

- (a) Derive the gradient with respect to the input of a softmax function when cross entropy loss is used for evaluation, i.e., find the gradients with respect to the softmax input vector θ , when the prediction is made by $\hat{y} = \text{softmax}(\theta)$. Cross entropy and softmax are defined as:

$$\text{CE}(\mathbf{y}, \hat{\mathbf{y}}) = - \sum_i y_i \cdot \log(\hat{y}_i)$$

$$\text{softmax}(\theta)_i = \frac{\exp(\theta_i)}{\sum_j \exp(\theta_j)}$$

The gold vector \mathbf{y} is a one-hot vector, and the predicted vector $\hat{\mathbf{y}}$ is a probability distribution over the output space.

y is one hot vec.

$$L = \text{CE}(\mathbf{y}, \text{softmax}(\theta)) = - \sum_i y_i \log \left(\frac{\exp(\theta_i)}{\sum_j \exp(\theta_j)} \right) = - \log \left(\frac{\exp(\theta_k)}{\sum_j \exp(\theta_j)} \right) = - \theta_k + \log \left(\sum_j \exp(\theta_j) \right)$$

$\therefore k = t$ \rightarrow $\log \sum_j \exp(\theta_j)$

$$\frac{\partial L}{\partial \theta_t} = -1 + \frac{\exp(\theta_t)}{\sum_j \exp(\theta_j)} = \text{softmax}(\theta_t) - 1$$

$\therefore k \neq t$

$$\frac{\partial L}{\partial \theta_t} = \frac{\exp(\theta_t)}{\sum_j \exp(\theta_j)} = \text{softmax}(\theta_t)$$

$$\frac{\partial L}{\partial \theta} = \text{softmax}(\theta) - \mathbf{y} = \hat{\mathbf{y}} - \mathbf{y}$$

Derive the gradients with respect to the input \mathbf{x} in a one-hidden-layer neural network (i.e., find $\frac{\partial J}{\partial \mathbf{x}}$, where J is the cross entropy loss $\text{CE}(\mathbf{y}, \hat{\mathbf{y}})$). The neural network employs a sigmoid activation function for the hidden layer, and a softmax for the output layer. Assume a one-hot label vector \mathbf{y} is used. The network is defined as:

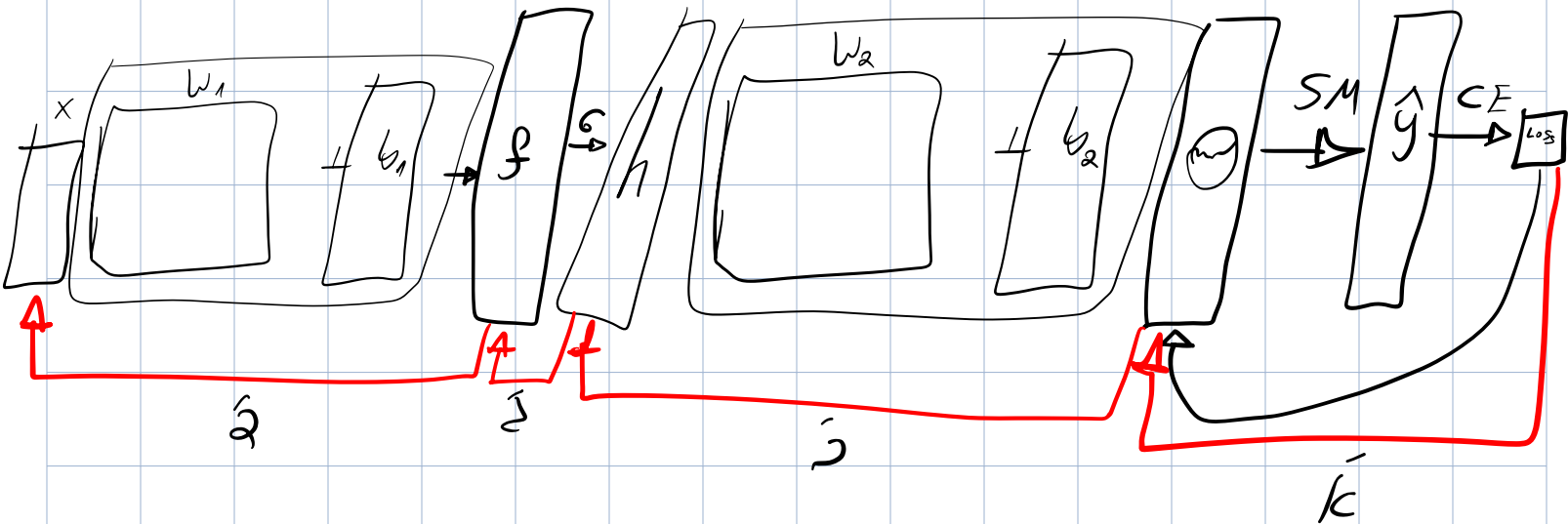
$$\mathbf{h} = \sigma(\mathbf{x}\mathbf{W}_1 + \mathbf{b}_1),_{\theta}$$

