Deep Learning | EX02

Back Propagation & FF Coding

800

Computer Vision | Zahra Amini

Telegram: @zahraamini_ai & Instagram:@zahraamini_ai & LinkedIn: @zahraamini-ai

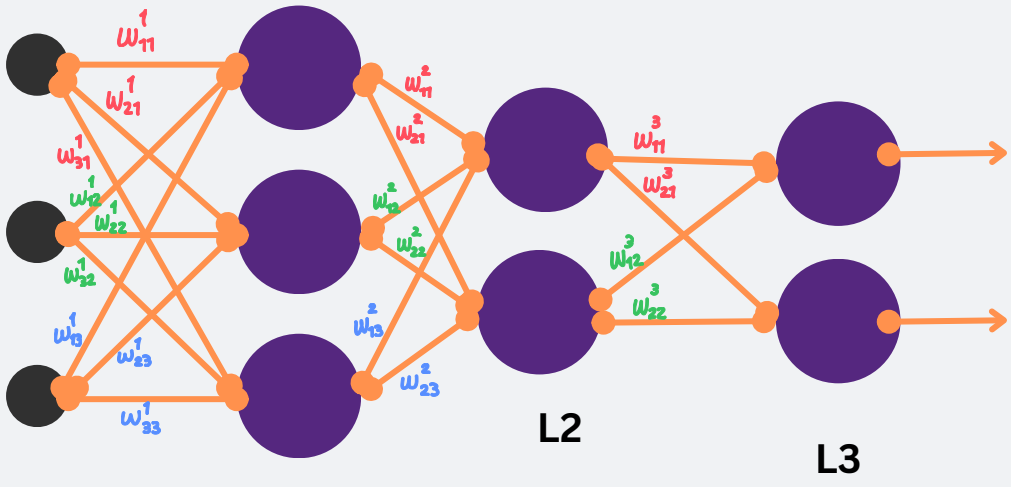
https://zil.ink/zahraamini

1. **For the below feed-forward neural network architecture obtain the following weights after 1st Back Propagation iteration (epoch): [ignore the biases change during backpropagation]**

