Data matrix: $X = \begin{bmatrix} — x_1 — \\ \vdots \\ — x_m — \end{bmatrix}$; W_1 matrix $= \begin{bmatrix} — w_1 — \\ \vdots \\ — w_{10} — \end{bmatrix}$; W_2 matrix $= \begin{bmatrix} — w_1^{(2)} — \\ \vdots \\ — w_{10}^{(2)} — \end{bmatrix}$; $b_1 = \begin{bmatrix} b_1^1 \\ \vdots \\ b_{10}^1 \end{bmatrix}$; $b_2 = \begin{bmatrix} b_1^2 \\ \vdots \\ b_{10}^2 \end{bmatrix}$

10×10

Forward Pass

Input → 1st layer

Weights (754 weights) · Samples

Activations

$$\underset{10 \text{ nodes} \downarrow}{\begin{bmatrix} | & & | \\ z_1 & \cdots & z_m \\ | & & | \end{bmatrix}}_{10 \times m} = \underbrace{\begin{bmatrix} — w_1 — \\ \vdots \\ — w_{10} — \end{bmatrix}}_{10 \times 754} \cdot \underbrace{\begin{bmatrix} | & & | \\ x_1 & \cdots & x_m \\ | & & | \end{bmatrix}}_{754 \times m} \downarrow \text{features} + \underbrace{\begin{bmatrix} | & & | \\ b & \cdots & b \\ | & & | \end{bmatrix}}_{10 \times m} \leftarrow \text{same row vector of biases } m \text{ times}$$

Focus on how one sample contributes to all 10 nodes

$$\underbrace{\begin{bmatrix} — w_1 — \\ \vdots \\ — w_{10} — \end{bmatrix}}_{10 \times 754} \underbrace{\begin{bmatrix} | \\ x_1 \\ | \end{bmatrix}}_{754 \times 1} = \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_{10} \end{bmatrix}_{10 \times 1}$$

Activations

$$\underset{10 \text{ nodes} \downarrow}{\begin{bmatrix} | & & | \\ z_1 & \cdots & z_m \\ | & & | \end{bmatrix}}_{10 \times m}$$

Each sample produces its own set of 10 values corresponding to the 10 nodes in layer one.

↑ obviously thats the case. You only combine the collective error of all the samples when computing loss. But for prediction obviously there will be m unique predictions corresponding to the m samples. Each prediction is a list of 10 probabilities indicating likelihood of each possible answer.

$$\underset{10 \times m}{A^{(1)}} = \text{ReLU}\left(\left[\text{activations}\right]\right)$$

↳ Apply ReLU to every node for every sample

Math Form:

$$\underset{10 \times m}{Z^1} = \underset{10 \times m}{W^1} \underset{784 \times m}{X^T} + \underset{(10 \times 1) \to 10 \times m}{b}$$

$$\underset{10 \times m}{A^1} = \text{ReLU}(Z^1) \quad ; \quad A^{(1)}_{k,i} = \max(0, Z^{(1)}_{k,i})$$

(each individual neuron goes through ReLU to determine weather it will contribute to the next layer. * This idea becomes important again during backprop *)

2nd Layer:

Treat Z^1 from previous layer as input for second layer. It works out nicely because the data already arranged as columns

$$\underset{10 \times m}{Z^2} = \underset{10 \times 10}{W^2} \underset{10 \times m}{A^1} + \underset{(10 \times 1) \to 10 \times m}{b^2}$$

Same 10×1 column m times

$$\underbrace{\begin{bmatrix} | & & | \\ z_1^{(2)} & \cdots & z_m^{(2)} \\ | & & | \end{bmatrix}}_{10 \times m} = \underbrace{\begin{bmatrix} — w_1^{(2)} — \\ \vdots \\ — w_{10}^{(2)} — \end{bmatrix}}_{10 \times 10} \underbrace{\begin{bmatrix} | & & | \\ A_1^{(1)} & \cdots & A_m^{(1)} \\ | & & | \end{bmatrix}}_{10 \times m} + \underbrace{\begin{bmatrix} | & & | \\ b^{(2)} & \cdots & b^{(2)} \\ | & & | \end{bmatrix}}_{10 \times m}$$

