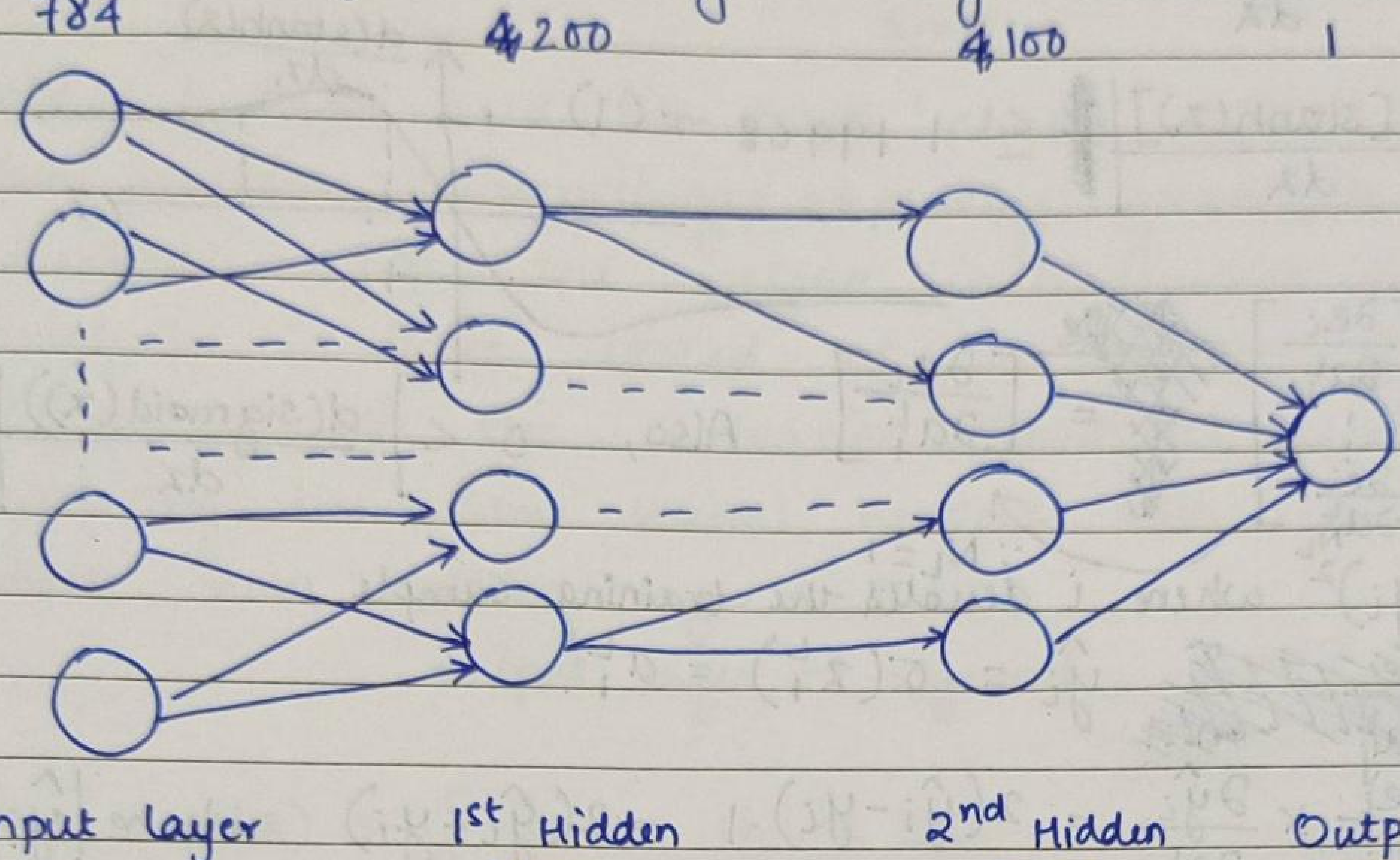
200100154

IE643

Assignment 2

(d) each hidden layer → stanh activation functions

output layer → logistic sigmoid activation function



In a generalized setting, we have shown in class that →

$\therefore \nabla_{W^\ell} e = \mathrm{Diag}(\phi^{\ell'}) V^{\ell+1} \cdots V^L \delta^L (a^{\ell-1})^T$