$$\hat{\mathbf{y}} = \text{softmax}(\mathbf{h}\mathbf{W}_2 + \mathbf{b}_2).$$

$$\sigma'(x) = \sigma(x)\sigma(1-x)$$

The dimensions of the vectors and matrices are $\mathbf{x} \in \mathbb{R}^{1 \times D_x}$, $\mathbf{h} \in \mathbb{R}^{1 \times D_h}$, $\hat{\mathbf{y}} \in \mathbb{R}^{1 \times D_y}$, $\mathbf{y} \in \mathbb{R}^{1 \times D_y}$.

The dimensions of the parameters are $\mathbf{W}_1 \in \mathbb{R}^{D_x \times D_h}$, $\mathbf{W}_2 \in \mathbb{R}^{D_h \times D_y}$, $\mathbf{b}_1 \in \mathbb{R}^{1 \times D_h}$, $\mathbf{b}_2 \in \mathbb{R}^{1 \times D_y}$.



$$k - \frac{\partial J}{\partial \theta} = \hat{\mathbf{y}} - \mathbf{y}$$

$$z - \frac{\partial \theta}{\partial \mathbf{h}} = \mathbf{W}_2^T$$

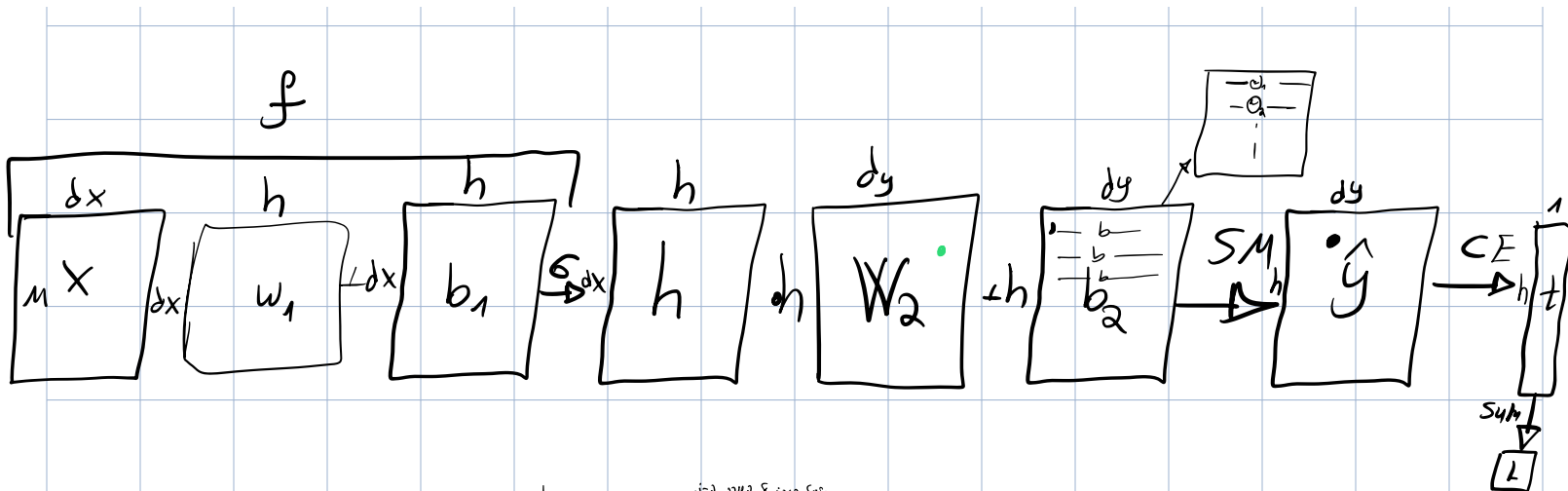
$$z - \frac{\partial \mathbf{h}}{\partial f} = \mathbf{h}(1-\mathbf{h})$$

$$q - \frac{\partial f}{\partial \mathbf{x}} = \mathbf{W}_1^T$$

$$\sigma(f)$$

$$\frac{\partial J}{\partial \mathbf{x}} = \frac{\partial J}{\partial \theta} \cdot \frac{\partial \theta}{\partial \mathbf{h}} \cdot \frac{\partial \mathbf{h}}{\partial f} \cdot \frac{\partial f}{\partial \mathbf{x}} = (\hat{\mathbf{y}} - \mathbf{y}) \cdot \mathbf{W}_2^T \cdot \mathbf{h}(1-\mathbf{h}) \cdot \mathbf{W}_1^T$$

- (c) Implement the forward and backward passes for a neural network with one sigmoid hidden layer. Fill in your implementation in `q1c_neural.py`. Sanity check your implementation with `python q1c_neural.py`.



