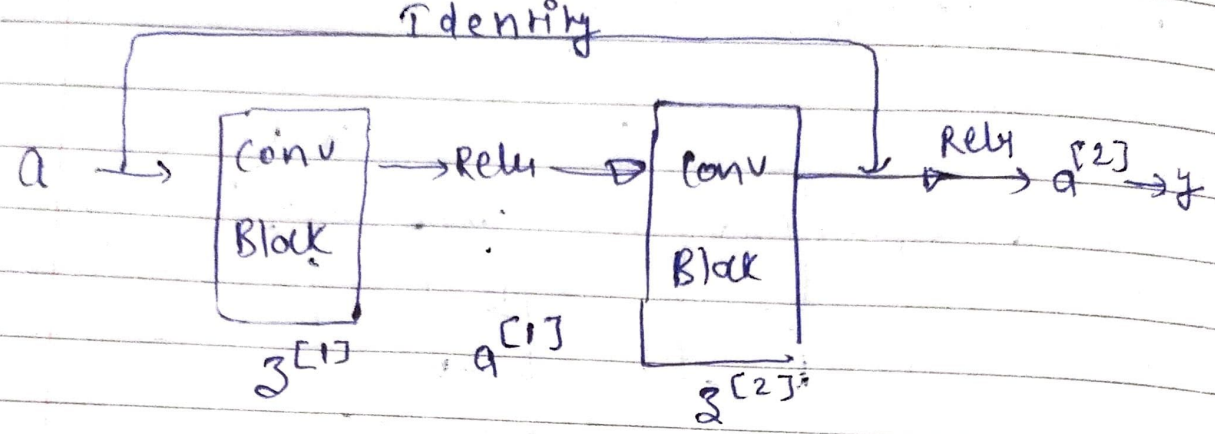
Ans-1



a → Last Layer activation as input to Residual Block.

$$z^{[1]} = w^{[1]} a + b^{[1]}$$

$$a^{[1]} = g(z^{[1]}) = \text{Relu}(z^{[1]})$$

$$z^{[2]} = w^{[2]} a^{[1]} + b^{[2]}$$

$$\hat{y} = a^{[2]} = g[z^{[2]} + a) = \text{Relu}[z^{[2]} + a]$$

$\frac{\partial E}{\partial \hat{y}} = \frac{\partial E}{\partial a^{[2]}} = \delta^{[2]}$ → derivative of Loss wrt Resblock output

$$\frac{\partial E}{\partial z^{[2]}} = \frac{\partial E}{\partial a^{[2]}} \times \frac{\partial a^{[2]}}{\partial z^{[2]}} = \delta^{[2]} \times g'(z^{[2]} + a)$$

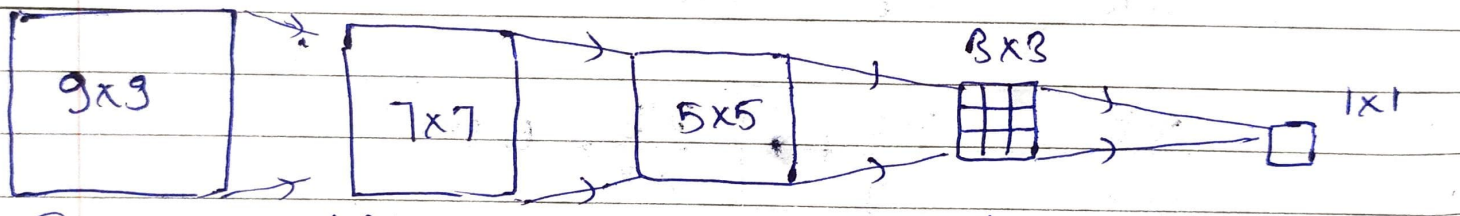
Similarly on solving

$$\frac{\partial E}{\partial z^{[1]}} = \delta^{[2]} \cdot \delta^{[1]} \cdot g'(z^{[1]}) \quad \left[\delta^{[1]} = \frac{\partial E}{\partial a^{[1]}}\right]$$

Ans-2 →

Given
Filter size = 3 x 3
Number of convolution Layers = 4
Stride = 1
Pooling = No

For every convolution Layer the size of image Layer reduces from n to n-2.



Hence Input image dimension of 9x9 will contribute to one activation neuron in the 4th non image Layer.

So support size for one Activation
9x9 = 81 Pixels.

Ans-3 →

If the number of hidden units is increased Network will be able to learn more complex function representation.

And as we know more complicated models result in Lower bias & high variance. models with high complexity tends to overfit as they try to follow training data too closely & does not generalise well on testing Data.

Ans-4→ Given a two layer neural network

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

h is a linear Activation function
$\sigma(a)$ is sigmoid Activation function.

First Let's prove that tanh is rescaled & shifted version of the sigmoid function.