$$\text{where } V^{\ell+1} = \begin{bmatrix} w_{11}^{\ell+1} & \cdots & w_{N_{\ell+1}1}^{\ell+1} \\ \vdots & & \vdots \\ w_{1N_\ell}^{\ell+1} & \cdots & w_{N_{\ell+1}N_\ell}^{\ell+1} \end{bmatrix} \begin{bmatrix} \phi'(z_1^{\ell+1}) & & \\ & \ddots & \\ & & \phi'(z_{N_{\ell+1}}^{\ell+1}) \end{bmatrix}$$

Note that the error gradients at the last layer flow back into the previous layers

- exploding gradient problem ≡ $\nabla_{W^\ell} e$ gradients become very large in magnitude i.e. approach ∞
- vanishing gradient problem ≡ $\nabla_{W^\ell} e$ gradients become very small in magnitude, i.e. they approach 0

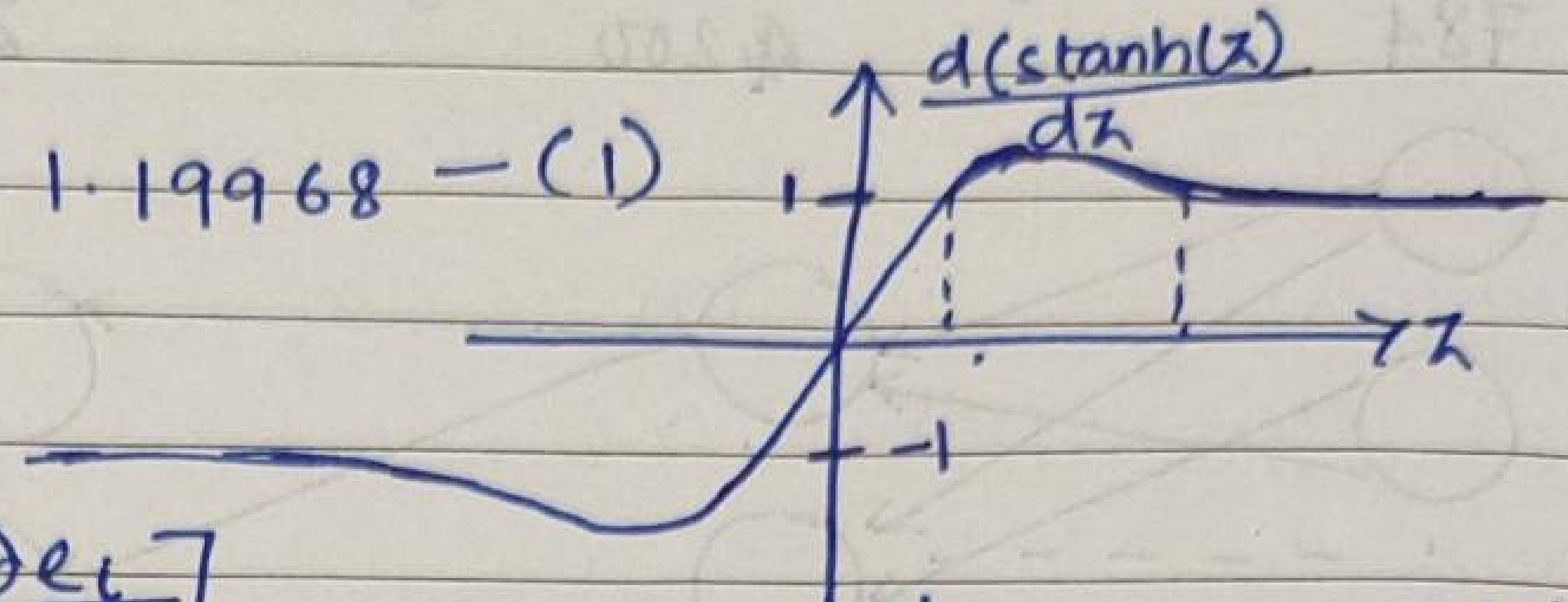
* The magnitude of $\nabla_{W^\ell} e$ gradients depends upon $V^{\ell+1} V^{\ell+2} \cdots V^L \delta^L$

* For our analysis, it suffices to only look at the derivatives of the activation functions and the derivative of the error function wrt the activation of the output layer.

$$\text{stanh}(z) = z\,\sigma_{tan}(z) \quad \text{where} \quad \sigma_{tan}(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

So, $$\frac{d[\text{stanh}(z)]}{dz} = \tanh(z) + z\,\text{sech}^2(z)$$

$$0 \leq \left|\frac{d[\text{stanh}(z)]}{dz}\right| \leq 1.19968 \quad -(1)$$



Now, $$\delta_L = \begin{bmatrix} \frac{\partial e_i}{\partial a_1^L} \\ \vdots \\ \frac{\partial e_i}{\partial a_{N_L}^L} \end{bmatrix} = \begin{bmatrix} \frac{\partial e_i}{\partial a_1^L} \end{bmatrix} \quad \because N_L = 1$$

Also, $$0 < \left|\frac{d(\text{sigmoid}(z))}{dz}\right| \leq \frac{1}{4}$$

- $e_i = (\hat{y}_i - y_i)^2$ where i denotes the training sample
- We have $\hat{y}_i = \sigma(z_1^L) = a_1^L$

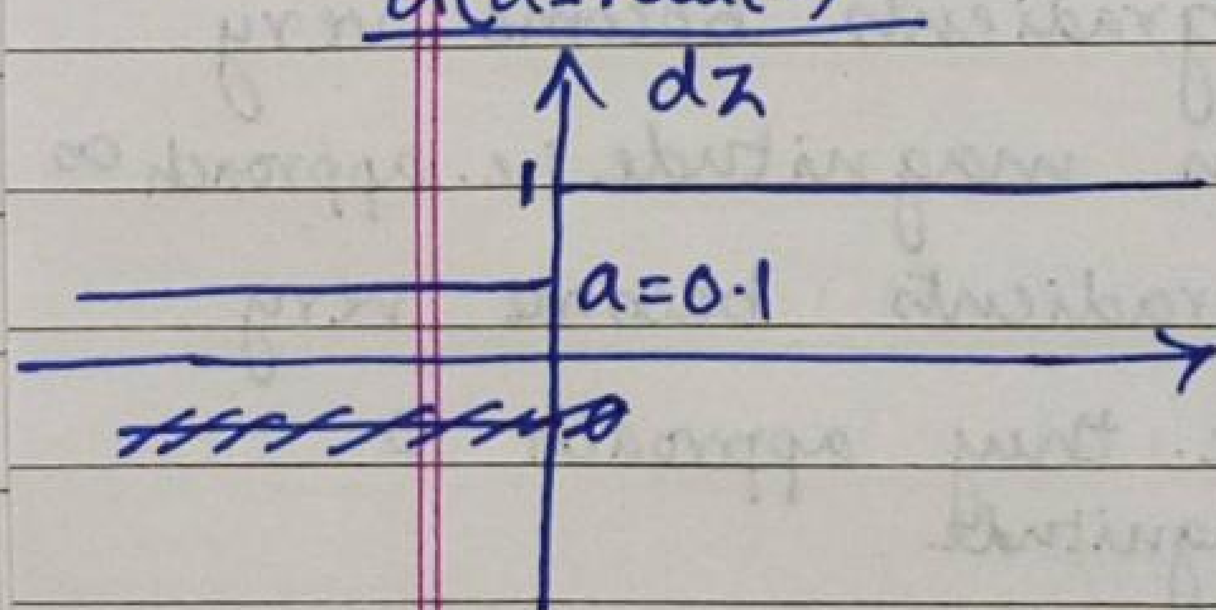
So, $$\frac{\partial e_i}{\partial a_1^L} = \frac{\partial e_i}{\partial \hat{y}_i} \times \frac{\partial \hat{y}_i}{\partial a_1^L} = 2(\hat{y}_i - y_i).1 = \underbrace{2(\hat{y}_i - y_i)}_{\text{finite}}$$ where $|\hat{y}_i| \leq 1$ and $y_i = \{+1, -1\}$

