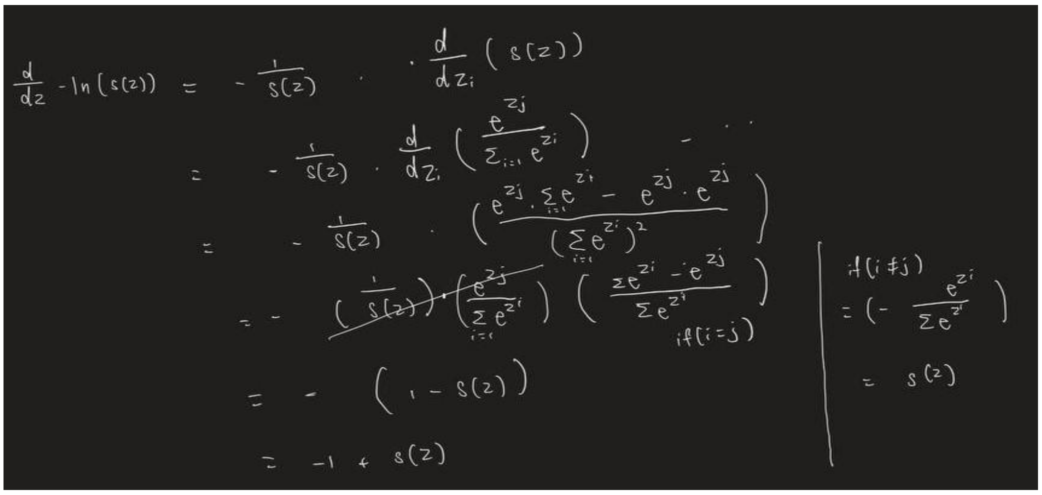
# Machine Learning Hand in 2

**Group 41**
**Tan Wei Yuan (202300860)**
**Ang Yu Juan (202300862)**
**Alton ZhiXian Cheah (202300865)**

## PART I: Derivative



$$\frac{d}{dz} -\ln(s(z)) = -\frac{1}{s(z)} \cdot \frac{d}{dz_i}(s(z))$$

$$= -\frac{1}{s(z)} \cdot \frac{d}{dz_i}\left(\frac{e^{z_j}}{\sum_{i=1} e^{z_i}}\right)$$

$$= -\frac{1}{s(z)} \cdot \left(\frac{e^{z_j} \cdot \sum_{i=1} e^{z_i} - e^{z_j} \cdot e^{z_j}}{(\sum_{i=1} e^{z_i})^2}\right)$$

$$= -\cancel{\left(\frac{1}{s(z)}\right) \cdot \left(\frac{e^{z_j}}{\sum_{i=1} e^{z_i}}\right)} \left(\frac{\sum e^{z_i} - e^{z_j}}{\sum e^{z_i}}\right)$$

if(i=j)

$$= -(1 - s(z))$$

$$= -1 + s(z)$$

if(i≠j)

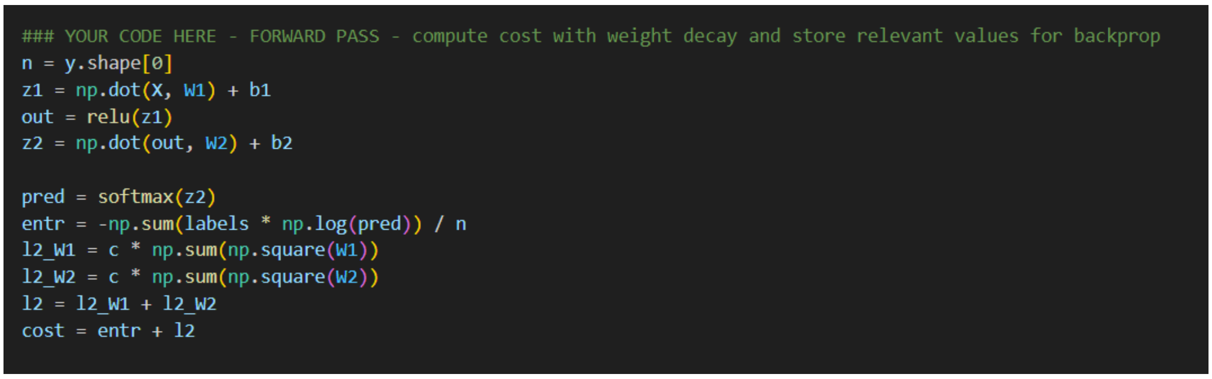
$$= \left(-\frac{e^{z_i}}{\sum e^{z_i}}\right)$$

$$= s(z)$$

Therefore derivative of the negative log likelihood of z gives **-1 + softmax(z)** if i = j and **0 + softmax(z)** if i != j

## PART II: Implementation and Test

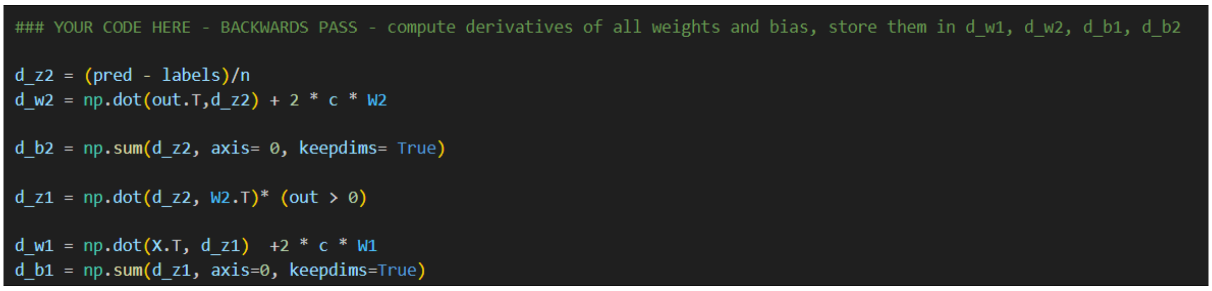
### Forward pass



```
### YOUR CODE HERE - FORWARD PASS - compute cost with weight decay and store relevant values for backprop
n = y.shape[0]
z1 = np.dot(X, W1) + b1
out = relu(z1)
z2 = np.dot(out, W2) + b2

pred = softmax(z2)
entr = -np.sum(labels * np.log(pred)) / n
l2_W1 = c * np.sum(np.square(W1))
l2_W2 = c * np.sum(np.square(W2))
l2 = l2_W1 + l2_W2
cost = entr + l2
```

