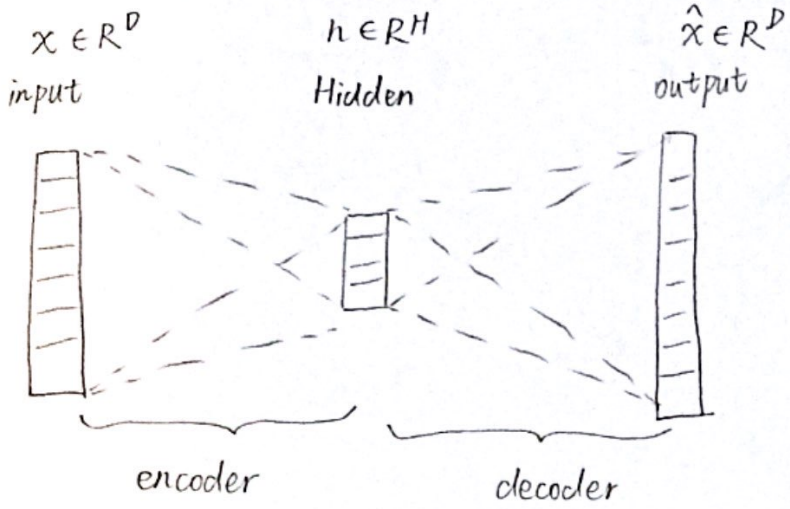

input $x \in R^D$

encoder network $h = ReLU(Wx + b)$ $W \in R^{H \times D}$ $b \in R^H$

decoder network $\hat{x} = ReLU(\tilde{W}h + \tilde{b})$ $\tilde{W} \in R^{D \times H}$ $\tilde{b} \in R^D$

output $\hat{x} \in R^D$

$ReLU = max(0, x)$ applied element-wise

Training objective:

$$L = \frac{1}{n} \sum_{i=1}^{n} \| x_i - ReLU(\tilde{W}\, ReLU(Wx_i + b) + \tilde{b} \|_2^2$$

Using back-propagation to find optimal solution of weights $W, \tilde{W}$ and bias $b, \tilde{b}$

① Randomly initiate $W, b, \tilde{W}, \tilde{b}$

② Compute gradient of Loss function with respect to $W, b, \tilde{W}, \tilde{b}$

③ Use stochastic gradient descent to update $W, b, \tilde{W}, \tilde{b}$ until convergence.

Quiz4: Algorithm of an one layer Autoencoder