$w_{11}^{1} = ?$

$w_{21}^{2} = ?$

$w_{22}^{3} = ?$

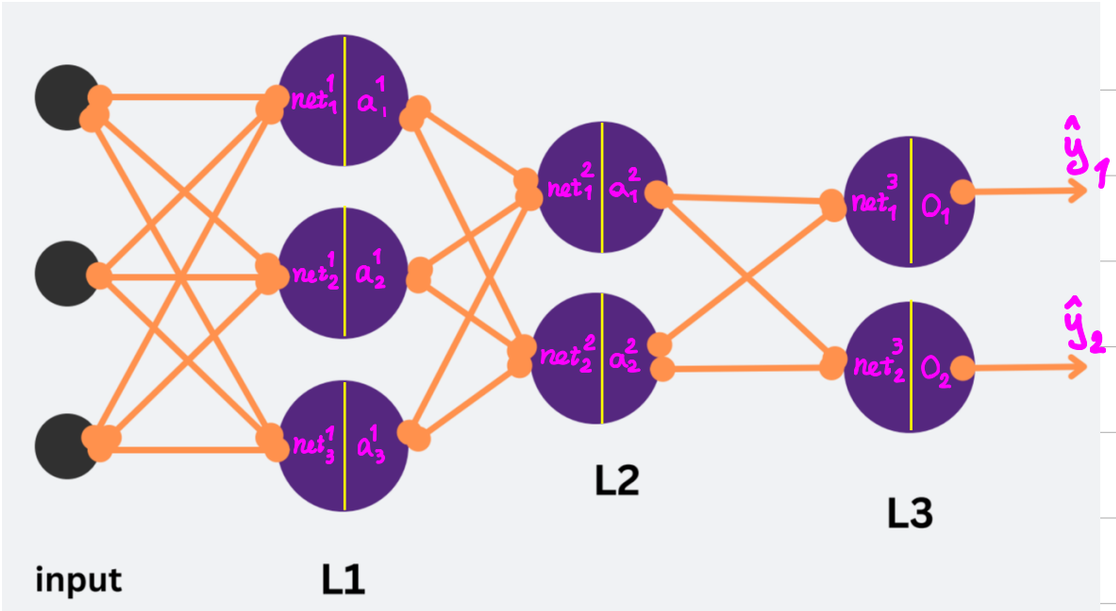


inputs	L1-Weights	L1-Biases	L2-Weights	L2-Biases	L3-Weights	L3-Biases
0.3	w_{11} w_{12} w_{13} 0.1 0.2 0.2	0.2	w_{11} w_{12} w_{13} 0.1 0.2 0.2	0.1	w_{11} w_{12} 0.5 0.1	0.5
0.5	w_{21} w_{23} w_{23} 0.4 0.1 0.2	0.2	w_{21} w_{22} w_{23} 0.4 0.1 0.2	0.4	w_{21} w_{22} 0.5 0.5	0.3
0.8	w_{31} w_{32} w_{33} 0.2 0.3 0.1	0.1				

- **Activation Functions:**
  - L1: Relu
  - L2: Tanh
  - L3: Sigmoid
- **Loss Function:** MSE
- **Optimizer:** GD

Alpha = 0.01

$y\_true = \begin{bmatrix} 0.5 \\ 0.8 \end{bmatrix}$



1. Feed Forward Pass:

Layer #1:

$$a_1^1 = ReLU(x_1 \cdot w_{11}^1 + x_2 \cdot w_{12}^1 + x_3 \cdot w_{13}^1 + b_1^1) = ReLU(0.3 \times 0.1 + 0.5 \times 0.2 + 0.8 \times 0.2 + 0.2) = ReLU(0.49) = 0.49$$

$$a_2^1 = ReLU(x_1 \cdot w_{21}^1 + x_2 \cdot w_{22}^1 + x_3 \cdot w_{23}^1 + b_2^1) = ReLU(0.3 \times 0.4 + 0.5 \times 0.1 + 0.8 \times 0.2 + 0.2) = ReLU(0.53) = 0.53$$

$$a_3^2 = ReLU(x_1 \cdot w_{31}^1 + x_2 \cdot w_{32} + x_3 \cdot w_{33}^1 + b_3^1) = ReLU(0.3 \times 0.2 + 0.5 \times 0.3 + 0.8 \times 0.1 + 0.1) = ReLU(0.39) = 0.39$$

Layer #2:

$$a_1^2 = \tanh(a_1^1 \cdot w_{11}^2 + a_2^1 \cdot w_{12}^2 + a_3^1 \cdot w_{13}^2 + b_1^2) = \tanh(0.49 \times 0.1 + 0.53 \times 0.2 + 0.39 \times 0.2 + 0.1) = \tanh(0.333) \approx 0.321$$

$$a_2^2 = \tanh(a_1^1 \cdot w_{21}^2 + a_2^1 \cdot w_{22}^2 + a_3^1 \cdot w_{23}^2 + b_2^2) = \tanh(0.49 \times 0.4 + 0.53 \times 0.1 + 0.39 \times 0.2 + 0.4) = \tanh(0.727) \approx 0.621$$

Layer #3:

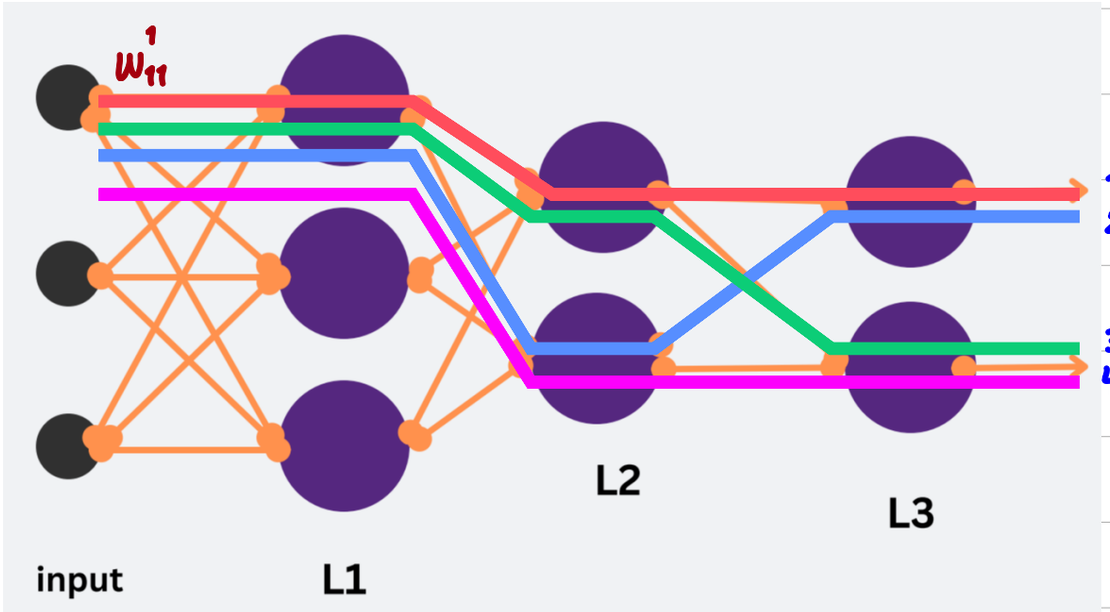
$$O_1 = \text{Sigmoid}(a_1^2 \cdot w_{11}^3 + a_2^2 w_{12}^3 + b_1^3) = \text{Sigmoid}(0.321 \times 0.5 + 0.621 \times 0.1 + 0.5) = \text{Sigmoid}(0.7226) = 0.673$$

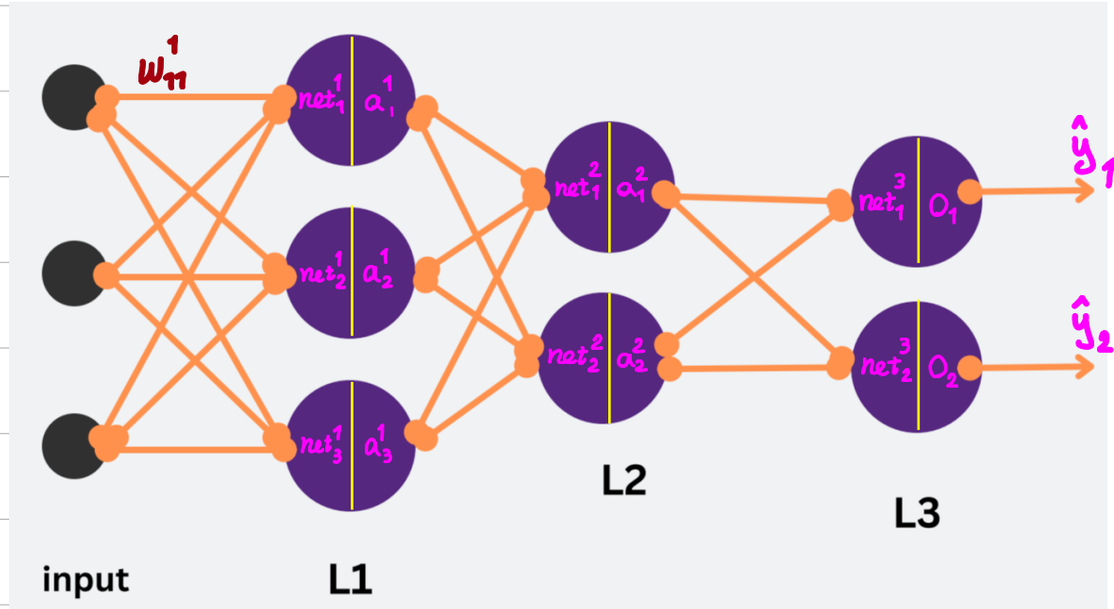
$$O_2 = \text{Sigmoid}(a_1^2 \cdot w_{21}^3 + a_2^2 w_{22}^3 + b_2^3) = \text{Sigmoid}(0.321 \times 0.5 + 0.621 \times 0.5 + 0.3) = \text{Sigmoid}(0.771) = 0.684$$