Final output

$$A^{(2)} = \text{Softmax}\left(\left[\text{activations}^{(2)}\right]\right)$$

$A^{(2)}$ in the form of probabilities

↳ Pick largest one to make decision

Backpropigation

1. Compute loss function

one hot encode labels

ex) $\begin{bmatrix}0\\0\\1\\0\end{bmatrix}$ ← correct answer

Calculate: sparse-categorical cross entropy loss (for one sample)

↳ The labels will not be one hot-encoded

$$\text{Loss}^{(i)} = -\log\left(\hat{y}^{(i)}_{y^{(i)}}\right)$$

↑ The predicted softmax prob. for the correct label.

• $\hat{y}$ = predicted softmax vector (A^2 in forward pass notation)

$$\text{Cost} = -\frac{1}{m}\sum_{i=1}^{m}\log\left(\hat{y}^{(i)}_{y^{(i)}}\right)$$

Cost in my notation:

$$\text{cost} = -\frac{1}{m}\sum_{i=1}^{m}\log\left(A^{2\,(i)}_{y^{(i)}}\right)$$ ← prediction for sample i corresponds to ith column of A^2.

$$\text{loss}^{i} = -\log\left(A^{2\,(i)}_{y^{(i)}}\right)$$; loss computed for ith sample

Gradient:

*** The math for gradient step done as though you had one-hot labels. But in code no need to form full 10×m Y matrix of one-hot-encoded vector for each label. Instead for each sample you just subtract 1 at the true-class labels index. See, no need to store a 10×m Y matrix.

$$\frac{\partial C}{\partial z^2} = \frac{1}{m}(A^2 - Y)$$

↳ in this expression this is a 10×m matrix with the each col being one-hot encoded label matrix for a given sample. Code implimentation will not use this!

Update W_2:

$$\frac{\partial C}{\partial W^2} = \frac{\partial C}{\partial z^2}\cdot\frac{\partial z^2}{\partial W^2} = \frac{1}{m}(A^2 - Y)\bullet(A^1)$$

Grad descent: $\underset{10\times10}{W^{2\,new}} = \underset{10\times10}{W^{2\,old}} - \eta\nabla C$

$$\underset{10\times m}{\frac{1}{m}(A^2 - Y)}\bullet\underset{10\times m}{(A^1)} \neq \underset{10\times10}{W^2} \rightarrow \boxed{\underset{10\times m}{\frac{1}{m}(A^2 - Y)}\underset{m\times10}{(A^1)^T}} = \underset{10\times10}{W^2}$$

Update b^2:

$$\frac{\partial C}{\partial b^2} = \frac{\partial C}{\partial z^2}\cdot\frac{\partial z^2}{\partial b^2} = \frac{1}{m}(A^2 - Y)(1)$$

Recall the b^2 matrix was formed by repeating the same b vector m times. We need to aggregate the contribution to gradient from every sample into a single (10×1) vector to match dimension of b^2 which is really just a 10×1 vector. Contribution of each sample to grad w/ respect to b_2 is stored as col of $\frac{1}{m}(A^2 - y)$ computed above.

$$\frac{\partial C}{\partial b^2} = \frac{1}{m}\sum_{i=1}^{m}\left(A^2_{:,i} - Y_{:,i}\right)$$

↳ means add up the cols of $(A^2 - Y)$ to get from 10×m to 10×1 vector.

