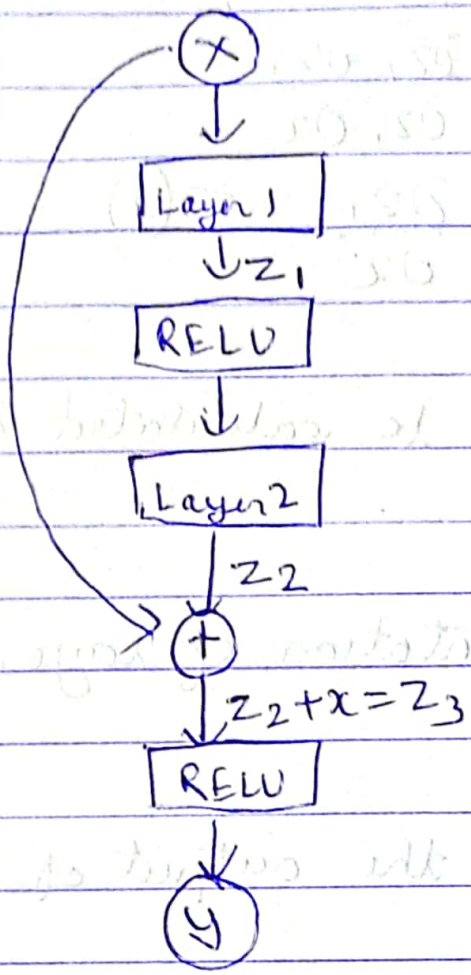
Name – T. Vijay

Roll.no – CS17BTECH11040

Q1



$\therefore$ Output $y = RELU(z_3)$

where $z_3 = z_2 + x$

and z_2 is layer 2 output

$\therefore \frac{\partial y}{\partial z_3} = RELU'(z_3)$

$RELU'(a) = (a > 0)$

$$\frac{\partial y}{\partial z_2} = \frac{\partial y}{\partial z_3}\frac{\partial z_3}{\partial z_2}$$

$$= RELU'(z_3)\frac{\partial (z_2+x)}{\partial z_2}$$

$$\frac{\partial y}{\partial z_2} = RELU'(z_3)$$

$$\frac{\partial y}{\partial z_1} = \frac{\partial y}{\partial z_2}\frac{\partial z_2}{\partial z_1} = RELU'(z_3)\frac{\partial z_2}{\partial z_1}$$

$$\frac{\partial y}{\partial x} = \frac{\partial y}{\partial z_3} \frac{\partial z_3}{\partial z_2} \frac{\partial z_2}{\partial z_1} \frac{\partial z_1}{\partial x}$$

$$= \text{RELU}'(z_3) * 1 \times \frac{\partial z_2}{\partial z_1} \frac{\partial z_1}{\partial x}$$

$$= (z_3 > 0) \frac{\partial z_2}{\partial z_1} \frac{\partial z_1}{\partial x} \quad - (1)$$

$\frac{\partial z_2}{\partial z_1}$ and $\frac{\partial z_2}{\partial z_1}$ can be calculated as

we know the differentation of layer 2 & layer 1

Now lets assume the output of Resnet is Y

$$\text{then } \frac{\partial Y}{\partial x} = \frac{\partial Y}{\partial y} \frac{\partial y}{\partial x}$$

$$= \frac{\partial Y}{\partial y} (z_3 > 0) \frac{\partial z_2}{\partial z_1} \frac{\partial z_1}{\partial x} \quad (\text{From } (1))$$

2. Let $(n \times n)$ image and $(f \times f)$ filter are used

$$\text{Output size} = \left\lfloor \frac{n - f + 2p}{s} \right\rfloor + 1$$

where p is padding and s is stride

$\therefore$ If $f = 3$, $p = 0$, $s = 1$

$$\text{Output size} = ((n-2) \times (n-2))$$

So to activate neuron at layer 4

Size at layer 3 $\Rightarrow (3 \times 3)$

" " " 2 $\Rightarrow (5 \times 5)$

" " " 1 $\Rightarrow (7 \times 7)$

" " " 0 $\Rightarrow (9 \times 9)$

$\therefore$ Input pixels $= 9 \times 9 = 81$

3. If the no. of hidden layers are increased, then it can learn more complex functions

So the bias decreases.

Now the representation power increased, it can become unnecessarily complex. Hence this leads to increase in variance as the loss surface become complex and it makes convergence difficult.

Q4 $\tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}}$ & $\sigma(a) = \frac{1}{1+e^{-a}}$

$\tanh(a) + 1 = \frac{e^a - e^{-a}}{e^a + e^{-a}} + 1$

$\tanh(a) + 1 = \frac{2e^a}{e^a + e^{-a}} = \frac{2}{1+e^{-2a}}$

$\tanh(a) + 1 = 2\sigma(2a)$

$\therefore \sigma(a) = \frac{\tanh(a/2) + 1}{2}$ — (1)

Now consider the given 2 layer NN

$$y_k(x, w) = \sum_{j=1}^{M} w_{kj}^{(2)} \sigma\left(\sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}\right) + w_{k0}^{(2)}$$

From (1) we can rewrite above eqn as

$$y_k(x, w) = \sum_{j=1}^{M} \frac{w_{kj}^{(2)}}{2} \tanh\left(\sum_{i=1}^{D} \frac{w_{ji}^{(1)}}{2} x_i + \frac{w_{j0}^{(1)}}{2}\right) + \sum_{j=1}^{M} \frac{w_{kj}^{(2)}}{2} + w_{k0}^{(2)}$$

In the new network equation, we can see that

new $w_{k0}^{(2)} = \sum_{j=1}^{M} \frac{w_{kj}^{(2)}}{2} + w_{k0}^{(2)}$ & other weights are halt

Thus the classmate is right as an equivalent NN exists with tanh activation

Q6 Since they have access to deep learning model trained on a similar dataset, they can use transfer learning.

In transfer learning, they can replace the last layer with 200 class output layer. They will have to randomly initialize weight to this layer and freeze weights of all other layers.

Now, they can train this new model on the small dataset to learn weights of the last layer.

So, in this way they can obtain better accuracy. This will also save time as we don't have to write Neural network from scratch, train and tune the model.

Q5 Given: ① $E(w) \approx E(w^*) + \frac{1}{2}(w-w^*)^T H (w-w^*)$ — ①

② $Hu_i = \lambda_i u_i$ — ②

So u_is are orthonormal eigen vectors of H.

$\therefore u_i^T u_j = \delta_{ij}$ i.e $u_i^T u_j = 0$ for $i \neq j$ or 1 for $i = j$ — ③

$\because w - w^* = \sum_i \alpha_i u_i$ — ④

From ① & ④

$$E(w) = E(w^*) + \frac{1}{2}\left(\sum_i \alpha_i u_i^T\right) H \left(\sum_i \alpha_i u_i\right)$$

$$= E(w^*) + \frac{1}{2}\left(\sum_i \alpha_i u_i^T\right)\left(\sum_i \lambda_i \alpha_i u_i\right) \quad \text{(From ②)}$$

$$= E(w^*) + \frac{1}{2}\sum_i \alpha_i^2 \lambda_i \quad \text{(From ③)}$$

We set this value to constant K to get a contour

$$K = E(w^*) + \frac{1}{2}\left(\sum_i \alpha_i^2 \lambda_i\right)$$

$$\therefore 2(k - F(w^*)) = \sum_i \frac{\alpha_i^2}{\left(\frac{1}{\sqrt{\lambda_i}}\right)^2}$$

We can see that above equation is a ellipse whose axis are aligned with eigenvector v_i with length $\propto \frac{1}{\sqrt{\lambda}}$