2. Calculate Loss :

$$\text{Loss} = \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2 \longrightarrow \text{Loss} = ((0.5 - 0.573)^2 + (0.8 - 0.628)^2) \div 2 \approx 0.0174$$

3. Backpropagation :





$$W_{11\ new}^{1} = W_{11\ old}^{1} - \alpha \left(\frac{\partial L}{\partial w_{11}^{1}} \right) \quad \text{← ?}$$

$$\frac{\partial L}{\partial w_{11}^{1}} = 1 + 2 + 3 + 4$$

① $\hat{y}_1 \rightarrow O_1 \rightarrow net_1^3 \rightarrow a_1^2 \rightarrow net_1^2 \rightarrow a_1^1 \rightarrow net_1^1 \rightarrow w_{11}^1$

② $\hat{y}_1 \rightarrow O_1 \rightarrow net_2^3 \rightarrow a_2^2 \rightarrow net_2^2 \rightarrow a_1^1 \rightarrow net_1^1 \rightarrow w_{11}^1$

③ $\hat{y}_2 \rightarrow O_2 \rightarrow net_2^3 \rightarrow a_1^2 \rightarrow net_1^2 \rightarrow a_1^1 \rightarrow net_1^1 \rightarrow w_{11}^1$

④ $\hat{y}_2 \rightarrow O_2 \rightarrow net_2^3 \rightarrow a_2^2 \rightarrow net_2^2 \rightarrow a_1^1 \rightarrow net_1^1 \rightarrow w_{11}^1$

$$\frac{\partial L}{\partial w_{11}^1} = \left[\frac{\partial L}{\partial \hat{y}_1}\frac{\partial \hat{y}_1}{\partial o_1}\frac{\partial o_1}{\partial net_1^3}\frac{\partial net_1^3}{\partial a_1^2}\frac{\partial a_1^2}{\partial net_1^2}\frac{\partial net_1^2}{\partial a_1^1}\frac{\partial a_1^1}{\partial net_1^1}\frac{\partial net_1^1}{\partial w_{11}^1}\right] + \left[\frac{\partial L}{\partial \hat{y}_1}\frac{\partial \hat{y}_1}{\partial o_1}\frac{\partial o_1}{\partial net_2^3}\frac{\partial net_2^3}{\partial a_2^2}\frac{\partial a_2^2}{\partial net_2^2}\frac{\partial net_2^2}{\partial a_1^1}\frac{\partial a_1^1}{\partial net_1^1}\frac{\partial net_1^1}{\partial w_{11}^1}\right] +$$

$$\left[\frac{\partial L}{\partial \hat{y}_2}\frac{\partial \hat{y}_2}{\partial o_2}\frac{\partial o_2}{\partial net_2^3}\frac{net_2^3}{a_1^2}\frac{a_1^2}{net_1^2}\frac{net_1^2}{a_1^1}\frac{a_1^1}{net_1^1}\frac{net_1^1}{w_{11}^1}\right] + \left[\frac{\partial L}{\partial \hat{y}_2}\frac{\partial \hat{y}_2}{\partial o_2}\frac{\partial o_2}{\partial net_2^3}\frac{net_2^3}{a_2^2}\frac{a_2^2}{net_2^2}\frac{net_2^2}{a_1^1}\frac{a_1^1}{net_1^1}\frac{net_1^1}{w_{11}^1}\right]$$

* $ReLU(x) = \begin{cases} x & x>0 \\ 0 & x<0 \end{cases}$

$$\frac{\partial net_1^1}{\partial w_{11}^1} = \frac{\partial}{\partial w_{11}^1}\left(x_1 \cdot w_{11}^1 + \overset{0}{x_2} \cdot w_{12}^1 + x_3 \cdot w_{13}^1 + b_1^1\right) = x_1 = 0.3 \qquad \frac{a_1^1}{net_1^1} = \frac{\partial}{\partial net_1^1}\left(ReLU(\overset{0.49}{net_1^1})\right) = 1$$

$$\frac{\partial net_1^2}{\partial a_1^1} = \frac{\partial}{\partial a_1^1}\left(a_1^1 \cdot w_{11}^2 + \overset{0}{a_2^1} \cdot w_{12}^2 + a_3^1 \cdot w_{13}^2 + b_1^2\right) = w_{11}^2 = 0.1$$