Update W_1:

$$\frac{\partial C}{\partial W^1} = \frac{\partial C}{\partial Z^2} \cdot \frac{\partial Z^2}{\partial A^1} \cdot \frac{\partial A^1}{\partial Z^1} \cdot \frac{\partial Z^1}{\partial W^1}$$

1. $dZ^{(2)} = \frac{1}{m}(A^{(2)} - Y)$ ex) $dZ^2 = \frac{\partial C}{\partial Z^2}$
2. $dA^{(1)} = W_2^T \, dZ^{(2)}$
3. $dZ^{(1)} = dA^{(1)} \odot (Z^{(1)} > 0)$ (element-wise ReLU mask)
4. $dW_1 = \frac{1}{m} dZ^{(1)} X$

ReLU:

$A^1_{k,i} = \max(0, Z^1_{k,i})$ turns off nuerons that won't contribute to next layer.

ReLU mask:

Why apply the mask and what does it actually do?

Why:

When you back-propagate through a nonlinearity like ReLU, you need to ask "for each neuron, did it actually contribute to the forward pass?" The ReLU mask is how you answer that.

Derivative of ReLU:

$$\frac{d}{dZ_{k,i}} \max(0, Z_{k,i}) = \begin{cases} 1, & Z_{k,i} > 0 \\ 0, & Z_{k,i} \leq 0 \end{cases}$$

↳ This restricts gradient flow to the nuerons that contributed to the second layer during forward pass.

The mask:

$(Z^{(1)} > 0)$ is a 10 x m matrix of 0's and 1's indicating active or off nuerons. A bit like A^1 but instead of having $Z_{k,i}$ where $Z_{k,i} > 0$ we have a 1.

What does the mask do?

You've already got the "raw" gradient into that layer, $dA^{(1)}$. To turn it into the gradient w.r.t. the pre-activation $Z^{(1)}$, you do an **elementwise** multiply:

$$dZ^{(1)} = dA^{(1)} \odot (Z^{(1)} > 0).$$

- Wherever the mask is 1, $dZ^{(1)}_{k,i} = dA^{(1)}_{k,i}$ (gradient flows unchanged). ← means gradient flows back to weights of nuerons that contributed to the forward propigation only.
- Wherever the mask is 0, $dZ^{(1)}_{k,i} = 0$ (no gradient flows back into that neuron).

$$\frac{\partial C}{\partial W^1} = \frac{\partial C}{\partial Z^2} \cdot \frac{\partial Z^2}{\partial A^1} \cdot \frac{\partial A^1}{\partial Z^1} \cdot \frac{\partial Z^1}{\partial W^1} = \underbrace{\frac{1}{m}}_{10\times 10}\Big[\underbrace{(W^2)^T}_{10\times 10}\underbrace{(A^2 - Y)}_{10\times m} \odot \underbrace{(Z^1 > 0)}_{10\times m}\Big]\underbrace{(X)}_{m\times 784}$$

Note: W_1 is 10×784

1. $dZ^{(2)} = \frac{1}{m}(A^{(2)} - Y)$ ex) $dZ^2 = \frac{\partial C}{\partial Z^2}$
2. $dA^{(1)} = W_2^T \, dZ^{(2)}$
3. $dZ^{(1)} = dA^{(1)} \odot (Z^{(1)} > 0)$ (element-wise ReLU mask)
4. $dW_1 = \frac{1}{m} dZ^{(1)} X$

Update b_1:

$$\frac{\partial C}{\partial b^1} = \frac{\partial C}{\partial Z^2} \cdot \frac{\partial Z^2}{\partial A^1} \cdot \frac{\partial A^1}{\partial Z^1} \cdot \frac{\partial Z^1}{\partial b^1}$$

$$\frac{\partial C}{\partial b^1} = \sum_{i=1}^{m} \underbrace{\frac{1}{m}}_{10\times 10}\Big[\underbrace{(W^2)^T}_{10\times 10}\underbrace{(A^2 - Y)}_{10\times m} \odot \underbrace{(Z^1 > 0)}_{10\times m}\Big]$$

(Sum up each column of this matrix leaving you with a 10x1 matrix.)

Additional Notes:

1. Be attentitive to the notation used for data matrix. Samson Zhang video uses X where cols are samples not rows.
2. Pay attention to summation vs matrix notation.