The cost for the forward pass is the average entropy + the l2 loss wish is the square loss of each weight

### Backward pass



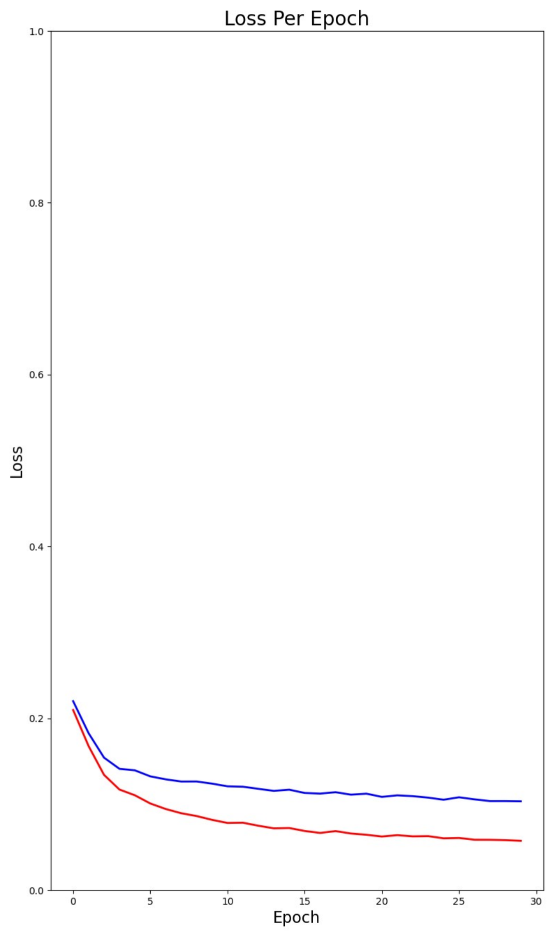
```
### YOUR CODE HERE - BACKWARDS PASS - compute derivatives of all weights and bias, store them in d_w1, d_w2, d_b1, d_b2

d_z2 = (pred - labels)/n
d_w2 = np.dot(out.T,d_z2) + 2 * c * W2

d_b2 = np.sum(d_z2, axis= 0, keepdims= True)

d_z1 = np.dot(d_z2, W2.T)* (out > 0)

d_w1 = np.dot(X.T, d_z1)  +2 * c * W1
d_b1 = np.sum(d_z1, axis=0, keepdims=True)
```

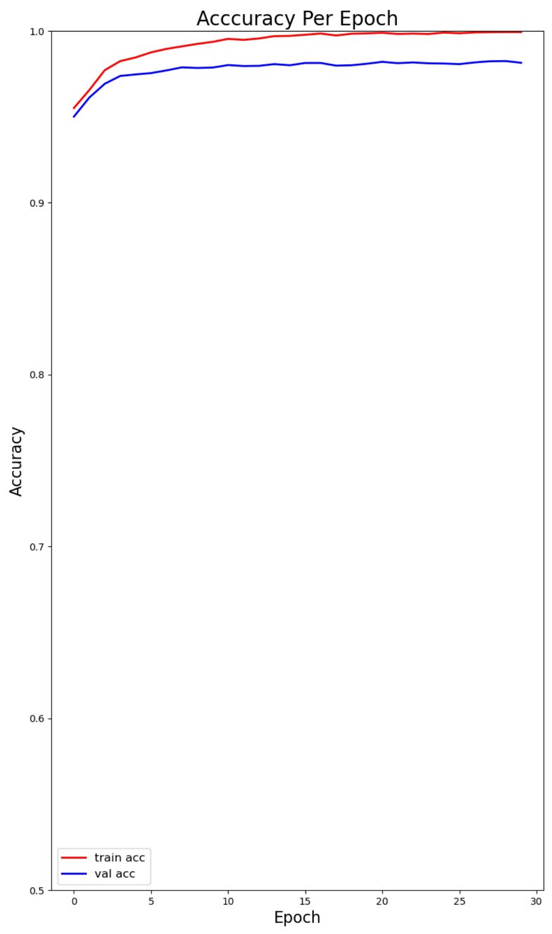
Using the derivative for the **-log(softmax(z))** found in **PART I** , the derivative of z2 becomes **softmax(x)  - label** , hence **pred - label**

### Difficulties

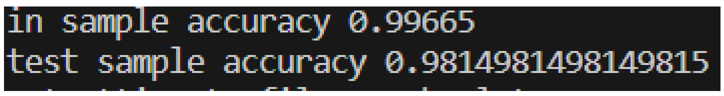
1. Ensuring that the shapes of each element match. W1 and W2 had different shapes which affected our L2 loss calculations
2. Not setting self.params to in the fit function resulted in params being None during the last epoch

## Loss Plots





## Accuracy



```
in sample accuracy 0.99665
test sample accuracy 0.9814981498149815
```

There seems to be no anomalous data observed in our loss and accuracy curves. Our model ends with having a high accuracy and a low training and validation loss.