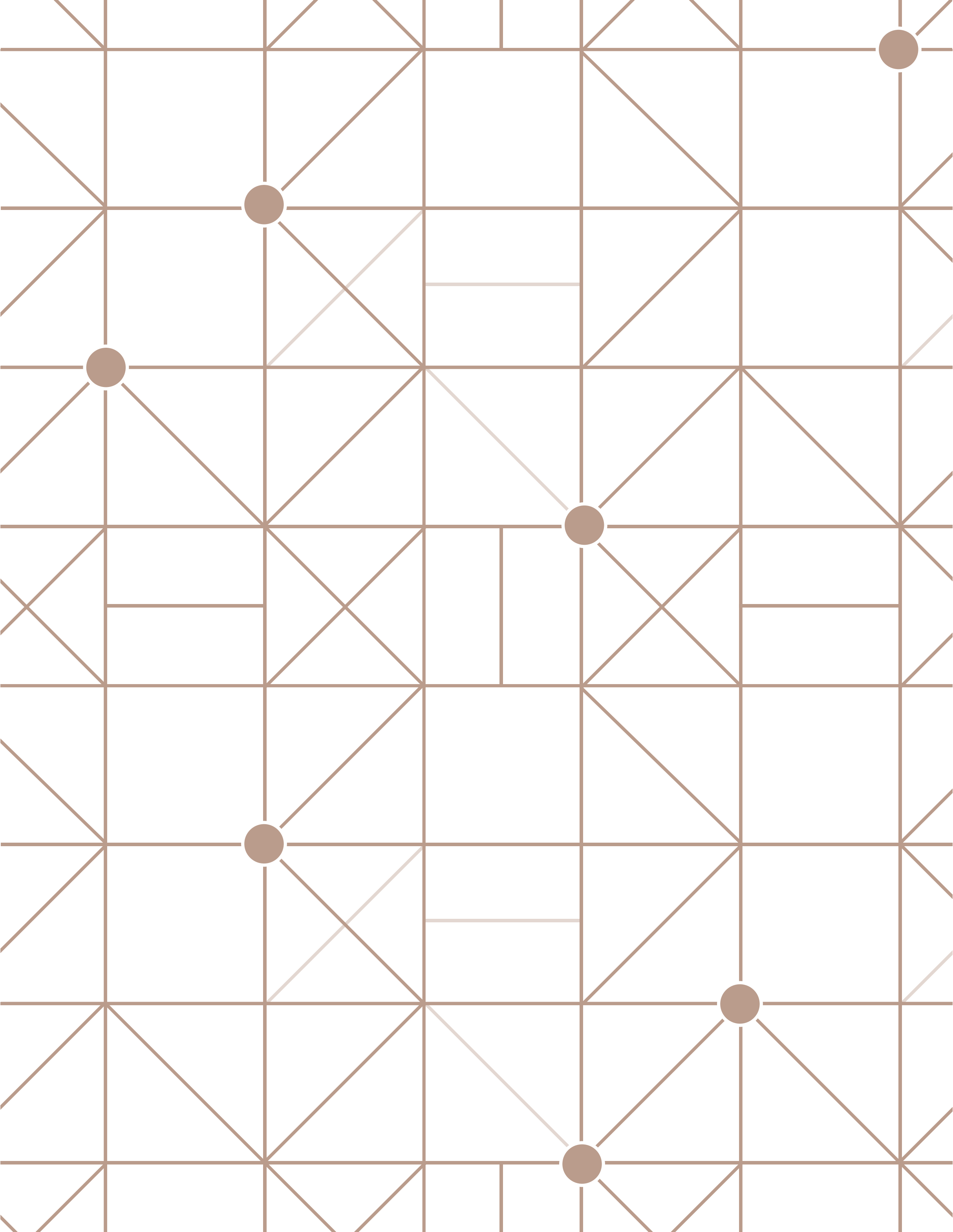


MOCK QUIZ 2

Q1. Prove that sigmoid is a special case of softmax –

i). take a 2 class classification problem:

$$\text{— last layer} \longrightarrow \begin{matrix} a_1 \\ a_2 \end{matrix} \text{ — softmax} \longrightarrow \begin{cases} \dfrac{e^{a_1}}{e^{a_1}+e^{a_2}} \\[2ex] \dfrac{e^{a_2}}{e^{a_1}+e^{a_2}} \end{cases}$$