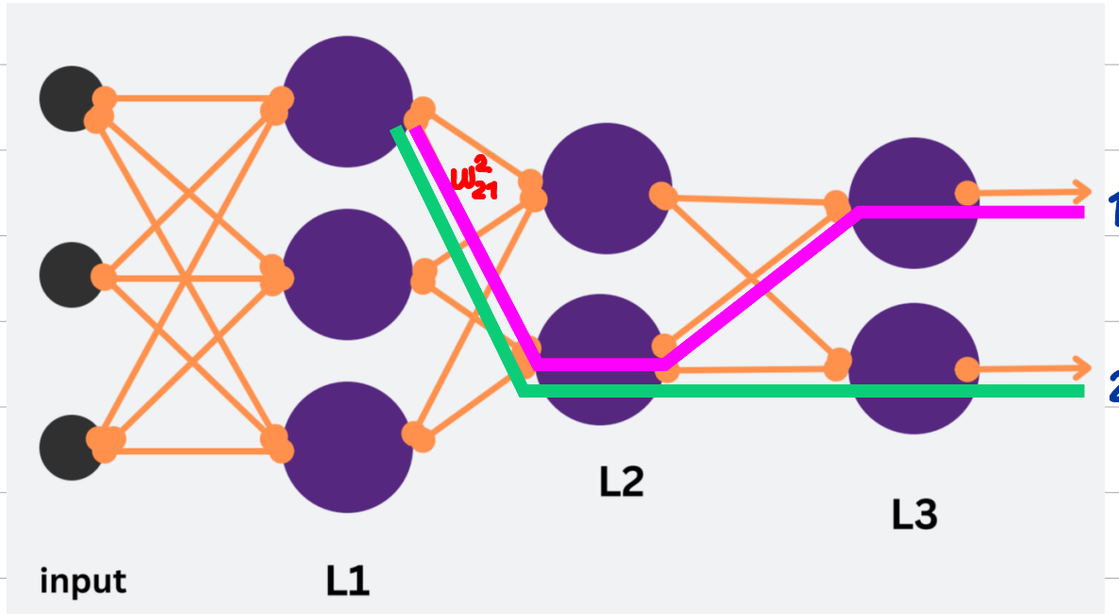
$$\frac{\partial a_1^2}{\partial net_1^2} = \frac{\partial}{\partial net_1^2}\left(\tanh(net_1^2)\right) = 1 - \tanh^2(\overset{0.333}{net_1^2}) = 0.897$$

$$\frac{\partial net_1^3}{\partial a_1^2} = \frac{\partial}{\partial a_1^2}\left(a_1^2 \cdot w_{11}^3 + \overset{0}{a_2^2} w_{12}^3 + b_1^3\right) = w_{11}^3 = 0.5$$

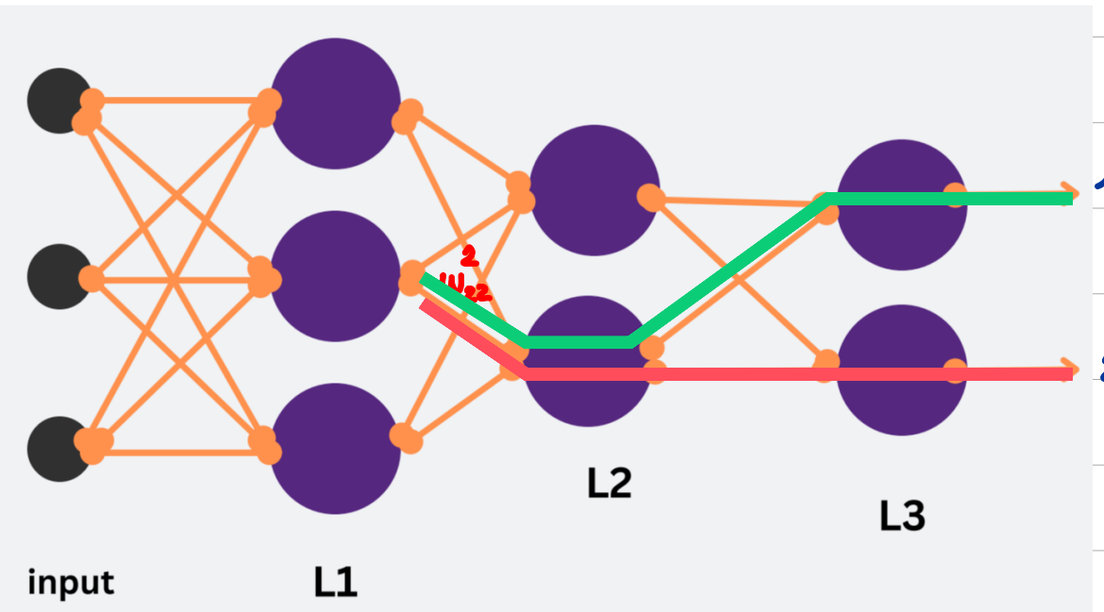
$$\frac{\partial o_1}{net_1^3} = \frac{\partial}{\partial net_1^3}\left(Sigmoid(net_1^3)\right) = Sigmoid(net_1^3).(1 - Sigmoid(net_1^3)) = 0.22$$

