

$X_{train} = \text{Matrix}(\text{Batch_size}, 28, 28)$
 $Y_{train} = \text{Matrix}(1 \times \text{batch_size})$

h hidden_Layer_size (100)

W_1 Matrix (28*28, h)
 b_1 Matrix (1, h)

W_2 Matrix (h, k) } $k=10$ [Anzahl Klassen]
 b_2 Matrix (1, k)

step-size float
 reg float

Learning
 ↙

for 1 to epochs (100):

forward pass
 $d = \text{mul}(X_{train}, W_1)$ $d_{ik} = \sum_{j=1}^m x_{train,ij} \cdot W_{1,jk}$ Matrix multiplication
 $h_l = d + b_1$ $h_{l,k} = d_{ik} + b_{1,k}$ Add Matrix & Vector
 $h_{lrelu} = \max(0, h_l)$ element wise $\max(0, x)$

$d_2 = \text{mul}(h_{lrelu}, W_2)$

$\text{score} = d_2 + b_2$

gradient backpropagation
 $\text{score} \Rightarrow$ alle Werte die ≤ 0 auf 0 setzen
 $\text{scores} \Rightarrow$ den Wert der richtigen Klasse um 1 verringern
 $\text{scores} \Rightarrow$ jeden Wert durch batch_size dividieren

$dW_2 = \text{mul}(h_{lrelu}^T, \text{score})$

$db_2 = \text{summe scores axis=0, wenn batch_size} \neq 1$ $db_2 = \text{scores}$

$d_{hidden} = \text{mul}(\text{scores}, W_2^T)$
 $d_{hidden} \Rightarrow$ alle Werte ≤ 0 auf 0 setzen

$dW_1 = \text{mul}(X_{train}^T, d_{hidden})$

$db_1 = \text{summe } d_{hidden} \text{ axis=0, wenn batch_size} \neq 1$ $db_1 = d_{hidden}$

$dW_2 += \text{reg} * W_2$
 $dW_1 += \text{reg} * W_1$

$W_1 += -\text{step_size} * dW_1$
 $b_1 += -\text{step_size} * db_1$
 $W_2 += -\text{step_size} * dW_2$
 $b_2 += -\text{step_size} * db_2$