Due to (1), we can say that as the derivatives or slope gets larger and larger as we go backward with every layer during back prop $\Rightarrow$ Exploding gradients

(e) Considering the same neural network architecture before where now hidden layer $\rightarrow$ a-relu
output layer $\rightarrow$ sigmoid

Now, $$\text{a-relu}(z) = \begin{cases} az & \text{if } z < 0 \\ z & \text{else} \end{cases} \quad \text{where } a = 0.1$$

So, $$\frac{d(\text{a-relu}(z))}{dz} = \begin{cases} a & \text{if } z < 0 \\ 1 & \text{else} \end{cases}$$



Also, $$0 < \left|\frac{d(\text{sigmoid}(z))}{dz}\right| \leq \frac{1}{4}$$

$$0.1 \leq \left|\frac{d(\text{a-relu}(z))}{dz}\right| \leq 1$$ From this bound and the previous discussion on $\frac{\partial e_i}{\partial a^L}$, we can safely interpret that a-ReLU causes vanishing gradients for inputs smaller than zero while for other inputs they allow the

gradient to pass through. The chance for ~~a~~ vanishing gradient can be minimized by taking a larger value of 'a'. Ideally, we would want $a = \lim_{x \to 1^-} x$ i.e. $a = 0.99, 0.999 \ldots$

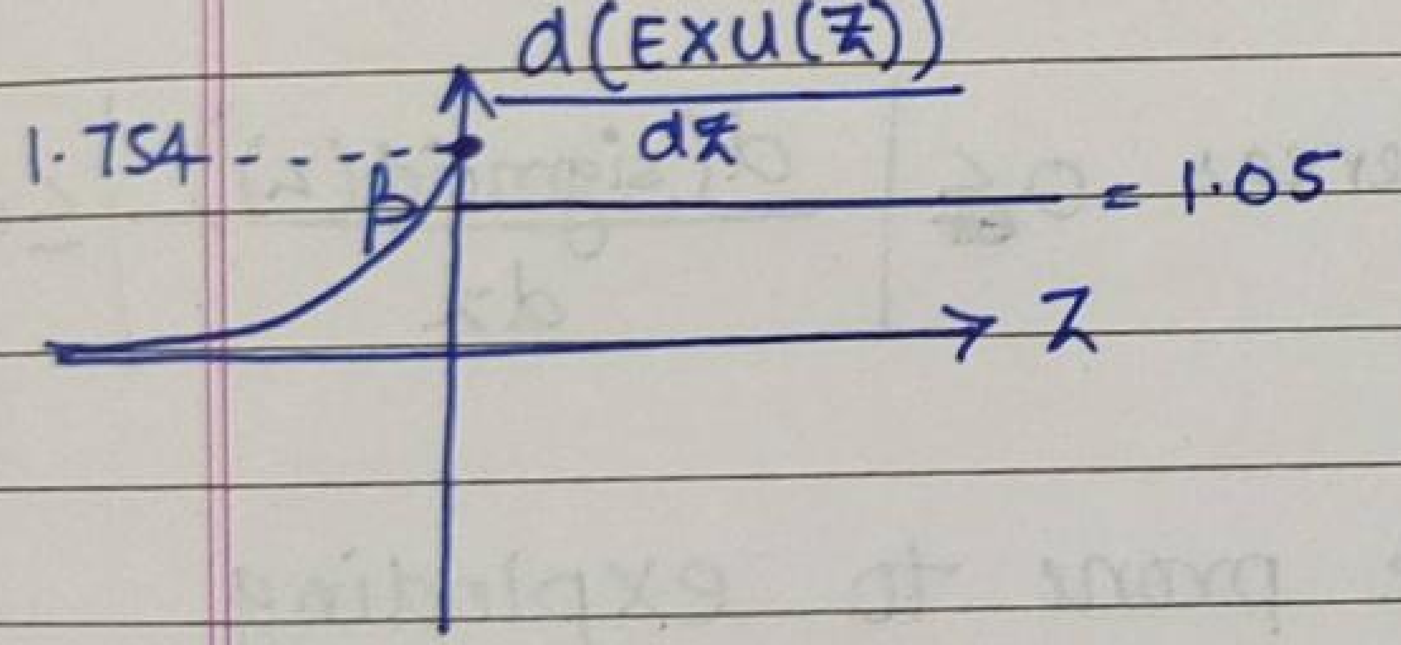
* Note :- I have used the term 'chance' above because this problem of vanishing/exploding gradients ~~allows~~ depends upon the weight initialization also. For my analysis, I have looked at the bounds of the derivatives of the activation functions used.

