```
Deep MLP training: stochastic gradient descent.
```

DEEP-MLP-TRAINING (**D**, h, η , maxiter, n_1, n_2, \dots, n_h , $f^1, f^2, \dots, \overline{f^{h+1}}$): 1 $n_0 \leftarrow d$ // input layer size

2 $n_{h+1} \leftarrow p //$ output layer size // Initialize weight matrices and bias vectors

for
$$l=0,1,2,\cdots,h$$
 do

3 for $l = 0, 1, 2, \dots, h$ do

$$\theta_l \leftarrow \text{random } n_{l+1} \text{ vector with small values}$$

 $\mathbf{W}_l \leftarrow \text{random } n_l \times n_{l+1} \text{ matrix with small values}$

5
$$\mathbf{w}_l \leftarrow \text{random } n_l \times n_{l+1} \text{ matrix with small val}$$
6 $t \leftarrow 0 \text{ // iteration counter}$

7 repeat

foreach $(\mathbf{x}_i, \mathbf{y}_i) \in \mathbf{D}$ in random order **do**

| Ioreach
$$(\mathbf{x}_i, \mathbf{y}_i) \in \mathbf{D}$$
 in random order do | // Feed-Forward Phase | $\mathbf{z}^0 \leftarrow \mathbf{x}$.

$$\mathbf{z}^0 \leftarrow \mathbf{x}_i$$

$$\mathbf{for} \ l = 0, 1, 2, \dots, h \ \mathbf{do}$$

9

10

11

12

13

14

15

16

19

20

21

22

23

 $t \leftarrow t + 1$

24 until t > maxiter

$$\mathbf{z}^{l+1} \leftarrow f^{l+1} \left(\mathbf{W}_l^T \cdot \mathbf{z}^l \right)$$

$$\mathbf{o}_i \leftarrow \mathbf{z}^{h+1}$$

$$\mathbf{o}_i \leftarrow \mathbf{z}^{h+1}$$
 // Backpropagation Pha

// Backpropagation Phase
$$oldsymbol{\delta}^{h+1} \leftarrow \partial \mathbf{f}^{h+1} \odot \partial \boldsymbol{\mathcal{E}}_{\mathbf{x}}$$
 // net gradients at output

$$\delta^{n+1} \leftarrow \partial \mathbf{f}^{n+1} \odot \partial \mathcal{E}_{\mathbf{x}} / / \text{ net gradients at}$$

$$// \text{ use } \partial \mathbf{F}^{n+1} \partial \mathcal{E}_{\mathbf{x}} \text{ for softmax}$$

$$\mathbf{for } l = h, h - 1, \dots, 1 \mathbf{do}$$

$$\begin{array}{c|c} \mathbf{for} \ l = h, h-1, \cdots, 1 \ \mathbf{do} \\ & \bigsqcup \ \delta^l \leftarrow \partial \mathbf{f}^l \odot \left(\mathbf{W}_l \cdot \boldsymbol{\delta}^{l+1} \right) / / \ \text{net gradients at layer} \ l \end{array}$$

for
$$l = 0, 1, \dots, h$$
 do
$$\begin{vmatrix} \mathbf{\nabla}_{\mathbf{W}_l} \leftarrow \mathbf{z}^l \cdot (\mathbf{\delta}^{l+1})^T \end{vmatrix}$$

for
$$l = 0, 1, \dots, h$$
 do
$$\nabla_{\mathbf{W}_{l}} \leftarrow \mathbf{z}^{l} \cdot (\boldsymbol{\delta}^{l+1})^{T}$$

$$\begin{array}{c|c}
\textbf{for } l = 0, 1, \cdots, h \textbf{ do} \\
 & \nabla_{\mathbf{W}_l} \leftarrow \mathbf{z}^l \cdot \left(\boldsymbol{\delta}^{l+1}\right)^T / / \\
 & \nabla_{\boldsymbol{\alpha}} \leftarrow \boldsymbol{\delta}^{l+1} / / \text{ bias}
\end{array}$$

for $l = 0, 1, \dots, h$ **do**

for
$$l = 0, 1, \dots, h$$
 do
$$\nabla_{\mathbf{W}_{l}} \leftarrow \mathbf{z}^{l} \cdot (\delta^{l+1})^{T} //$$

or
$$l = 0, 1, \cdots, h$$
 do
$$\nabla_{\mathbf{W}_l} \leftarrow \mathbf{z}^l \cdot \left(\boldsymbol{\delta}^{l+1} \right)^T / / \text{ weight gradient matrix at layer } l$$

$$\nabla_{\mathbf{G}} \leftarrow \boldsymbol{\delta}^{l+1} / / \text{ bias gradient vector at layer } l$$

$$abla_{\mathbf{W}_l} \leftarrow \mathbf{z}^l \cdot \left(\delta^{l+1}\right)^T / / \text{ weight gradient matrix at }
abla_{ heta_l} \leftarrow \delta^{l+1} / / \text{ bias gradient vector at layer } l$$

 $oldsymbol{ heta}_l \leftarrow oldsymbol{ heta}_l - \eta \cdot oldsymbol{
abla}_{ heta_l}$ // update $oldsymbol{ heta}_l$

 $\mathbf{W}_l \leftarrow \mathbf{W}_l - \eta \cdot \mathbf{\nabla}_{\mathbf{W}_l}$ // update \mathbf{W}_l

$$0.01$$
 , 1 \mathbf{do} , 1 \mathbf

crix at layer
$$l$$

natrix at layer
$$l$$

$${\tt layer}\ l$$