$$\text{Prob(class 1)} = \frac{e^{a_1}}{e^{a_1}+e^{a_2}} = \frac{1}{1+e^{a_2-a_1}}$$

$$\text{Prob(class 2)} = \frac{e^{a_2}}{e^{a_1}+e^{a_2}} = \frac{e^{a_2-a_1}}{1+e^{a_2-a_1}}$$

let $a_1 - a_2 = x$

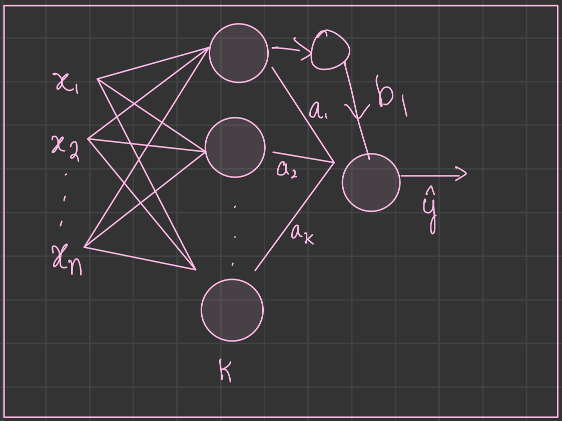
$$\Rightarrow \boxed{\text{Prob(class 1)} = \frac{1}{1+e^{-x}}}$$

sigmoid function

$$\text{Prob(class 2)} = \frac{e^{-x}}{1+e^{-x}}$$

Q2. A Deep NN without non-linear activations is still equivalent to Linear regression

Consider the following neural network with 2 layers:



at the i^{th} neuron in the first layer:

$$x_1 w^1_{i1} + x_2 w^1_{i2} + \dots + x_n w^1_{in} = a_1$$

$$\Rightarrow \boxed{a_i = \sum_{j=1}^{n} x_j w^1_{ij}} \longrightarrow 1.$$

$$\hat{y} = \sum_{i=1}^{K} w^2_i a_i \longrightarrow 2.$$

using 1 in 2,

$$\hat{y} = \sum_{i=1}^{K} w^2_i \sum_{j=1}^{n} w^1_{ij} x_j$$

$$b_i = w^4_i a_i + b$$

$$\hat{y} = \sum_{i=1}^{K} w^2_i b_i$$

$$y = \sum_{i=1}^{K} w^2_i (w^4_i a_i + b)$$

$$= \sum w^2_i w^4_i a_i + w^2_i b$$

$$\hat{y} = \sum_{j=1}^{n} \sum_{i=1}^{K} w_i^2 w_{ij}^1 x_j$$

$$= \sum w_i^2 w_i^4 \left(\sum_{j=1}^{n} x_j w_{ij}^1 \right) + w_i^2 b$$

$$= \sum_{j=1}^{n} x_j \underbrace{\sum_{i=1}^{K} w_i^2 w_{ij}^1}$$

↳ can be written in terms of w^3

$$= \sum_{j=1}^{n} x_j w_j^3 \quad \because \boxed{\text{where } w_j^3 = \sum_{i=1}^{K} w_i^2 w_{ij}^1}$$

This is equivalent to a linear reg. problem.
We can say that by induction, it will work for multiple layers.

Q3. Domain of sigmoid, tanh. Write tanh in terms of sigmoid.

i). $\text{Sigmoid}(x) = \dfrac{1}{1+e^{-x}}$

→ sigmoid:
domain : $(-\infty, \infty)$
Range : $(0,1)$

ii). $\tanh(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$

→ tanh:
domain : $(-\infty, \infty)$
Range : $(-1,1)$

tanh in terms of sigmoid:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^x}{e^x + e^{-x}} - \frac{e^{-x}}{e^x + e^{-x}}$$

$$= \frac{1}{1+e^{-2x}} - \frac{1}{e^{2x}+1}$$

$\because$ division by e^x, e^{-x} respectively

$$= \boxed{\text{sigmoid}(2x) - \text{sigmoid}(-2x)}$$

4. Input size (32x32x6) {image}, train for 100 class classification

i) Layer 1 → 200 neurons, Relu
ii) Layer 2 → 120 neurons,
iii) 100 class classification.

