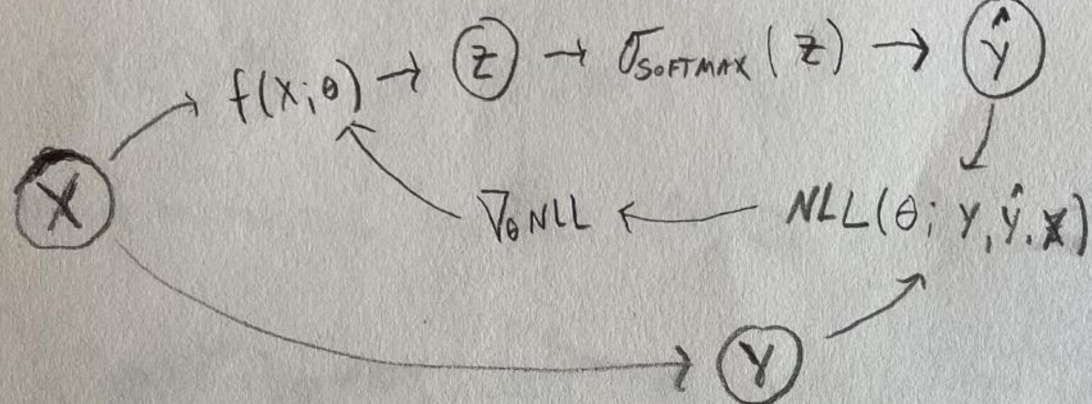
INTRODUCTION TO ANNs

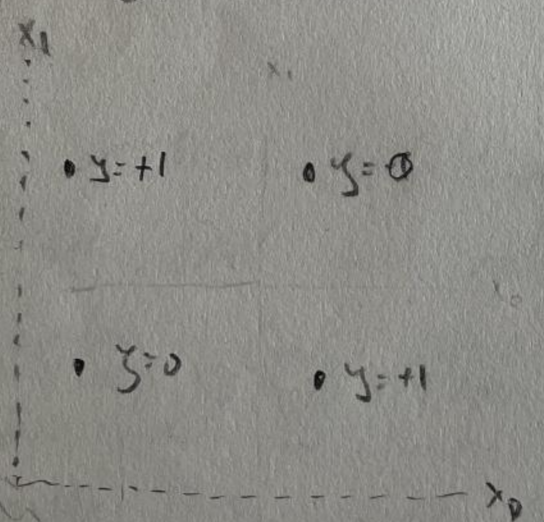
RECALL THE DISCRIMINATIVE MODELING APPROACH FROM LECTURE 03:



SOFTMAX REGRESSION : $f(X;\theta) : XW^T + b$

- CONVEX
- LINEAR IN X
- DECISION BOUNDARY LIES DIRECTLY IN THE SPACE OF THE INPUT X

XOR PROBLEM



$\Rightarrow$ A LINEAR MANIFOLD FAILS TO CLASSIFY X CORRECTLY!

ANN with Single Hidden Layer

$$f(X;\theta) = \sigma_a(X W^{(1)T} + b^{(1)}) W^{(2)T} + b^{(2)}$$

Where

$$\theta = \{ W^{(1)}, b^{(1)}, W^{(2)}, b^{(2)} \}$$

$\sigma_a(\cdot)$ is an "Activation function"

↳ Apply this to the XOR Problem

LET $W^{(1)} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \in \mathbb{R}^{2\times 2}$

$b^{(1)} = [0 \ \ -1] \in \mathbb{R}^2$

$W^{(2)} = [1 \ \ -2] \in \mathbb{R}^2$

$b^{(2)} = 0 \in \mathbb{R}$

$y \in \{0, 1\}$

$$X = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$$

$\sigma_a(z_j) = \max(0, z_j) \Rightarrow$ "Rectified Linear Unit"

"RELU"

ANN / XOR cont...

$$\rightarrow XW^{(1)T} + b^{(1)} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{bmatrix} + \begin{bmatrix} 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix} \rightarrow Z^{(1)}$$

$$\rightarrow \sigma_a(Z^{(1)}) = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix} \rightarrow a^{(1)}$$

$$\rightarrow a^{(1)} W^{(2)T} + b^{(2)} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} + 0 = \overset{\hat{y}}{\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}} \Leftrightarrow \overset{X}{\begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}}$$

↳ CORRECT XOR CLASS ASSIGNMENTS!

$\boxed{\text{SLIDE 10}}$

UNIVERSAL APPROXIMATION THEOREM

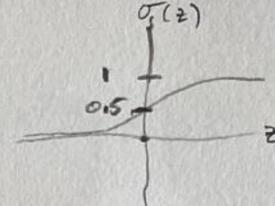
- LINEAR MODELS SUCH AS $XW^T + b$ CAN ONLY MODEL LINEAR FUNCTIONS
- THE ACTIVATION FUNCTIONS ALLOW ANNs TO LEARN NON-LINEAR MANIFOLDS IN THE INPUT SPACE.