(f) Considering the same neural network architecture before where now hidden layer → EXU
output layer → sigmoid

Now, $$\text{EXU}(z) = \begin{cases} \beta z & \text{if } z \geq 0 \\ \beta y(\exp(z) - 1) & \text{else} \end{cases}$$

where $\beta = 1.05$ and $y = 1.67$

$$\frac{d(\text{EXU}(z))}{dz} = \begin{cases} \beta & \text{if } z \geq 0 \\ \beta y \exp(z) & \text{else} \end{cases}$$



So, $$0 \leq \left| \frac{d(\text{EXU}(z))}{dz} \right| \leq 1.754$$

Also, $$0 < \left| \frac{d(\text{sigmoid}(z))}{dz} \right| \leq \frac{1}{4}$$

Note, here we may be tempted to say that this model would suffer from exploding gradients since $\left| \frac{d(\text{EXU}(z))}{dz} \right|_{max} = 1.754$. However, if we closely examine the graph of $\frac{d(\text{EXU}(z))}{dz}$ v/s z, we will notice that the range of z for ~~are~~ which $\frac{d(\text{EXU}(z))}{dz} \geq 1.05$ is extremely small. Rather, this model ~~should~~ should more likely to suffer from vanishing gradients since $\lim_{z \to -\infty} \frac{d(\text{EXU}(z))}{dz} = 0$

(g) d → stanh, stanh, sigmoid
e → a-ReLU, a-ReLU, sigmoid
f → EXU, EXU, sigmoid

• Networks that were more prone to exploding gradient issue → 'd' mode with stanh

Reasons →

• large weight initialization
• accumulation of large error gradients which result in extremely large updates to neural network model weights during training.
• $|\text{mag. of derivative of stanh}|_{max} > 1$

• Networks that were more prone to vanishing gradient issue → 'f' model with EXU

Reason → • many times multiplication of smaller number (magnitude of activation functions' derivative) results in a very small number (close to 0)

• derivative of sigmoid is always below 0.25
• derivative of EXU over a longer range of z gives a value close to 0.

(h) d → stanh, stanh, linear
e → a-ReLU, a-ReLU, linear
f → EXU, EXU, linear

$$\left|\frac{d(\text{linear}(z))}{dz}\right| = 1 \quad \text{whereas} \quad 0 \le \left|\frac{d(\text{sigmoid}(z))}{dz}\right| \le 0.25$$

⇒ The model becomes more prone to exploding gradient problem and becomes less prone to vanishing gradient problem. due to an increase in the maximum absolute value of derivative of activation function used in the output layer.

$$0 \le \left|\frac{d(\text{ReLU}(z))}{dz}\right| \le 1$$

⇒ The model should be more stable wrt to both exploding and vanishing gradient problems due to the range of $\frac{d(\text{ReLU}(z))}{dz} \in [0,1]$.