$\because$ Number of params$_{(\text{layer } i)}$

$$= (\underbrace{N_{i-1} + 1}_{\text{Bias.}}) N_i$$

Steps in the forward pass:

i). flatten the image, each image has $32 \times 32 \times 6 = 6144$ data points.

ii). layer 1: 200 neuron:

each neuron will have 6144 parameters along with a bais term.

Thus each neuron has: 6145 parameters

$\Rightarrow$ Layer 1 has $200 \times 6145 = \underline{1229000}$ parameters.

iii). Relu does not have any parameters.

iii). Layer 2: 120 neurons

→ input to this layer will have a size of 200

→ Thus, each neuron will have 201 params (with bais)

→ layer 2 has $120 \times 201 = \underline{24120}$ params.

iv). Output Layer: 100 neurons (100 class classification).

→ input will be of size 120, each neuron will have 121 params.

→ Layer 3 has $100 \times 121 = \underline{12100}$ params.

$\Rightarrow$ total number of params $= 12100 + 24120 + 1229000$

$$\boxed{\text{total params} = 24{,}94{,}220}$$

Q5. Derive the vectorized form of gradient descent for logistic reg.

$$\hat{y} = \frac{1}{1+e^{-(x^T\theta+b)}} \quad \because \text{sigmoid.}$$

$$\sigma(x^T\theta + b)$$

Binary Classification Case

loss function :

$$L(\hat{y},y) = -y\log\hat{y} - (1-y)\log(1-\hat{y})$$

Now $\frac{dL}{d\theta} = \frac{dL}{d\hat{y}} \times \frac{d\hat{y}}{d\theta}$ —— 1.

i). $\frac{dL}{d\hat{y}} = \frac{-y}{\hat{y}} - \frac{(1-y)(-1)}{1-\hat{y}}$

$$= \frac{(1-y)}{(1-\hat{y})} - \frac{y}{\hat{y}}$$

$\because \hat{y}$ is not a vector in this case

$$\boxed{\frac{dL}{d\hat{y}} = \frac{(1-y)}{(1-\hat{y})} - \frac{y}{\hat{y}}} \quad \text{—— 2.}$$

$$\frac{d\sigma(a)}{da} = \sigma(a) \times (1-\sigma(a))$$

$\because \sigma' = \sigma(1-\sigma)$

ii). $\frac{d\hat{y}}{d\theta} = \frac{d(\sigma(x^T\theta+b))}{d\theta} = \hat{y}(1-\hat{y}) \cdot \frac{d(x^T\theta+b)}{d\theta}$

$$= \hat{y}(1-\hat{y})\frac{d(\theta^T x+b)}{d\theta}$$

$$= \hat{y}(1-\hat{y})x \quad \text{—— 3.}$$

$$\frac{d(\theta^T x)}{d\theta} = x$$

Sub 3,2 in 1 :

$$\frac{dL}{d\theta} = \hat{y}(1-\hat{y})x \times \left\{ \frac{(1-y)}{1-\hat{y}} - \frac{y}{\hat{y}} \right\}$$

$$= \cancel{\hat{y}(1-\hat{y})}\, x \left(\frac{\hat{y} - y\hat{y} - y + y\hat{y}}{\cancel{\hat{y}(1-\hat{y})}} \right)$$

$$\boxed{\frac{dL}{d\theta} = x(\hat{y} - y)}$$

With the entire data set

X -> complete dataste

$$\underbrace{\hat{y}}_{\text{vector}} = \sigma(X\theta)$$

loss function $= J(\theta) = \frac{1}{m}\sum_{i=1}^{m} -y_i \log \hat{y}_i - (1-y_i)\log(1-\hat{y}_i)$

$$\Rightarrow \frac{dL}{d\theta} = \frac{1}{m}\sum_{i=1}^{m} (\hat{y}_i - y_i)\left(X[i,:]\right)^T$$

$$\boxed{\begin{aligned} J(\theta) &= \frac{1}{m}\sum_{i=1}^{m} L_i(\theta) \\ \frac{dJ(\theta)}{d\theta} &= \frac{1}{m}\sum_{i=1}^{m} \sum L_i(\theta) \end{aligned}}$$

$$\hat{y} = \frac{1}{1 + e^{-(x^T\theta + b)}} \quad \because \text{sigmoid.}$$

$$\sigma(x^T\theta + b)$$

loss function :

$$L(\hat{y}, y) = \underline{-y \log \hat{y} - (1-y)\log(1-\hat{y})}$$

Now $\frac{dL}{d\theta} = \frac{dL}{d\hat{y}} \times \frac{d\hat{y}}{d\theta}$ ——— 1.

