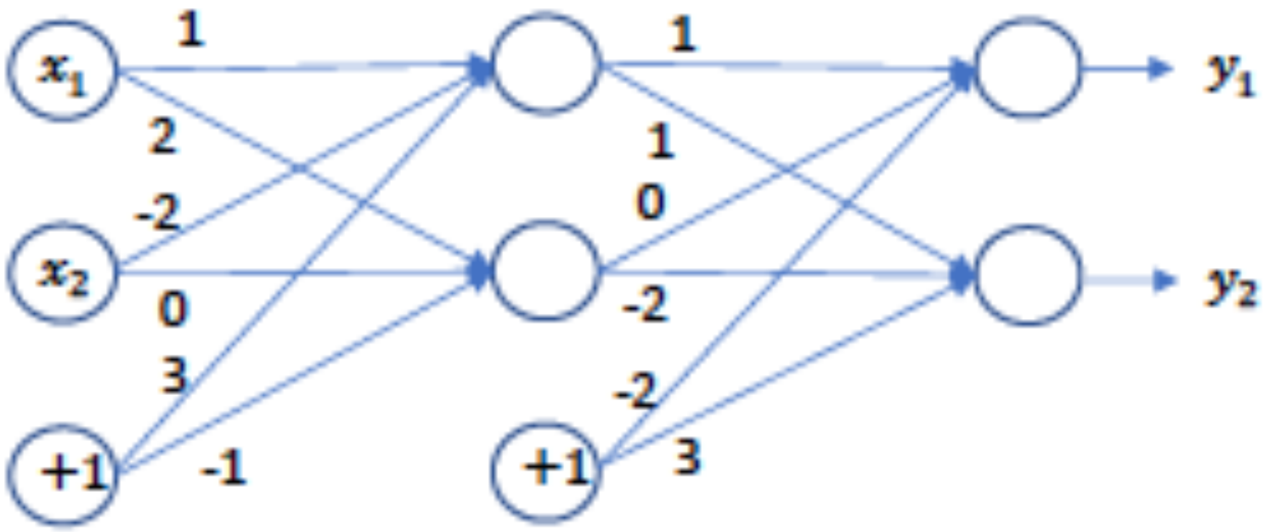
Deep feedforward neural networks

CE/CZ4042 – Tutorial 5



x_1
x_2
+1
+1
1
2
-2
0
3
-1
1
1
0
-2
-2
3
y_1
y_2

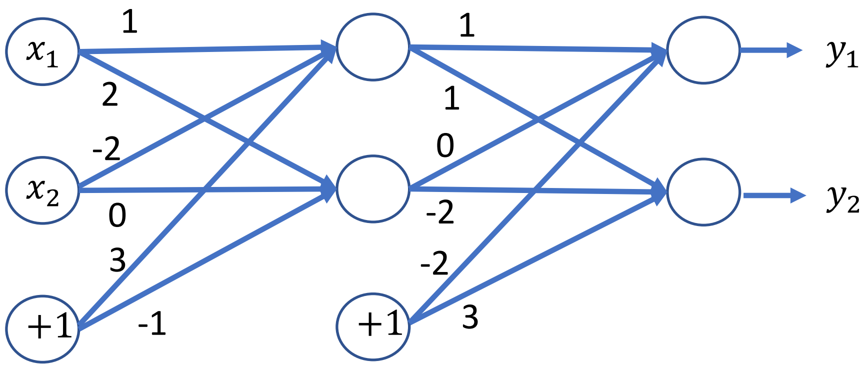
1. The three-layer feedforwad perceptron network shown in figure 1 has weights and biases initialized as indicated and receives 2-dimensional inputs (x_1, x_2). The network is to respond with $\boldsymbol{d}_1 = \binom{0}{1}$ and $\boldsymbol{d}_2 = \binom{1}{0}$ for input patterns $\boldsymbol{x}_1 = \binom{1.0}{3.0}$ and $\boldsymbol{x}_2 = \binom{-2.0}{-2.0}$, respectively.

