#2 2) I will backpropagation in conjunction with a forward pass to update the parameters and ultimately perform training.

$$X = \text{input}$$
$$z = W^1 x + B^1$$
$$h = \text{ReLU}(z) = \max(0, z)$$
$$\theta = W^2 H + B^2$$
$$\hat{y} = \text{softmax}(\theta) = \frac{e^{z_j}}{\sum_{i=1}^{c} e^{z_i}}$$
$$J = \text{cross-entropy}(y, \hat{y}) = -\sum_i y_i \log(\hat{y}_i)$$

Note: I follow the convention in the online notes and transpose the final result so that shape of the gradient equals shape of the parameter.

$$\frac{\partial J}{\partial W^2} = \frac{\partial J}{\partial \theta} \cdot \frac{\partial \theta}{\partial W^2} = (\hat{y} - y)^T \cdot \frac{\partial \theta}{\partial W^2} = (\hat{y} - y)^T \cdot h^T$$

$$\frac{\partial J}{\partial B^2} = \frac{\partial J}{\partial \theta} \cdot \frac{\partial \theta}{\partial B^2} = (\hat{y} - y)^T \cdot \frac{\partial \theta}{\partial B^2} = (\hat{y} - y)$$

$$\frac{\partial J}{\partial W^1} = \frac{\partial J}{\partial \theta} \cdot \frac{\partial \theta}{\partial h} \cdot \frac{\partial h}{\partial z} \cdot \frac{\partial z}{\partial W^1} = \frac{\partial J}{\partial z} \cdot \frac{\partial z}{\partial W^1}$$

$$\frac{\partial J}{\partial z} = (\hat{y} - y)^T \cdot W^2 \circ \text{sgn}(h) = \sigma^1$$

$$= (\sigma^1)^T \cdot x^T \quad \text{(using Jacobian identity 5)}$$

$$\frac{\partial J}{\partial B^1} = \frac{\partial J}{\partial z} \frac{\partial z}{\partial B^1} = \underbrace{\sigma^1}_{1 \times D} \underbrace{\cancel{\frac{\partial z}{\partial B^1}}}_{\text{identity matrix with shape} = D \times D}^{1} = (\sigma^1)^T \quad \text{(using third identity)}$$

$$\frac{\partial J}{\partial x} = \frac{\partial J}{\partial z} \frac{\partial z}{\partial x} = \underbrace{\sigma^1}_{1 \times D} \underbrace{W^1}_{D \times D} = (\sigma^1 W)^T$$

Using the above derivatives we can perform backpropagation to update the parameters. The training procedure in psuedo code follows in the text document.