$$\frac{dL}{db} = \frac{dL}{d\hat{y}} \times \frac{d\hat{y}}{db}$$

$$\frac{d\hat{y}}{db} = \frac{d(\sigma(x^T\theta + b))}{db}$$

$$= \hat{y}(1-\hat{y}) \times \frac{d(x^T\theta + b)}{db}$$

$$= \hat{y}(1-\hat{y})$$

$$\frac{dL}{db} = \hat{y}(1-\hat{y}) \times \frac{dL}{d\hat{y}} = \hat{y}(1-\hat{y}) = \cancel{\hat{y}(1-\hat{y})} \times \frac{\hat{y} - y}{\cancel{\hat{y}(1-\hat{y})}}$$

$$= \hat{y} - y$$

$$\frac{dL}{db} = \hat{y} - y$$

$$\boxed{\begin{aligned} \frac{dL}{db} &= \sum \frac{dL_i}{db} \\ &= \sum_{i=1}^{m} \hat{y}_i - y_i \end{aligned}}$$

for the complete dataset

Q.6.]

$$\text{Entropy} = -\sum_{i=1}^{K} p_i \log(p_i)$$

no cross terms

$$\text{Entropy (for 2 classes)} = -p_1 \log p_1 - p_2 \log p_2$$

Decision Trees

$$\text{Cross-Entropy} = -\sum_{i=1}^{K} y_i \log(\hat{y}_i)$$

cross terms

$$\text{(for 2 classes)} = -y \log \hat{y} - (1-y) \log(1-\hat{y})$$

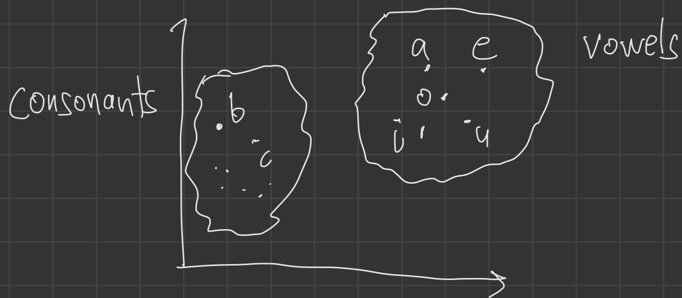
Logistic Regression

Q.7] ① Extremely sparse embeddings for large vocabularies (wasteful)

$$a = \underbrace{[1\ 0\ 0\ 0\ 0\ \ldots\ldots\ldots]}_{|V|}$$

② • 1-hot encodings make each embedding orthogonal (independent and unrelated)

• We ideally want embeddings which capture the semantics / usage patterns too.



How do we solve this ???

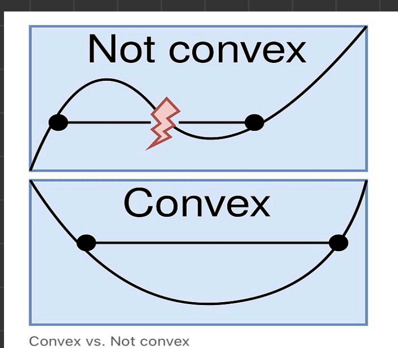
Q.8.] $J(\theta) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$

$\hat{y}_i = \sigma(\theta^T x)$

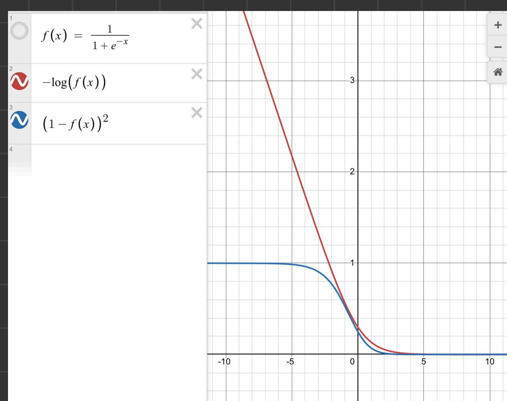
Why don't we use Squared Error ?