Analyse a single feedforward and feedback step for gradient decent learning of the two patterns by doing the following:

(a) Find the weight matrix $\boldsymbol{W}$ to the hidden-layer and weight matrix $\boldsymbol{V}$ to the output-layer, and the corresponding biases.
(b) Calculate the synaptic input $\boldsymbol{z}$ and output $\boldsymbol{h}$ of the hidden-layer, and the synaptic input $\boldsymbol{u}$ and output $\boldsymbol{y} = (y_1, y_2)$ of the output layer.
(c) Find the mean square error cost J between the outputs and targets.
(d) Calculate the gradients $\nabla_{\boldsymbol{u}} J$ and $\nabla_{\boldsymbol{z}} J$ at the output-layer and hidden-layer, respectively.
(e) Compute the new weights and biases.
(f) Write a program to continue iterations until convergence and find the final weights and biases.

Assume a learning rate of 0.05.

Repeat above (a) – (f) for stochastic gradient decent learning.



Weight matrix to the hidden layer, $\boldsymbol{W} = \begin{pmatrix} 1.0 & 2.0 \\ -2.0 & 0.0 \end{pmatrix}$

Bias vector to the hidden-layer $\boldsymbol{b} = \begin{pmatrix} 3.0 \\ -1.0 \end{pmatrix}$

Weight matrix to the output-layer, $\boldsymbol{V} = \begin{pmatrix} 1.0 & 1.0 \\ 0.0 & -2.0 \end{pmatrix}$

Bias vector to the output-layer $\mathbf{c} = \begin{pmatrix} -2.0 \\ 3.0 \end{pmatrix}$

Gradient descent learning for 3-layer perceptron network:

Given a training dataset $(\boldsymbol{X}, \boldsymbol{D})$

Set learning parameter α

Initialize $\boldsymbol{W}$, $\boldsymbol{b}$, $\boldsymbol{V}$, $\boldsymbol{c}$

Repeat until convergence:

$$\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{W} + \boldsymbol{B}$$
$$\boldsymbol{H} = g(\boldsymbol{Z})$$
$$\boldsymbol{U} = \boldsymbol{H}\boldsymbol{V} + \boldsymbol{C}$$
$$\boldsymbol{Y} = f(\boldsymbol{U})$$

$$\nabla_{\boldsymbol{U}} J = -(\boldsymbol{D} - \boldsymbol{Y}) \cdot f'(\boldsymbol{U})$$
$$\nabla_{\boldsymbol{Z}} J = (\nabla_{\boldsymbol{U}} J)\boldsymbol{V}^T \cdot f'(\boldsymbol{Z})$$

$$\boldsymbol{V} \leftarrow \boldsymbol{V} - \alpha \boldsymbol{H}^T\, \nabla_{\boldsymbol{U}} J$$
$$\boldsymbol{c} \leftarrow \boldsymbol{c} - \alpha (\nabla_{\boldsymbol{U}} J)^T \boldsymbol{1}_P$$
$$\boldsymbol{W} \leftarrow \boldsymbol{W} - \alpha \boldsymbol{X}^T\, \nabla_{\boldsymbol{Z}} J$$
$$\boldsymbol{b} \leftarrow \boldsymbol{b} - \alpha (\nabla_{\boldsymbol{Z}} J)^T \boldsymbol{1}_P$$

$$\boldsymbol{x}_1 = \begin{pmatrix} 1.0 \\ 3.0 \end{pmatrix} \text{ and } \boldsymbol{d}_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$
$$\boldsymbol{x}_2 = \begin{pmatrix} -2.0 \\ -2.0 \end{pmatrix} \text{ and } \boldsymbol{d}_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\boldsymbol{X} = \begin{pmatrix} 1.0 & 3.0 \\ -2.0 & -2.0 \end{pmatrix} \text{ and } \boldsymbol{D} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Forward propagation:

Synaptic input to hidden-layer, $\boldsymbol{Z} = \boldsymbol{XW} + \boldsymbol{B}$

$$= \begin{pmatrix} 1.0 & 3.0 \\ -2.0 & -2.0 \end{pmatrix} \begin{pmatrix} 1.0 & 2.0 \\ -2.0 & 0.0 \end{pmatrix} + \begin{pmatrix} 3.0 & -1.0 \\ 3.0 & -1.0 \end{pmatrix}$$
$$= \begin{pmatrix} -2.0 & 1.0 \\ 5.0 & -5.0 \end{pmatrix}$$

Output of the hidden layer, $\boldsymbol{H} = f(\boldsymbol{Z}) = \frac{\mathbf{1}}{\mathbf{1}+\boldsymbol{e}^{-\boldsymbol{Z}}} = \begin{pmatrix} 0.12 & 0.73 \\ 0.99 & 0.01 \end{pmatrix}$

Synaptic input to output-layer, $\boldsymbol{U} = \boldsymbol{HV} + \boldsymbol{C}$

$$= \begin{pmatrix} 0.12 & 0.73 \\ 0.99 & 0.01 \end{pmatrix} \begin{pmatrix} 1.0 & 1.0 \\ 0.0 & -2.0 \end{pmatrix} + \begin{pmatrix} -2.0 & 3.0 \\ -2.0 & 3.0 \end{pmatrix}$$

$$= \begin{pmatrix} -1.88 & 1.66 \\ -0.99 & 3.98 \end{pmatrix}$$

Output of the output layer, $\boldsymbol{Y} = f(\boldsymbol{U}) = \frac{\mathbf{1}}{\mathbf{1}+\boldsymbol{e}^{-\boldsymbol{U}}} = \begin{pmatrix} 0.13 & 0.84 \\ 0.27 & 0.98 \end{pmatrix}$

$$m.s.e. = \frac{1}{2} \sum_{p=1}^{2} \sum_{k=1}^{2} \left(d_{pk} - y_{pk}\right)^2 = 0.769$$

Computing gradients:

$$f'(\boldsymbol{U}) = \boldsymbol{Y} \cdot (\boldsymbol{1} - \boldsymbol{Y}) = \begin{pmatrix} 0.13 & 0.84 \\ 0.27 & 0.98 \end{pmatrix} \cdot \left(\begin{pmatrix} 1.0 & 1.0 \\ 1.0 & 1.0 \end{pmatrix} - \begin{pmatrix} 0.13 & 0.84 \\ 0.27 & 0.98 \end{pmatrix} \right) = \begin{pmatrix} 0.11 & 0.13 \\ 0.20 & 0.02 \end{pmatrix}$$

$$\nabla_{\boldsymbol{U}} J = -(\boldsymbol{D} - \boldsymbol{Y}) \cdot f'(\boldsymbol{U}) = -\left(\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} - \begin{pmatrix} 0.13 & 0.84 \\ 0.27 & 0.98 \end{pmatrix} \right) \begin{pmatrix} 0.12 & 0.13 \\ 0.20 & 0.02 \end{pmatrix} = \begin{pmatrix} 0.02 & -0.02 \\ -0.14 & 0.02 \end{pmatrix}$$

$$f'(\boldsymbol{Z}) = \boldsymbol{H} \cdot (1 - \boldsymbol{H}) = \begin{pmatrix} 0.12 & 0.73 \\ 0.99 & 0.01 \end{pmatrix} \cdot \left(\begin{pmatrix} 1.0 & 1.0 \\ 1.0 & 1.0 \end{pmatrix} - \begin{pmatrix} 0.12 & 0.73 \\ 0.99 & 0.01 \end{pmatrix} \right) = \begin{pmatrix} 0.10 & 0.2 \\ 0.01 & 0.01 \end{pmatrix}$$

$$\nabla_{\boldsymbol{Z}} J = (\nabla_{\boldsymbol{U}} J) \boldsymbol{V}^T \cdot f'(\boldsymbol{Z}) = \begin{pmatrix} 0.02 & -0.02 \\ -0.14 & 0.02 \end{pmatrix} \begin{pmatrix} 1.0 & 0.0 \\ 1.0 & -2.0 \end{pmatrix} \cdot \begin{pmatrix} 0.10 & 0.2 \\ 0.01 & 0.01 \end{pmatrix} = \begin{pmatrix} -0.001 & -0.01 \\ -0.001 & 0.00 \end{pmatrix}$$

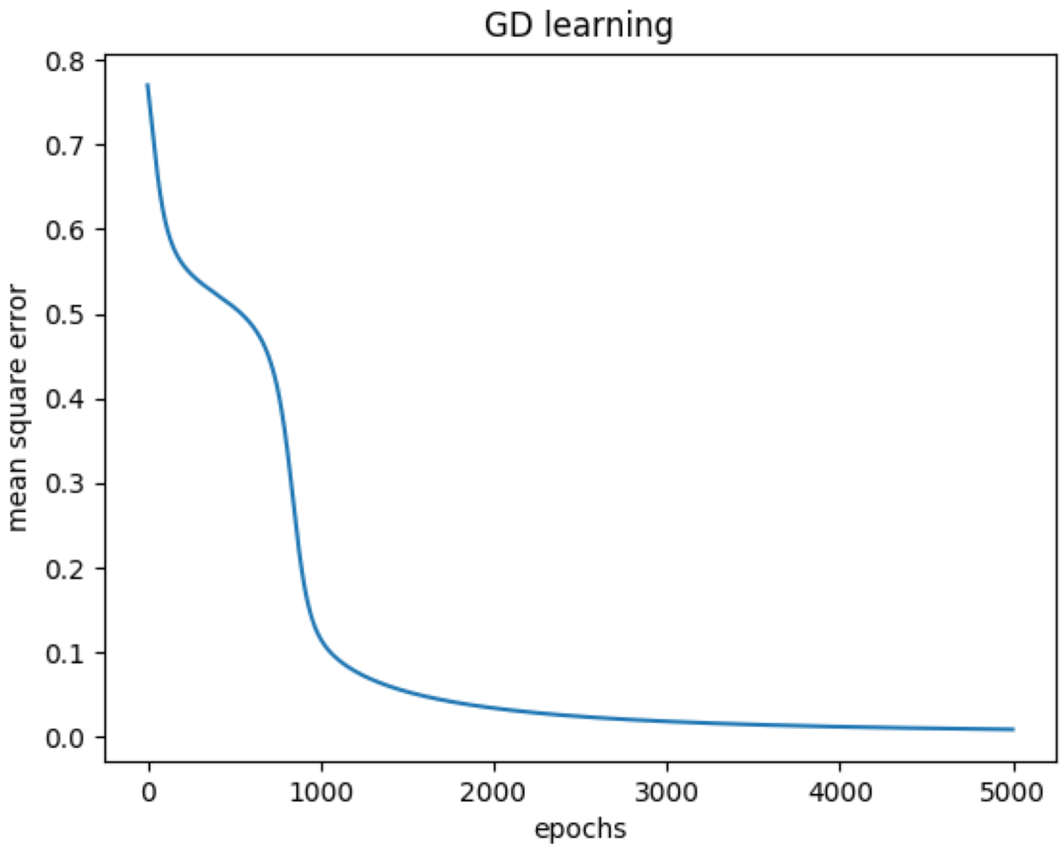
Updating weights:

$$\boldsymbol{V} \leftarrow \boldsymbol{V} - \alpha \boldsymbol{H}^T \, \nabla_{\boldsymbol{U}} J = \begin{pmatrix} 1.01 & 1.0 \\ 0.0 & -2.0 \end{pmatrix}$$

$$\boldsymbol{c} \leftarrow \boldsymbol{c} - \alpha (\nabla_{\boldsymbol{U}} J)^T \mathbf{1}_P = \begin{pmatrix} -1.99 \\ 3.00 \end{pmatrix}$$

$$\boldsymbol{W} \leftarrow \boldsymbol{W} - \alpha \boldsymbol{X}^T \, \nabla_{\boldsymbol{Z}} J = \begin{pmatrix} 1.0 & 2.0 \\ -2.0 & 0.0 \end{pmatrix}$$

$$\boldsymbol{b} \leftarrow \boldsymbol{b} - \alpha (\nabla_{\boldsymbol{Z}} J)^T \mathbf{1}_P = \begin{pmatrix} 3.0 \\ -1.0 \end{pmatrix}$$



GD learning
mean square error
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
1000
2000
3000
4000
5000
epochs

At convergence:

$$\boldsymbol{W} = \begin{pmatrix} 0.63 & 0.60 \\ -3.0 & -2.0 \end{pmatrix}, \boldsymbol{b} = \begin{pmatrix} 2.72 \\ -0.74 \end{pmatrix}$$

$$\boldsymbol{V} = \begin{pmatrix} 4.97 & -3.46 \\ 0.25 & -2.37 \end{pmatrix}, \mathbf{c} = \begin{pmatrix} -2.42 \\ 2.56 \end{pmatrix}$$

Predicted values:

$\boldsymbol{y}_1 = \begin{pmatrix} 0.08 \\ 0.93 \end{pmatrix}$ and $\boldsymbol{y}_2 = \begin{pmatrix} 0.94 \\ 0.05 \end{pmatrix}$

m.s.e. = 0.009

Stochastic gradient descent learning for 3-layer perceptron network:

Given a training dataset $\{(\boldsymbol{x}, \boldsymbol{d})\}$
Set learning parameter α
Initialize $\boldsymbol{W}$, $\boldsymbol{b}$, $\boldsymbol{V}$, $\boldsymbol{c}$
Repeat until convergence:

For every pattern $(\boldsymbol{x}, \boldsymbol{d})$:

$$\boldsymbol{z} = \boldsymbol{W}^T \boldsymbol{x} + \boldsymbol{b}$$
$$\boldsymbol{h} = f(\boldsymbol{z})$$
$$\boldsymbol{u} = \boldsymbol{V}^T \boldsymbol{h} + \boldsymbol{c}$$
$$\boldsymbol{y} = f(\boldsymbol{u})$$

$$\nabla_{\boldsymbol{u}} J = -(\boldsymbol{d} - \boldsymbol{y}) \cdot f'(\boldsymbol{z})$$
$$\nabla_{\boldsymbol{z}} J = \boldsymbol{V} \nabla_{\boldsymbol{u}} J \cdot f'(\boldsymbol{z})$$

$$\boldsymbol{V} \leftarrow \boldsymbol{V} - \alpha \boldsymbol{h} (\nabla_{\boldsymbol{u}} J)^T$$
$$\boldsymbol{c} \leftarrow \boldsymbol{c} - \alpha \nabla_{\boldsymbol{u}} J$$
$$\boldsymbol{W} \leftarrow \boldsymbol{W} - \alpha \boldsymbol{x} (\nabla_{\boldsymbol{z}} J)^T$$
$$\boldsymbol{b} \leftarrow \boldsymbol{b} - \alpha \nabla_{\boldsymbol{z}} J$$

Iteration 1:

Apply **first** pattern $\boldsymbol{x} = \begin{pmatrix} 1.0 \\ 3.0 \end{pmatrix}$ and $\boldsymbol{d} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$:

Synaptic input to the hidden-layer

$$\boldsymbol{z} = \boldsymbol{W}^T\boldsymbol{x} + \boldsymbol{b} = \begin{pmatrix} 1.0 & -2.0 \\ 2.0 & 0.0 \end{pmatrix}\begin{pmatrix} 1.0 \\ 3.0 \end{pmatrix} + \begin{pmatrix} 3.0 \\ -1.0 \end{pmatrix} = \begin{pmatrix} -2.0 \\ 1.0 \end{pmatrix}$$

Output of the hidden-layer $\boldsymbol{h} = f(\boldsymbol{z}) = \frac{1}{1+e^{-\boldsymbol{z}}} = \begin{pmatrix} 0.12 \\ 0.73 \end{pmatrix}$

Synaptic input to output-layer

$$\boldsymbol{u} = \boldsymbol{V}^T\boldsymbol{h} + \boldsymbol{c} = \begin{pmatrix} -1.88 \\ 1.66 \end{pmatrix}$$

Output of the output-layer $\boldsymbol{y} = f(\boldsymbol{u}) = \frac{1}{1+e^{-\boldsymbol{u}}} = \begin{pmatrix} 0.13 \\ 0.84 \end{pmatrix}$

$s.e. = (d_1 - y_1)^2 + (d_2 - y_2)^2 = 0.043$

Computing gradients:

$$f'(\boldsymbol{u}) = \boldsymbol{y} \cdot (1 - \boldsymbol{y}) = \begin{pmatrix} 0.13 \\ 0.84 \end{pmatrix} \cdot \left(\begin{pmatrix} 1.0 \\ 1.0 \end{pmatrix} - \begin{pmatrix} 0.13 \\ 0.84 \end{pmatrix} \right) = \begin{pmatrix} 0.11 \\ 0.13 \end{pmatrix}$$

$$\nabla_{\boldsymbol{u}} J = -(\boldsymbol{d} - \boldsymbol{y}) \cdot f'(\boldsymbol{u}) = -\left(\begin{pmatrix} 0 \\ 1 \end{pmatrix} - \begin{pmatrix} 0.13 \\ 0.84 \end{pmatrix} \right) \cdot \begin{pmatrix} 0.12 \\ 0.14 \end{pmatrix} = \begin{pmatrix} 0.02 \\ -0.02 \end{pmatrix}$$

