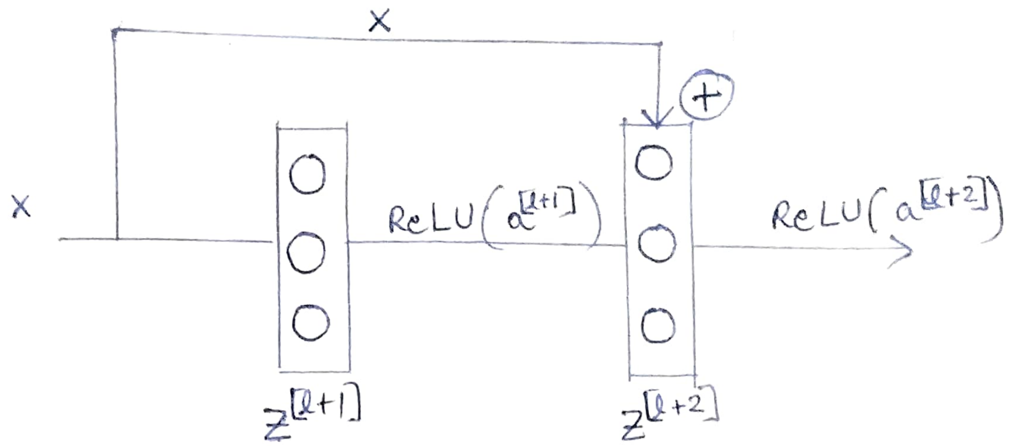
Solution - 1



Forward pass (assuming bias = 0 everywhere)

$$z^{[l+1]} = x \cdot w^{[l+1]}$$

$$a^{[l+1]} = \text{ReLU}(x \cdot w^{[l+1]})$$

$$z^{[l+2]} = a^{[l+1]} \cdot w^{[l+2]} + x$$

↳ because of identity connection.

$$a^{[l+2]} = \text{ReLU}(a^{[l+1]} \cdot w^{[l+2]} + x)$$

Note :- if in any layer $[a^{l+1} \cdot w^{l+2}] = 0$.

$a^{[l+2]}$ will be equal to input X itself

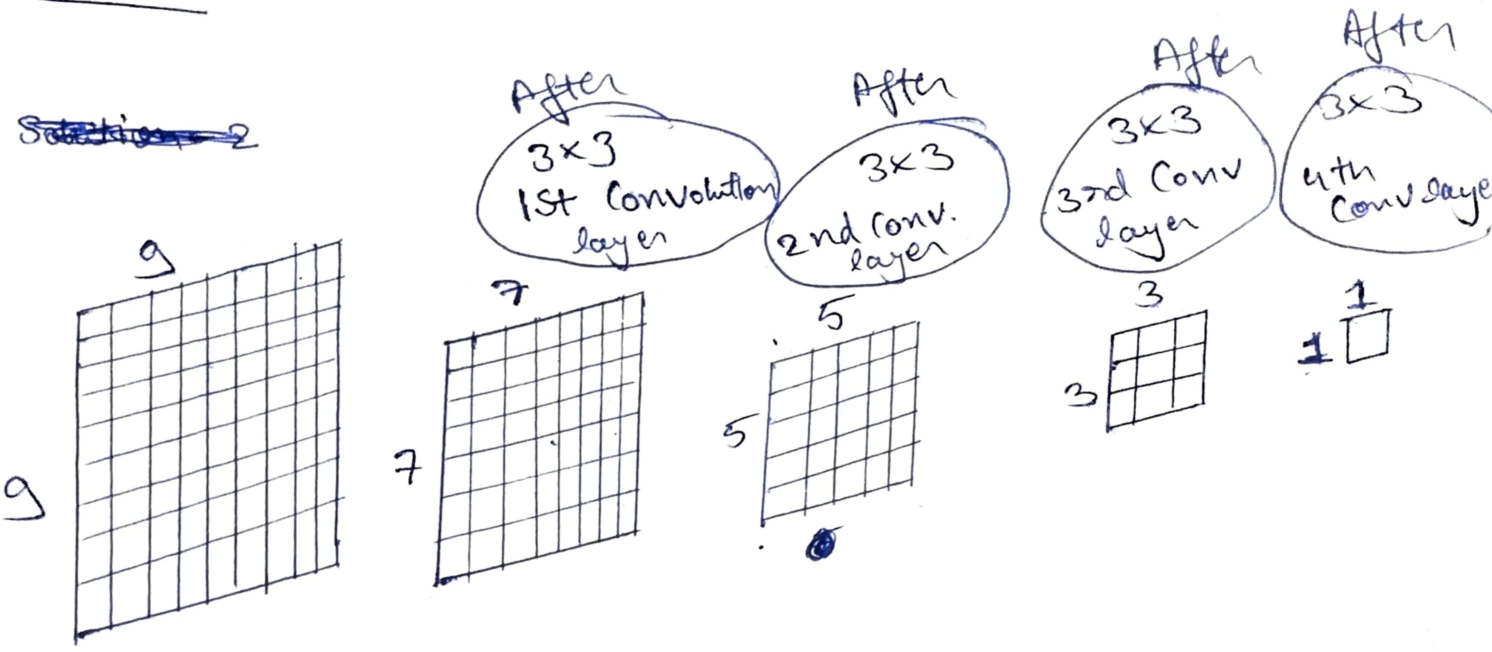
as $\text{ReLU}(x) = x$

Backward Pass

$$\frac{\partial E}{\partial x} = \frac{\partial E}{\partial z^{[l+2]}} \cdot \frac{\partial z^{[l+2]}}{\partial x} = \frac{\partial E}{\partial z^{[l+2]}} \frac{\partial (a^{[l+1]} w^{[l+2]} + x)}{\partial x}$$

$$\boxed{\frac{\partial E}{\partial x} = \frac{\partial E}{\partial z^{[l+2]}} \cdot \left(\frac{\partial a^{[l+1]} w^{[l+2]}}{\partial x} + 1 \right)}$$