① Squared Error is not convex.

(Difficult for GD to reach global optima.
May get stuck in local optima.)



Convex vs. Not convex



We generally want to optimize convex functions

Using Gradient Descent.

→ For a function of 1 variable, if the 2nd derivative is non-negative in the domain, the function is convex in the domain

→ Quick test for Convexity - A line joining any two points on the curve must lie above the curve.

(2) MSE doesn't penalize much even for a perfect mismatch. Eg. $y_i = 1$ $\hat{y}_i = 0$

$$MSE = (y_i - \hat{y}_i)^2 = (1-0)^2 = 1$$

$$\text{Log Loss} = -1 \log 0 - 0 \log 1$$

$$= -\log 0$$ (tends to infinity)

Q.9.] For N datapoints and K classes,

$$\text{Multi-class Cross-entropy} = -\sum_{i=1}^{N} \sum_{k=1}^{K} y_i^k \log \hat{y}_i^k$$

Eg. ① $\begin{bmatrix} 0.8 \\ 0.1 \\ 0.1 \end{bmatrix}$ $\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ ② $\begin{bmatrix} 0.3 \\ 0.3 \\ 0.4 \end{bmatrix}$ $\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$

$\hat{y}_1$ y_1 $\hat{y}_2$ y_2

$$\text{Loss} = \left(-1\log(0.8) - 0\log(0.1) - 0\log(0.1)\right)$$

$$+ \left(-0\log(0.3) - 1\log(0.3) - 0\log(0.4)\right)$$

$$= -\left(\log(0.8) + \log(0.3)\right) = -\log 0.24$$

$$= \boxed{\log\left(\tfrac{25}{6}\right)}$$

Q.10.]

$$CE(p,y) = \begin{cases} -\log(p) & \text{if } y=1 \\ -\log(1-p) & \text{if } y=0 \end{cases}$$

Note:

y=1 is the majority class (we have a lot of it's samples)

$$P_t = \begin{cases} P & \text{if } y=1 \\ (1-p) & \text{if } y=0 \end{cases}$$

$$CE(p,y) = CE(p_t) = -\log(p_t)$$

Consider $\boxed{FL(p_t) = -(1-p_t)^{\gamma} \log(p_t) \;;\; \gamma \geq 0}$

→ Why is CE (pt) not well-suited for imbalanced datasets (Eg. Cancer detection)??

- Biased towards majority class

→ Can Focal Loss (FL(pt)) help here??

How??

→ In practice, $\alpha_t \begin{cases} \alpha & \text{if } y=1 \\ 1-\alpha & \text{if } y=0 \end{cases}$

'α' is a hyperparameter. }
'γ' is a hyperparameter. } can be tuned

$$\rightarrow \quad FL'(p_t) = -\alpha_t (1-p_t)^{\gamma} \log(p_t)$$

(used in practice.)

<u>Answer</u>. When p_t is high (easily classified samples from the majority class), the $(1-p_t)^{\gamma}$ downscales the loss contributed by such majority class samples, but when p_t is low, there is lesser downscaling ⇒ more contribution from minority class.

Q2: In case of linear activation - continuation.

$$= \sum w_i^2 w_i^4 \left(\sum_{j=1}^{n} x_j w_{ij}^1\right) + w_i^2 b$$

$$= \sum_{i=1}^{m} w_i^2 w_i^4 \sum_{j=1}^{n} x_j w_{ij}^1 + w_i^2 b$$

$$= \sum_{j=1}^{m} \sum_{l=1}^{m} w_i^2 w_i^4 x_j w_{ij}^1 + \sum_{i=1}^{m} w_i^2 b$$

$$= \sum_{j=1}^{m} x_j \underbrace{\left(\sum_{i=1}^{m} w_i^2 w_i^4 w_{ij}^1\right)}_{w_j'}$$