$$L = CE(\hat{y}, \text{softmax}(\theta)) = \sum_{i=1}^h -\log \left(\frac{\exp(\theta_{i, \text{label}[a]})}{\sum_j \exp(\theta_{i,j})} \right) = \sum_{i=1}^h \left[q \left(\sum_j \exp(\theta_{i,j}) \right) - w_{i, \text{label}[a]} \right]$$

$$\frac{\partial L}{\partial \theta_{ab}} = \begin{cases} \frac{\exp(\theta_{a,b})}{\sum_j \exp(\theta_{a,j})} - 1 & b = \text{label}[a] \\ \frac{\exp(\theta_{a,b})}{\sum_j \exp(\theta_{a,j})} & b \neq \text{label}[a] \end{cases} = \begin{cases} \text{SM}(\theta_{a,b}) - 1 & b = \text{label}[a] \\ \text{SM}(\theta_{a,b}) & b \neq \text{label}[a] \end{cases} = (\hat{y} - y)_{a,b}$$

$$\frac{\partial L}{\partial b^2} = \sum_{i=1}^h \sum_{j=1}^{dy} \frac{\partial L}{\partial \theta_{ij}} \cdot \frac{\partial \theta_{ij}}{\partial b^2} = \sum_{i=1}^h \frac{\partial L}{\partial \theta_{ij}} \cdot \frac{\partial \theta_{ij}}{\partial b^2} = \sum_{i=1}^h (\hat{y} - y)_{ij} = \sum_{i=1}^h \hat{y}_{ij} - y_{ij} \Rightarrow \frac{\partial L}{\partial b^2} = \sum_{i=1}^h \hat{y}_{ij} - y_{ij}$$

$$\frac{\partial \theta_{ij}}{\partial w_{ab}^2} = \begin{cases} h_{i,j} & j=b \\ 0 & \text{else} \end{cases}, \quad \frac{\partial L}{\partial w_{ab}^2} = \sum_{i=1}^h \sum_{j=1}^{dy} \frac{\partial L}{\partial \theta_{ij}} \cdot \frac{\partial \theta_{ij}}{\partial w_{ab}^2} = \sum_{i=1}^h \frac{\partial L}{\partial \theta_{ij}} \cdot \frac{\partial \theta_{ij}}{\partial w_{ab}^2} = \sum_{i=1}^h (\hat{y} - y)_{ij} \cdot h_{iz} = \sum_{i=1}^h h_{zi}^T (\hat{y} - y)_{ij} \Rightarrow \frac{\partial L}{\partial w_2} = h^T \cdot (\hat{y} - y)$$

$$\frac{\partial \theta_{ij}}{\partial w_{ab}} = \begin{cases} w_{bi} & a=i \\ 0 & a \neq i \end{cases}, \quad \frac{\partial L}{\partial w_{ab}} = \sum_{i=1}^h \sum_{j=1}^{dy} \frac{\partial L}{\partial \theta_{ij}} \cdot \frac{\partial \theta_{ij}}{\partial w_{ab}} = \sum_{j=1}^{dy} \frac{\partial L}{\partial \theta_{aj}} \cdot \frac{\partial \theta_{aj}}{\partial w_{ab}} = \sum_{j=1}^{dy} \frac{\partial L}{\partial \theta_{aj}} \cdot w_{bj} = \sum_{j=1}^{dy} (\hat{y} - y)_{aj} \cdot w_{jb}^T = \left[(\hat{y} - y) \cdot w_2^T \right]_{a,b} \Rightarrow \frac{\partial L}{\partial h} = \hat{y} - y \cdot w_2^T$$

$$\frac{\partial h}{\partial f} = \text{sg}(h), \quad \frac{\partial L}{\partial f} = \hat{y} - y \cdot w_2^T \cdot \text{sg}(h)$$

$$\frac{\partial L}{\partial b^1} = \sum_{i=1}^h \sum_{j=1}^{dy} \frac{\partial L}{\partial f_{ij}} \cdot \frac{\partial f_{ij}}{\partial b^1} = \sum_{i=1}^h \frac{\partial L}{\partial f_{it}} \cdot \frac{\partial f_{it}}{\partial b^1} = \sum_{i=1}^h (\hat{y} - y \cdot w_2^T \cdot \text{sg}(h))_{it} \Rightarrow \frac{\partial L}{\partial b^1} = \sum_{i=1}^h (\hat{y} - y \cdot w_2^T \cdot \text{sg}(h))_i = \text{hp.sum}(\hat{y} - y \cdot w_2^T \cdot \text{sg}(h), \text{axis}=0)$$

$$\frac{\partial f}{\partial w_1} = X^T \cdot \hat{y} - y \cdot w_2^T \cdot \text{sg}(h)$$