$$\frac{\partial \hat{y}_1}{\partial o_1} = 1 \qquad \frac{\partial L}{\partial \hat{y}_1} = \frac{\partial}{\partial \hat{y}_1}\left(\frac{1}{m}(y_1 - \hat{y}_1)^2\right) = \frac{2}{m}(y_1 - \hat{y}_1)(-1) = -\frac{2}{m}(y_1 - \hat{y}_1) = -2(0.5 - 0.673) = 0.346$$

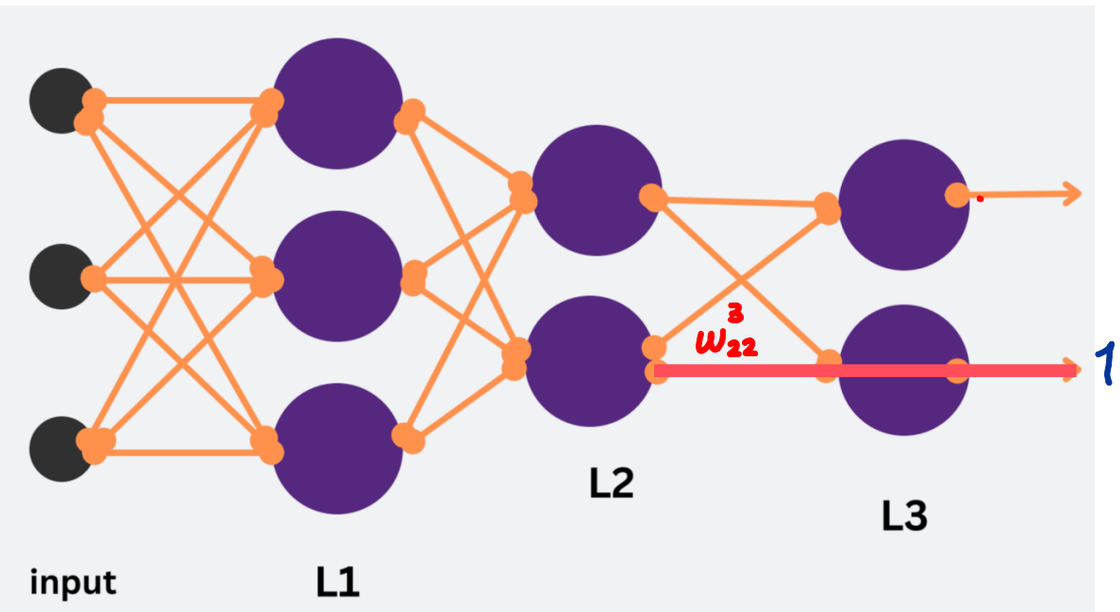
W_{21}^{2}



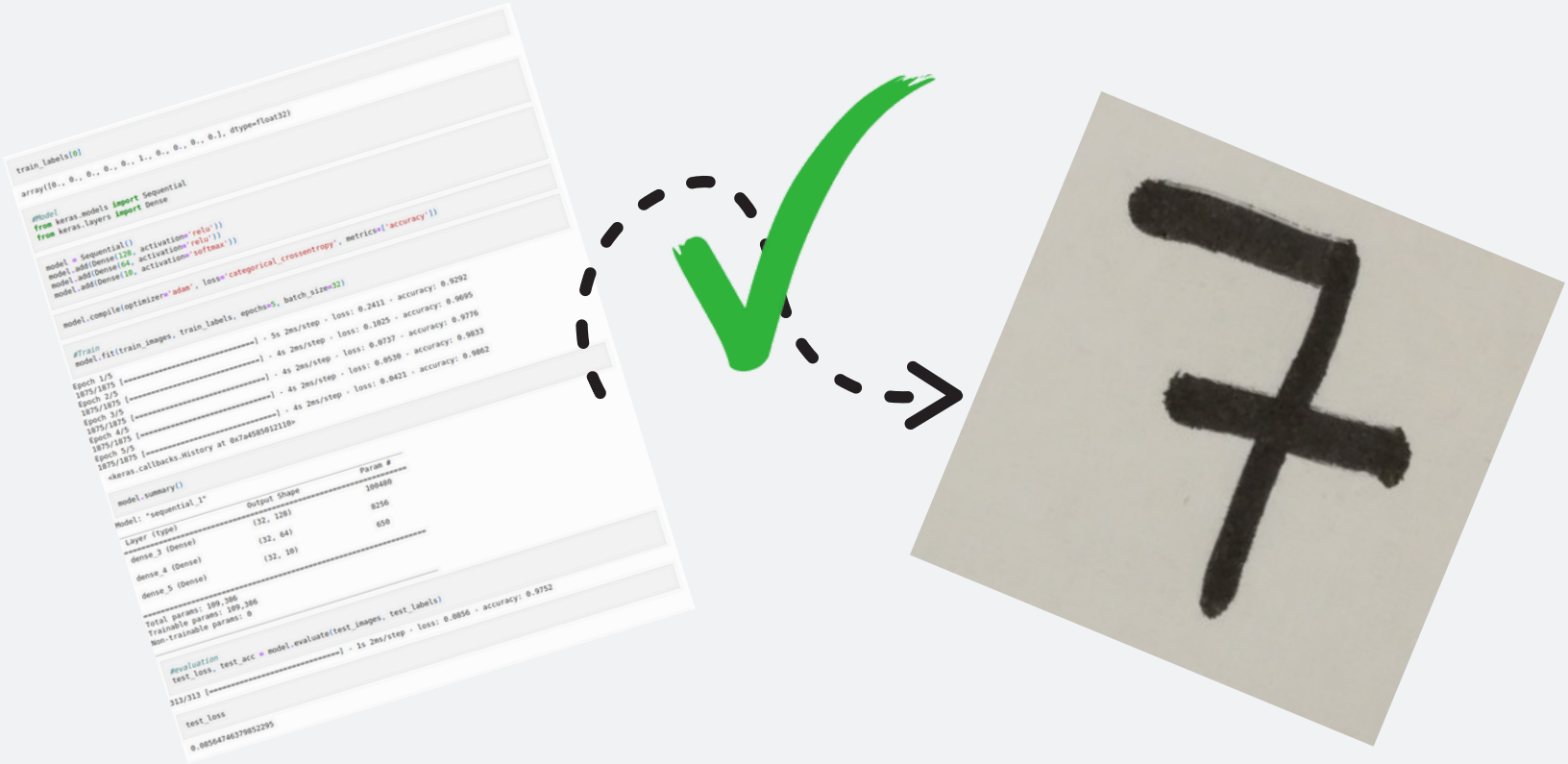
W_{22}^{2}



W_{22}^{3}



2. As you saw in the previous session, the designed model could not predict the input image correctly. Change the code and Design the network using Dense layers so that it can classify the input image (seven.png) correctly.



Improving Model:

1. Increase Model Complexity
2. Hyperparameter Tuning
    1. Learning Rate
    2. Batch Size
    3. Optimizer
    4. Number of Epochs
3. Increase Training Epochs
4. Data Augmentation
5. Model Regularization