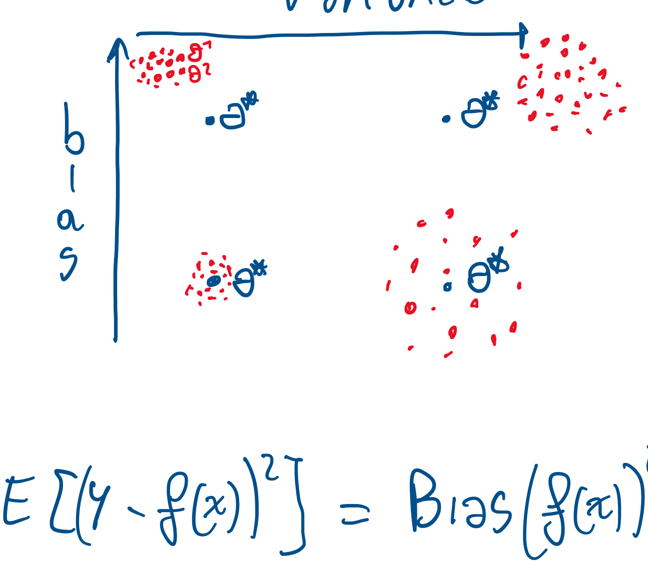




bias² → difference between the expected prediction and the correct one

Variance : variability of the prediction for a given dataset

expected dataset → several trainings on different data partitions



θ^* obtained by minimizing Expected Risk
$\theta^1 \theta^2 \theta^3 \ldots$ obtained by minimizing the Empirical Risk

$$E\left[(Y - f(x))^2\right] = \text{Bias}(f(x))^2 + \text{Variance}(f(x)) + \underbrace{\sigma^2}_{\text{Irriducible error}}$$