Sollution - 2



- For a neuron in feature map after 4th Convolution layer of 3x3, receptive field size in previous layer will be 3x3.
- Similary in further previous layer the receptive field for generating 3x3 weights in layer 3, receptive field for layer 2 will be 5x5
- Similarly layer 2 to layer 1 will have receptive field of 7x7.
- And finally receptive field of single neuron after 4th layer will have ~~receptive~~ Contribution of 9x9 pixels of input image

$$\boxed{1\times1 \rightarrow 3\times3 \rightarrow 5\times5 \rightarrow 7\times7 \rightarrow 9\times9}$$

Solution-3

Increasing the number of neurons in the hidden layer should increase the variance and reduce the bias of our model.

Increasing the neurons in the hiddenlayer is useful when the model is underfitting.

underfitting.

But if we increase the neurons in hiddenlayer to very large values than variance of our model will increase and thus our model will over fit.

Overfitting.

Solution - 4

Sigmoid function is given as

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

Hyperbolic tangent function is given as

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

this can be further broken as

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

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

$$\tanh(z) = \left(\frac{1}{1+e^{-2z}}\right) - \left(\frac{1}{e^{2z}+1}\right)$$

$$\tanh(z) = \sigma(2z) - \sigma(-2z)$$

$$\tanh(z) = \sigma(2z) - (1 - \sigma(2z))$$

$$\boxed{\tanh(z) = 2\,\sigma(2z) - 1}$$

Replacing $\sigma(z)$ with $2(\sigma(2z) - 1)$ in given function

we get

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

$$\boxed{y'(x, w) = 2 \sum_{j=1}^{M} w_{kj}^{(2)} \, \sigma\left(2 w_{ji}^{(1)} x_i + 2 w_{j0}^{(1)}\right) - w_{kj}^{(2)}}$$

which is just some linear transformation of

Previously given function as output.

Solution-5

- error is defined by.

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

$$H u_i = \lambda_i u_i \quad - (2)$$

- u_i are the eigenvectors forming complete Orthogonal.

$$u_i^T u_j = \delta_{ij} \quad - (3)$$

- as given in question $w - w^*$ can be written as $\sum \alpha_i u_i$

$$w - w^* = \sum \alpha_i u_i \quad - (4)$$

- Putting equation - (2) & eqn - (4) in eqn - (1) we get

$$E(w) = E(w^*) + \frac{1}{2}\left(\sum_i \alpha_i u_i\right)^T \lambda_i \left(\sum_i \alpha_i u_i\right) \quad \to (5)$$

- using eqn (3), eqn (5) can be written as

$$\boxed{E(w) = E(w^*) + \frac{1}{2}\sum_i \lambda_i \alpha_i^2} \to (6)$$

- Now for plotting contour for any constant error K eqn (6) can be written as

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

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

where $K - E(w^*)$ will again be a constant, Let K'

$$\frac{1}{2}\sum \lambda_i \alpha_i^2 = K'$$

$$\boxed{\sum \lambda_i \alpha_i^2 = K''}$$

for 2-D contour.

$$\lambda_1 \alpha_1^2 + \lambda_2 \alpha_2^2 = K''$$

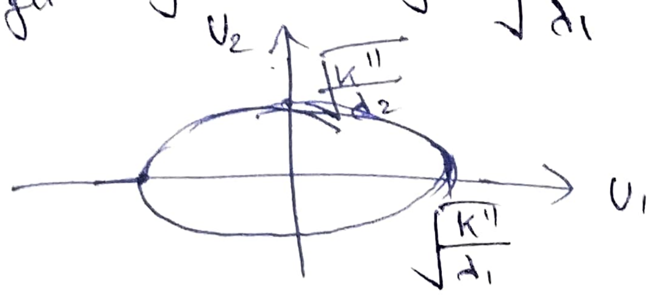
$$\boxed{\frac{\alpha_1^2}{\left(\sqrt{\frac{K''}{\lambda_1}}\right)^2} + \frac{\alpha_2^2}{\left(\sqrt{\frac{K''}{\lambda_2}}\right)^2} = 1}$$

This is a equation of ellipse. and the ~~axes~~ axis of this ellipse is alligned with the direction of eigen vectors

length of both the axis can be given as

Put $\alpha_1 = 0$ to get minor axis length $= \sqrt{\frac{K''}{\lambda_2}}$

Put $\alpha_2 = 0$ to get major axis length $= \sqrt{\frac{K''}{\lambda_1}}$



Hence we can clearly see that length of axis are inversely proportional to square root of eigen values

$$\boxed{\text{length of axis} \propto \frac{1}{\sqrt{\lambda}}}$$

Solution - 6

in the case of Zoo. Dataset of Kaziranga National park will be quite similar to that of Olympic National Park, washington. Only few species that are dependent on Geographic location will vary and will be new for us.

So, as the data set of image collected from our ~~Kaz~~ Kaziranga national park is small but similar to Olympic National park, we can use the concept of ~~transfer~~ transfer learning in the following steps.

Step 1 Remove fully connected layers at the end of Pretrained ~~data~~ model from washingdon.

Step 2 Add new fully connected layers with same number of classes as species in our zoo (i.e 200)

Step-3 ~~Read~~ Freeze all the weights in Previously trained model except in final fully connected layer. and in final fully connected layer we can randomly assign the weights.

Step 4 : Train our network with new fully connected layers for few epochs to get the updated weights for final fully connected layers.

Step 5 : Our model is ready for testing & hyperparameter tuning.