⟶ "UNIVERSAL APPROXIMATION THEOREM"
- AN ANN WITH AT LEAST ONE ACTIVATION LAYER CAN APPROXIMATE "ANY" BOREL-MEASURABLE FUNCTION IN FINITE DIMENSIONS GIVEN ENOUGH HIDDEN UNITS. THIS EXTENDS TO THE DERIVATIVES OF THE FUNCTION.

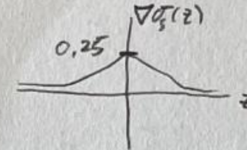
ACTIVATION FUNCTIONS

SIGMOID

$$\sigma_s(z) = \frac{1}{1+e^{-z}}$$

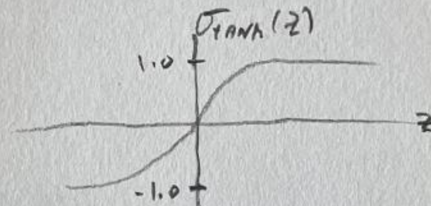


$$\nabla_z \sigma_s(z) = \sigma_s(z)\left(1-\sigma_s(z)\right)$$

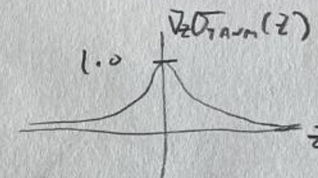


HYPERBOLIC TANGENT

$$\sigma_{TANH}(z) = \frac{2}{1+e^{-2z}} - 1$$

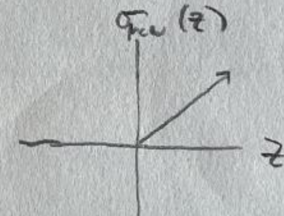


$$\nabla_z \sigma_{TANH}(z) = 1 - \sigma_{TANH}(z)^2$$

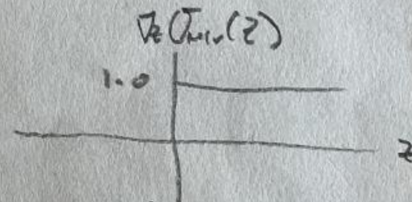


RECTIFIED LINEAR UNIT

$$\sigma_{relu}(z) = \begin{cases} z & z \geq 0 \\ 0 & \text{else} \end{cases}$$

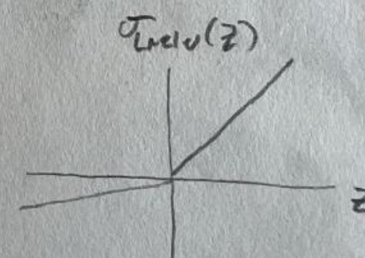


$$\nabla_z \sigma_{relu}(z) = \begin{cases} 1 & z \geq 0 \\ 0 & \text{else} \end{cases}$$

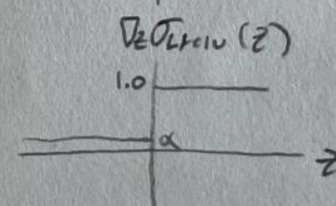


Leaky ReLU

$$\sigma_{Lrelu}(z) = \begin{cases} z & z \geq 0 \\ \alpha & \text{else} \end{cases}$$



$$\nabla_z \sigma_{Lrelu}(z) = \begin{cases} 1 & z \geq 0 \\ \alpha & \text{else} \end{cases}$$



ANN Learning

Recall from lecture 03:

$$\hat{\theta} = \underset{\theta}{\text{argmin}}\ NLL(\theta; \text{DATA})$$

$$= \underset{\theta}{\text{argmin}} -\sum_{i=1}^{M} f(x^{(i)};\theta)_{y^{(i)}} - \log \sum_{k=0}^{K-1} e^{f(x^{(i)};\theta)_k}$$

$\hat{\theta}$ found via gradient descent:

$$\nabla_\theta NLL = -\sum_{i=1}^{M} \nabla_\theta f(x^{(i)};\theta)_{y^{(i)}} - \nabla_\theta \sum_{k=0}^{K-1} e^{f(x^{(i)};\theta)_k}$$

→ Softmax regression: $f(x;\theta) = xW^T + b$

- NLL is convex
- Any local minimum is a global minimum

→ ANN

- ~~NLL is non-convex~~
- ~~Many local minima~~
- $\nabla_\theta f(x;\theta)$ requires successive application of the chain rule, aka back propagation
  - ↳ Lecture 08
- NLL is non-convex, many local minima

STATISTICAL FOUNDATION FOR "REGULARIZATION"