$$f'(\boldsymbol{z}) = \boldsymbol{h} \cdot (1 - \boldsymbol{h}) = \begin{pmatrix} 0.12 \\ 0.73 \end{pmatrix} \cdot \left(\begin{pmatrix} 1.0 \\ 1.0 \end{pmatrix} - \begin{pmatrix} 0.12 \\ 0.73 \end{pmatrix} \right) = \begin{pmatrix} 0.10 \\ 0.20 \end{pmatrix}$$

$$\nabla_{\boldsymbol{z}} J = \boldsymbol{V} \nabla_{\boldsymbol{u}} J \cdot f'(\boldsymbol{z}) = \begin{pmatrix} 1.0 & 1.0 \\ 0.0 & -2.0 \end{pmatrix} \begin{pmatrix} 0.02 \\ -0.02 \end{pmatrix} \cdot \begin{pmatrix} 0.11 \\ 0.20 \end{pmatrix} = \begin{pmatrix} -0.001 \\ 0.008 \end{pmatrix}$$

Updating weights:

$$\boldsymbol{V} \leftarrow \boldsymbol{V} - \alpha \boldsymbol{h} (\nabla_{\boldsymbol{u}} J)^T = \begin{pmatrix} 1.0 & 1.0 \\ 0.0 & -2.0 \end{pmatrix} - 0.2 \begin{pmatrix} 0.12 \\ 0.73 \end{pmatrix} \begin{pmatrix} -0.02 & 0.022 \end{pmatrix} = \begin{pmatrix} 1.0 & 1.0001 \\ 0.00 & -2.0 \end{pmatrix}$$

$$\boldsymbol{c} \leftarrow \boldsymbol{c} - \alpha \nabla_{\boldsymbol{u}} J = \begin{pmatrix} -2.0 \\ 3.0 \end{pmatrix} + 0.2 \begin{pmatrix} 0.02 \\ -0.02 \end{pmatrix} = \begin{pmatrix} -2.00 \\ 3.001 \end{pmatrix}$$

$$\boldsymbol{W} \leftarrow \boldsymbol{W} - \alpha \boldsymbol{x} (\nabla_{\boldsymbol{z}} J)^T = \begin{pmatrix} 1.0 & 2.0 \\ -2.00 & -0.001 \end{pmatrix}$$

$$\boldsymbol{b} \leftarrow \boldsymbol{b} - \alpha \nabla_{\boldsymbol{z}} J = \begin{pmatrix} 3.00 \\ -1.00 \end{pmatrix}$$

Apply **second** pattern $\boldsymbol{x} = \begin{pmatrix} -2.0 \\ -2.0 \end{pmatrix}$ and $\boldsymbol{d} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$:

Synaptic input to the hidden-layer

$$\boldsymbol{z} = \boldsymbol{W}^T \boldsymbol{x} + \boldsymbol{b} = \begin{pmatrix} 5.0 \\ -5.0 \end{pmatrix}$$

Output of the hidden-layer $\boldsymbol{h} = f(\boldsymbol{z}) = \frac{1}{1+e^{-\boldsymbol{z}}} = \begin{pmatrix} 1.0 \\ 0.007 \end{pmatrix}$

Synaptic input to output-layer

$$\boldsymbol{u} = \boldsymbol{V}^T \boldsymbol{h} + \boldsymbol{c} = \begin{pmatrix} -0.99 \\ 3.98 \end{pmatrix}$$

Output of the output-layer $\boldsymbol{y} = f(\boldsymbol{u}) = \frac{1}{1+e^{-\boldsymbol{u}}} = \begin{pmatrix} 0.27 \\ 0.98 \end{pmatrix}$