Sigmoid Property :→

$$1 - \frac{1}{1+e^{-x}} = \frac{1}{1+e^{x}} \Rightarrow 1-\sigma(x) = \sigma(-x)$$

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

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

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

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

$$= 1 - 2(\sigma(-2n))$$

$$= 1 - 2 + 2\sigma(2n) \quad [\text{Sigmoid Property}]$$

$$\tanh(n) = 2\sigma(2n) - 1 \qquad \text{eq (1)}$$

which Prove that tanh is rescaled & shifted version of sigmoid function.

This shift of -1 will be taken care of by bias learning of hidden layer.

The inner 2 in $2\sigma(2n)-1$, which scale input by a factor of 2, will be handled by weights present in first hidden Layer.

& the outer scaling of 2 will be handled by weights in 2nd layer which contain linear activation function.

$$y_k(n, w) = \sum_{j=1}^{M} \frac{w_{kj}^{[2]}}{2} \cdot 2\sigma\left(\sum_{i=1}^{D} 2\left[\frac{w_{ji}}{2} n_i + \frac{w_{j0}}{2}\right]\right) + w_{k0}^{[2]}$$

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

$$y_k(x, w) =$$

$$\sum_{j=1}^{M} \frac{w_{kj}^{[2]}}{2} \left[2\sigma \left(\sum_{i=1}^{D} 2\left[\frac{w_{ji}}{2} \cdot x_i + \frac{w_{j0}}{2} \right] \right) - 1 \right] + w_{k0}^{[2]} + \frac{w_{kj}^{[2]}}{2}$$

which is same as.

$$\rightarrow \sum_{j=1}^{M} \frac{w_{kj}^{[2]'}}{2} \tanh\left[\frac{w_{ji}'}{2} x_i + \frac{w_{j0}}{2} \right] + w_{k0}^{[2]} + \frac{w_{kj}^{[2]}}{2}$$

→ By eq (1)

$$\rightarrow \sum_{j=1}^{M} w_{kj}'^{[2]} \tanh\left[\vec{w}_{ji} \cdot x_i + w_{j0}' \right] + w_{k0}'^{[2]}$$

So only weights will change for learning same function.

Ans-5 →

Quadratic Error

$$E(w) \simeq E(w^*) + \frac{1}{2}(w-w^*)^T H (w-w^*) \quad -(1)$$

H = Hessian matrix evaluated at w^*

Eigen value equation

$$\rightarrow Hu_i = d_i u_i \quad -(2)$$

We have to show that contours of constant error are ellipses whose axes are aligned with eigen vectors u_i with the lengths that are inversely proportional to the square root of the corresponding eigen values d_i.

Our Hint → $$w - w^* = \sum_i \alpha_i u_i \quad -(3)$$

→ Eigen vectors for orthonormal set

$$u_i^T u_j = \delta_{ij} \quad -(4)$$

putting $$w - w^* = \sum_i \alpha_i u_i$$

into Quadratic Error we get →

$$E(w) = E[w^*] + \frac{1}{2} \sum_i q_i H_i \cdot H\left(\sum_i q_i H_i\right)$$

Using eq 2 & 4 we get

$$E(w) = E[w^*] + \frac{1}{2} \sum_i d_i q_i^2$$

For finding constant Error contour putting the value of $E(w)$ to c we get

$$E[w^*] + \frac{1}{2} \sum_i d_i q_i^2 = C$$

which is

$$\sum_i d_i q_i^2 = 2C - 2E[w^*] = \bar{C} \text{ [Another constant]}$$

This is the equation for an ellipse. The axes of this ellipse are alligned with the coordinates described by the variables q_i.

For finding Length of $\perp$ axis j, we set $q_i = 0$ for all $i \neq j$ & solving q_j gives

$$\alpha_j = \left(\frac{c}{d_j}\right)^{1/2}$$

which is inversely proportional to square root of eigen value d_j.

□ Hence proved

ANS-6→

Number of species = 200
Images per species = 20

With small amount of data like this it is impossible to train a decent size neural network from scratch.

So the only option for deploying deep learning model is to use transfer Learning.

→ Base model being trained on 1 million images from 1000 classes learn basic features such as edges & blobs in initial layers so we can fridge some initial layers while retraining based model on target dataset.

→ Or we can set very Low Learning rate for initial layer, medium Learning rate for middle layers and High Learning rate for Last layer

→ We change Last Layer Prediction from 1000 class to 20 class as target dataset only contain 20 species.

→ We devide the available images into training & validation & test set if needed.
or just train & test set so we are well aware that our model is not overfitting or underfitting.

→ With this small number of images it may still be difficult to transfer base model on target dataset so we can use image augmentation like Random Crop & Flip, shear, Colour Augmentation & so on to increase number of images.
& by this way our model will learn more robust representation.

→ Early stopping can be performed by simultaneously checking the model output on test set.

→ IF Base model is Large (Parameter wise) some of the Last layers can be removed & then softmax with 20 output can be added.