RECALL THAT MLE FINDS $\hat{\theta}_{MLE}$ THAT MAXIMIZES THE LIKELIHOOD OF THE OBSERVED DATA:

$$\hat{\theta}_{MLE} = \underset{\theta}{\text{Argmax}} \sum_{D} P(D;\theta)$$

→ LETS USE BAYES THEOREM TO EXPRESS AN RELATED OBJECTIVE:

$$\underset{\text{R.V.}}{P(\underline{\theta} \mid D^{(i)})} \leftarrow L(D;\theta) \triangleleft\!- P(X,Y) -\!\triangleright L(\theta;D) \rightarrow \underset{\text{NOT R.V.}}{P(D;\underline{\theta})}$$

↳ NEW OBJECTIVE

$$\hat{\theta}_{MAP} = \underset{\theta}{\text{Argmin}} - \sum_{D} \log P(\theta \mid D)$$

$$= \underset{\theta}{\text{Argmin}} - \sum_{D} \log \left[\frac{P(D \mid \theta) P(\theta)}{P(D)} \right]$$

$$= \underset{\theta}{\text{Argmin}} - \sum_{D} \log P(D;\theta) + \log P(\theta) - \log P(D)$$

$$= \underset{\theta}{\text{Argmin}} - \sum_{D} \log P(D;\theta) + \log P(\theta)$$

2 IMPORTANT THINGS:

① WE STARTED BY DESCRIBING θ AS A R.V. IN $L(D;\theta)$, BUT HAVE REDUCED THE PROBLEM TO WHERE θ REPRESENTS A POINT ESTIMATE!

② THE DIFFERENCE BETWEEN $\hat{\theta}_{MAP}$ AND $\hat{\theta}_{MLE}$ LIES IN THE SECOND TERM

REGULARIZATION CONT...

Look at common parameterizations for $P(\theta)$

① $\theta \sim \text{unif}(\lambda)$

$$\hat{\theta}_{MAP} = \underset{\theta}{\text{argmin}} - \sum_{D} \log P(D;\theta) + \log \text{unif}(\lambda)$$

$$= \underset{\theta}{\text{argmin}} - \sum_{D} \log P(D;\theta)$$

$$= \hat{\theta}_{MLE}$$

Thus, MAP is a generalization of MLE

② $\theta \sim N(0, \sigma_\theta^2)$ - Gaussian prior on θ

$$\hat{\theta}_{MAP} = \underset{\theta}{\text{argmin}} - \sum_{D} \log P(D;\theta) + \log \left[\frac{1}{\sqrt{2\pi\sigma_\theta^2}} e^{-\frac{\theta^2}{2\sigma_\theta}} \right]$$

$$= \underset{\theta}{\text{argmin}} - \sum_{D} \log P(D;\theta) - \underline{\underline{\frac{\theta^2}{2\sigma_\theta}}}$$

$\boxed{L_2 \text{ REGULARIZATION}}$

③ $\theta \sim \text{Laplace}(0, \sigma_\theta)$

$$\hat{\theta}_{MAP} = \underset{\theta}{\text{argmin}} - \sum_{D} \log P(D;\theta) + \log \left[\frac{1}{2\sigma} e^{-\frac{|\theta|}{\sigma_\theta}} \right]$$

$$= \underset{\theta}{\text{argmin}} - \sum_{D} \log P(D;\theta) - \underline{\underline{\frac{|\theta|}{\sigma_\theta}}}$$

$\boxed{L_1 \text{ REGULARIZATION}}$

REGULARIZATION CONT..

DROPOUT

RANDOMLY SELECTS NODES IN EACH HIDDEN LAYER, ℓ, AND SETS THEM TO ZERO DURING TRAINING.

THIS IS AN ELEMENT-WISE OPERATION:

$$a'^{(\ell)} = \gamma\, M \odot a^{(\ell)}$$

Where

$$M_j \sim \begin{cases} 0 & \text{with } P = P_{DROPOUT} \\ 1 & \text{with } P = 1 - P_{DROPOUT} \end{cases},$$

γ IS A CONST FUNCTION OF $P_{DROPOUT}$

THUS:

DURING TRAINING: $z_j^{(\ell+1)} = a'^{(\ell)} W_j^{T(\ell+1)}$

DURING INFERENCE: $z_j^{(\ell+1)} = a^{(\ell)} W_j^{T(\ell+1)}$

GRADIENT MOMENTUM

SGD W/ SIMPLE MOMENTUM

SET MOMENTUM α
SET LEARNING RATE η

INITIALIZE θ, VELOCITY V

REPEAT:

$X, y \sim$ DATA MINIBATCH SZ m

∇_θ FROM $\frac{1}{m} \sum_{i=1}^{m} LL(x^{(i)}, y^{(i)}, \theta)$

$V = \alpha V - \eta \nabla_\theta$

$\theta = \theta + V$

UNTIL: STOPPING CONDITION

POPULAR VARIANTS

- NESTEROV
- RMS PROP
- ADAM