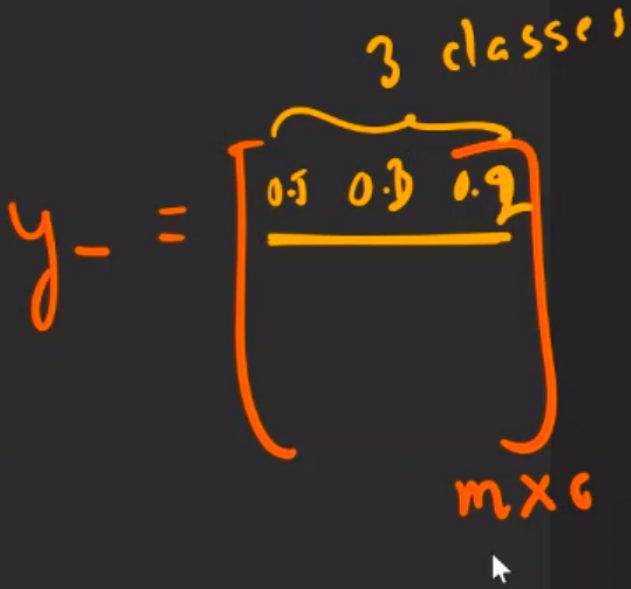
```
        z2 = np.dot(a1,W2) + b2
        a2 = np.tanh(z2)

        z3 = np.dot(a2,W3) + b3
        y_ = softmax(z3)

        self.activation_outputs = (a1,a2,y_)
        return y_

    def backward(self,x,y,learning_rate=0.001):
        W1,W2,W3 = self.model['W1'],self.model['W2'],self.model['W3']
        b1, b2, b3 = self.model['b1'],self.model['b2'],self.model['b3']

        y_ =
        delta3 = y_ - y
```

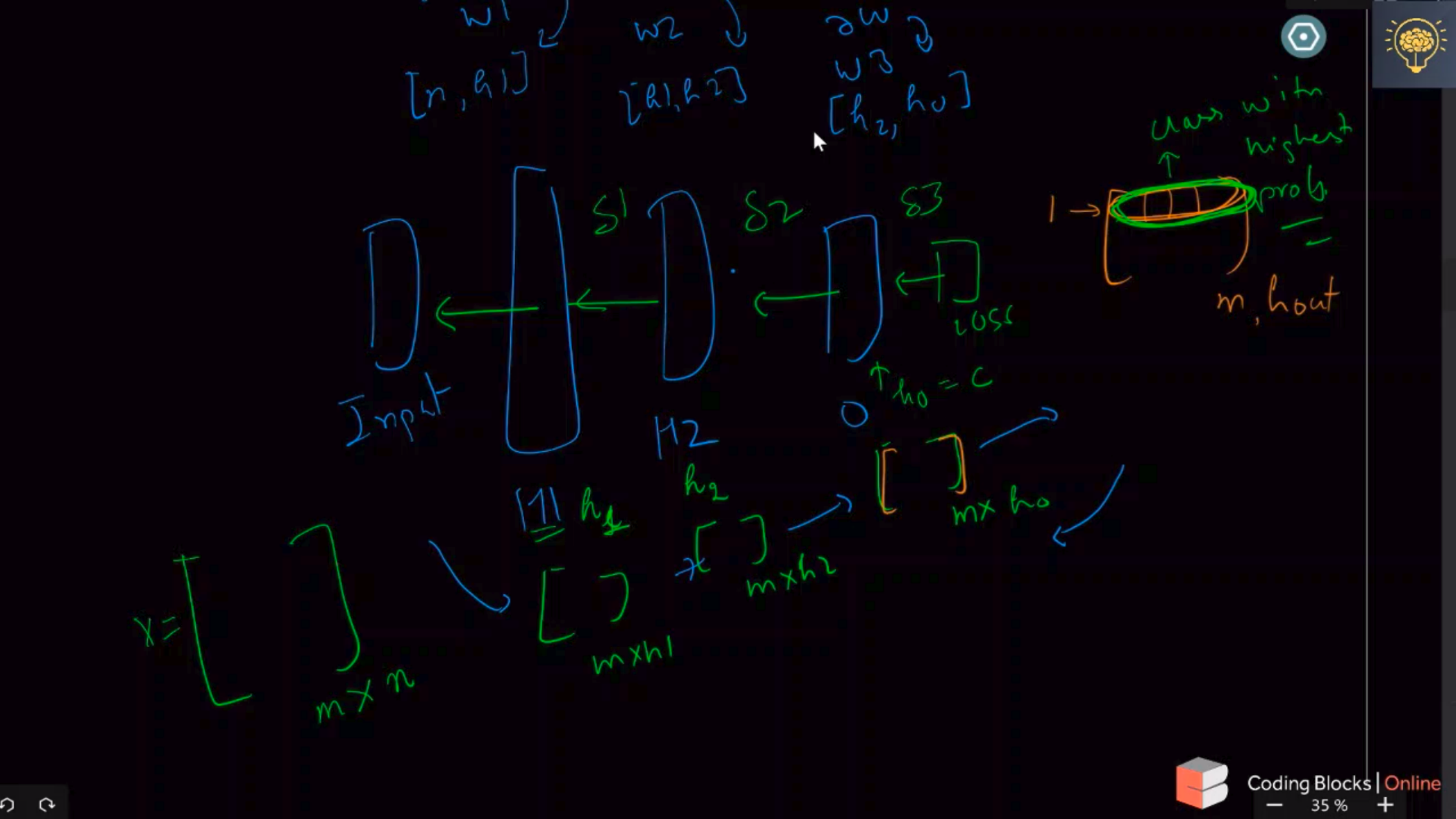
In [ ]:

$$a2 = \tanh(z2)$$

$$1 - \tanh(z2)^2$$

$$1 - a^2$$

rohit58405840@gmail.com



```
        return np.argmax(y_out,axis=1)

    def summary(self):
        W1,W2,W3 = self.model['W1'],self.model['W2'],self.model['W3']
        a1,a2,y_ = self.activation_outputs

        print("W1 ",W1.shape)
        print("A1 ",a1.shape)

        print("W2 ",W2.shape)
        print("A2 ",a2.shape)

        print("W3 ",W3.shape)
        print("Y_ ",y_.shape)
```

$$-\sum_{i=1}^{m}\sum_{c}^{n_i} y_{i,c}\log\hat{y}_{i,c}$$

In [ ]:

```
def loss(y_oht,p):
```

In [ ]: