(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 $\boldsymbol{\theta}$, when the prediction is made by $\hat{\boldsymbol{y}} = \text{softmax}(\boldsymbol{\theta})$. Cross entropy and softmax are defined as:

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

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

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

y is one hot vec.

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

$k = t$:נגזור לפי

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

$k \neq t$:

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

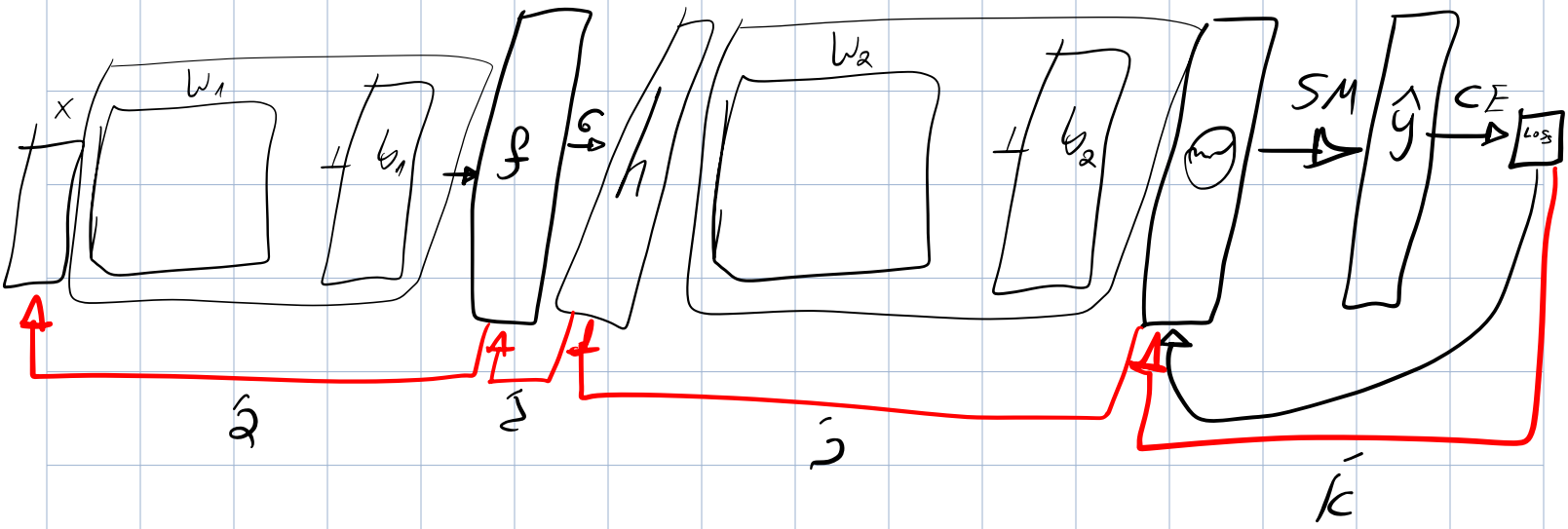
$$\frac{\partial L}{\partial \theta} = Softmax(\theta) - y = \hat{y} - y.$$

Derive the gradients with respect to the input $\boldsymbol{x}$ in a one-hidden-layer neural network (i.e., find $\frac{\partial J}{\partial \boldsymbol{x}}$, where J is the cross entropy loss $\text{CE}(\boldsymbol{y}, \hat{\boldsymbol{y}})$). 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 $\boldsymbol{y}$ is used. The network is defined as:

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

$$\boldsymbol{h} = \sigma(\boldsymbol{x}\boldsymbol{W}_1 + \boldsymbol{b}_1),$$
$$\hat{\boldsymbol{y}} = \text{softmax}(\underbrace{\boldsymbol{h}\boldsymbol{W}_2 + \boldsymbol{b}_2}_{\theta}).$$

The dimensions of the vectors and matrices are $\boldsymbol{x} \in \mathbb{R}^{1\times D_x}, \boldsymbol{h} \in \mathbb{R}^{1\times D_h}, \hat{\boldsymbol{y}} \in \mathbb{R}^{1\times D_y}, \boldsymbol{y} \in \mathbb{R}^{1\times D_y}$. The dimensions of the parameters are $\boldsymbol{W}_1 \in \mathbb{R}^{D_x\times D_h}, \boldsymbol{W}_2 \in \mathbb{R}^{D_h\times D_y}, \boldsymbol{b}_1 \in \mathbb{R}^{1\times D_h}, \boldsymbol{b}_2 \in \mathbb{R}^{1\times D_y}$.



1. $\frac{\partial J}{\partial \theta} = \hat{y} - y$

2. $\frac{\partial \theta}{\partial h} = W_2^T$

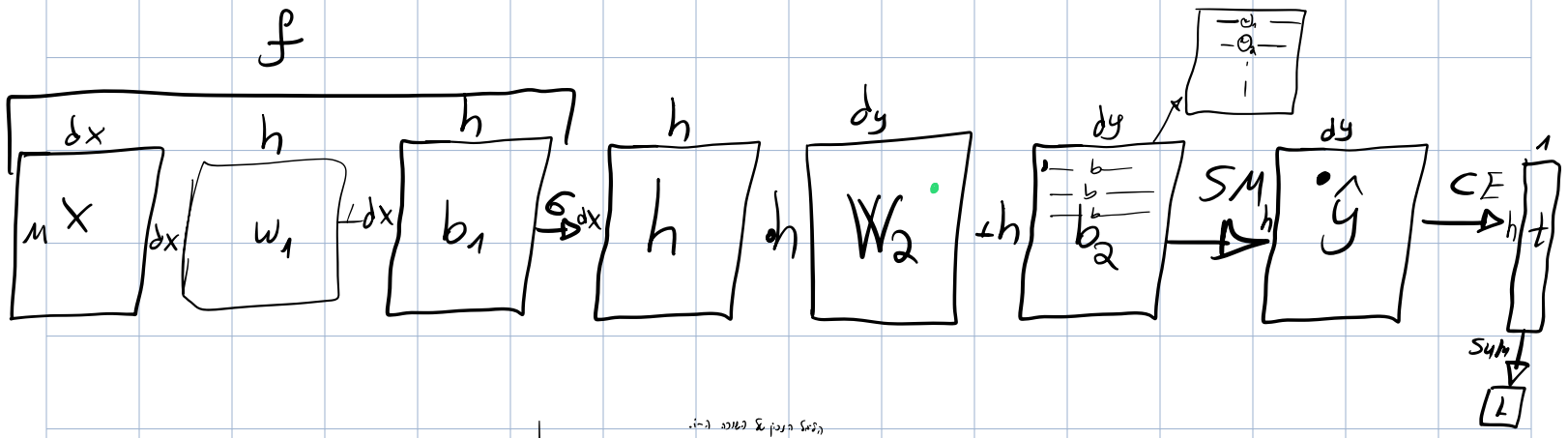
3. $\frac{\partial h}{\partial f} = h(1-h)$

4. $\frac{\partial f}{\partial x} = W_1^T$

$\sigma(f)$

$$\frac{\partial J}{\partial x} = \frac{\partial J}{\partial \theta} \cdot \frac{\partial \theta}{\partial h} \cdot \frac{\partial h}{\partial f} \cdot \frac{\partial f}{\partial x} = (\hat{y} - y) \cdot W_2^T \cdot h(1-h) \cdot W_1^T$$

(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`.



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

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

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

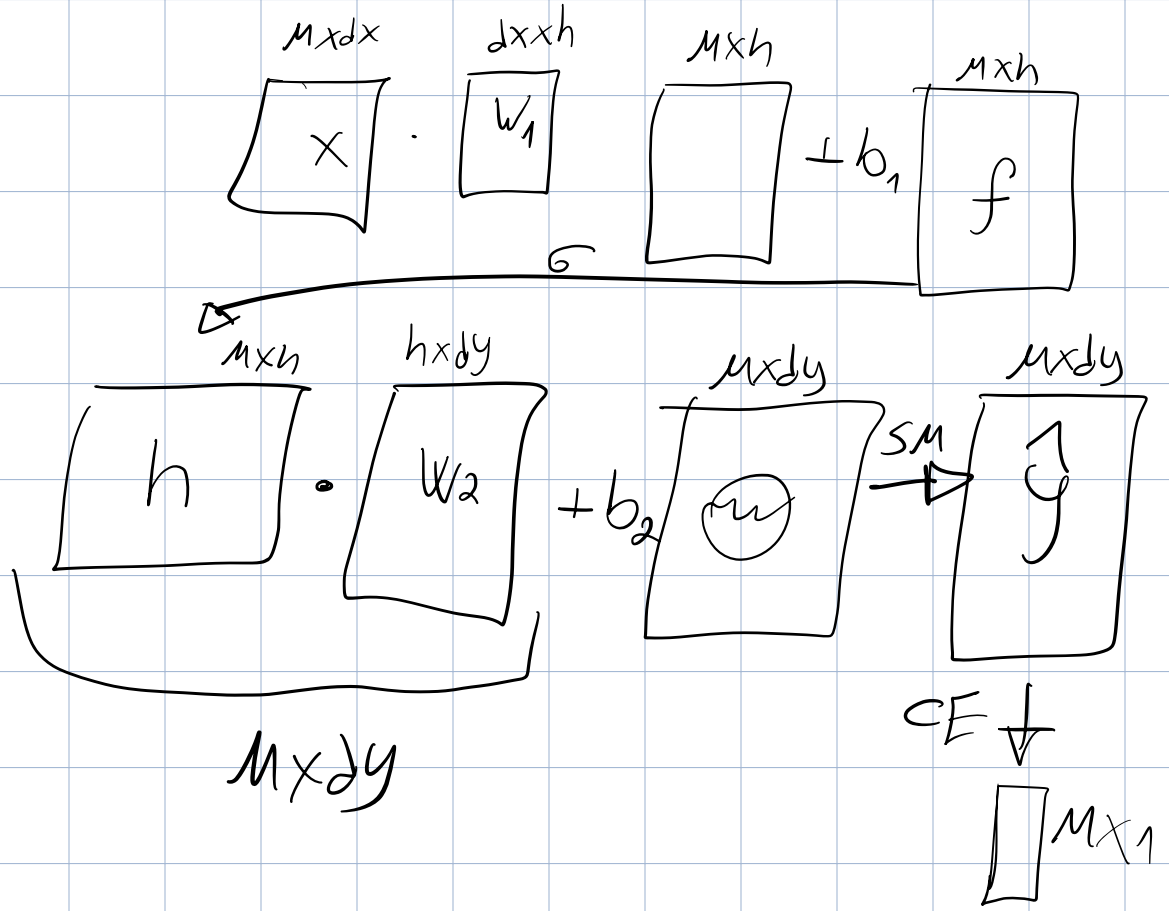
$$\frac{\partial \theta_{ij}}{\partial W_{ab}^2} = \begin{cases} h_{ij} & j = b \\ 0 & else \end{cases}, \quad \frac{\partial L}{\partial W_{fz}^2} = \sum_{i=1}^{h}\sum_{j=1}^{dy} \frac{\partial L}{\partial \theta_{ij}} \cdot \frac{\partial \theta_{ij}}{\partial W_{fz}^2} = \sum_{i=1}^{h} \frac{\partial L}{\partial \theta_{iz}} \cdot \frac{\partial \theta_{iz}}{\partial W_{fz}^2} = \sum_{i=1}^{h} (\hat{y} - y)_{i,f} \cdot h_{iz} = \sum_{i=1}^{h} h_{zi}^T (\hat{y} - y)_{i,f} \Rightarrow \frac{\partial L}{\partial W_2} = h^T \circ (\hat{y} - y)$$

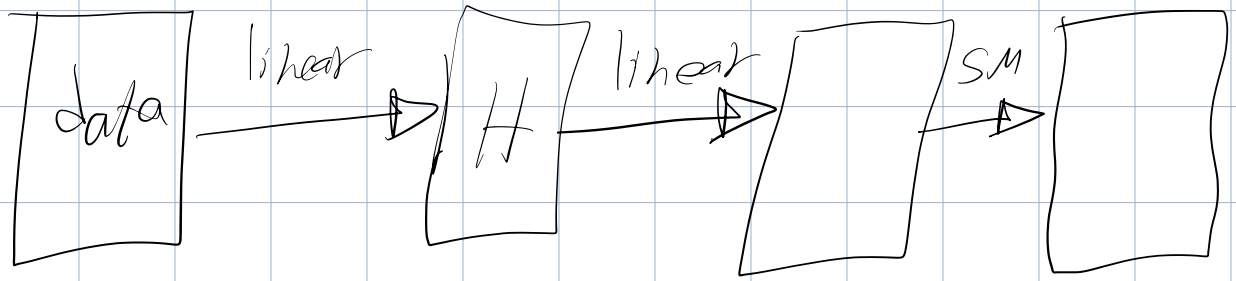
$$\frac{\partial \theta_{ij}}{\partial h_{ab}} = \begin{cases} W_{bj} & a = i \\ 0 & a \neq i \end{cases}, \quad \frac{\partial L}{\partial h_{ab}} = \sum_{i=1}^{h}\sum_{j=1}^{dy} \frac{\partial L}{\partial \theta_{ij}} \cdot \frac{\partial \theta_{ij}}{\partial h_{ab}} = \sum_{j=1}^{dy} \frac{\partial L}{\partial \theta_{aj}} \cdot \frac{\partial \theta_{aj}}{\partial h_{ab}} = \sum_{j=1}^{dy} \frac{\partial L}{\partial \theta_{aj}} \cdot W_{bj} = \sum_{j=1}^{dy} (\hat{y} - y)_{aj} \cdot W_{jb}^T = \left[(\hat{y} - y) \cdot W_2^T\right]_{a,b} \Rightarrow \frac{\partial L}{\partial h} = \hat{y} - y \cdot W_2^T$$

$$\frac{\partial h}{\partial f} = sg(h), \quad \frac{\partial L}{\partial f} = \hat{y} - y \cdot W_2^T \overset{*}{\underset{\text{element wise}}{\circ}} sg(h)$$

$$\frac{\partial L}{\partial b_t^1} = \sum_{i=1}^{h}\sum_{j=1}^{dy} \frac{\partial L}{\partial f_{ij}} \cdot \frac{\partial f_{ij}}{\partial b_t^1} = \sum_{i=1}^{h} \frac{\partial L}{\partial f_{it}} \cdot \frac{\partial f_{it}}{\partial b_t^1} = \sum_{i=1}^{h} (\hat{y} - y \cdot W_2^T sg(h))_{it} \Rightarrow \frac{\partial L}{\partial b^1} = \sum_{i=1}^{h} (\hat{y} - y \cdot W_2^T sg(h))_i = hp.sum(\hat{y} - y \cdot W_2^T \overset{*}{\underset{\text{element wise}}{\circ}} sg(h), Axis = 0)$$

$$\frac{\partial f}{\partial W_1} = X^T \circ \hat{y} - y \cdot W_2^T \overset{*}{\underset{\text{element wise}}{\circ}} sg(h)$$


mxdx
dxxh
mxh
mxh
x
W1
+b1
f
mxh
hxdy
mxdy
mxdy
h
W2
+b2
SM
ŷ
mxdy
CE
mx1


data
linear
H
linear
SM