$s.e. = (d_1 - y_1)^2 + (d_2 - y_2)^2 = 1.5$

Computing gradients:

$$f'(\boldsymbol{u}) = \boldsymbol{y} \cdot (1 - \boldsymbol{y}) = \begin{pmatrix} 0.195 \\ 0.018 \end{pmatrix}$$

$$\nabla_{\boldsymbol{u}} J = -(\boldsymbol{d} - \boldsymbol{y}) \cdot f'(\boldsymbol{u}) = \begin{pmatrix} -0.14 \\ 0.018 \end{pmatrix}$$

$$f'(\boldsymbol{z}) = \boldsymbol{h} \cdot (1 - \boldsymbol{h}) = \begin{pmatrix} 0.007 \\ 0.007 \end{pmatrix}$$

$$\nabla_{\boldsymbol{z}} J = \boldsymbol{V} \nabla_{\boldsymbol{u}} J \cdot f'(\boldsymbol{z}) = \begin{pmatrix} -0.0008 \\ -0.0002 \end{pmatrix}$$

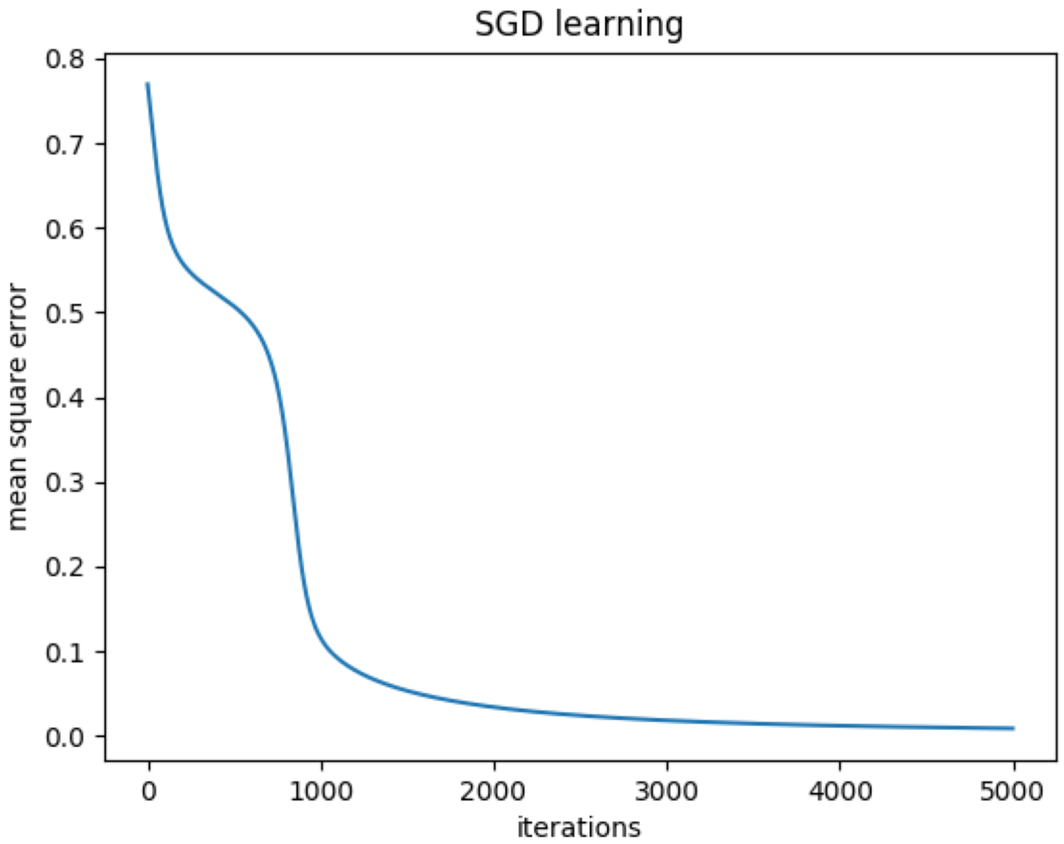
Updating weights:

$$\boldsymbol{V} \leftarrow \boldsymbol{V} - \alpha \boldsymbol{h} (\nabla_{\boldsymbol{u}} J)^T = \begin{pmatrix} 1.007 & 0.99 \\ 0.0 & -2.0 \end{pmatrix}$$

$$\boldsymbol{c} \leftarrow \boldsymbol{c} - \alpha \nabla_{\boldsymbol{u}} J = \begin{pmatrix} -1.99 \\ 3.0 \end{pmatrix}$$

$$\boldsymbol{W} \leftarrow \boldsymbol{W} - \alpha \boldsymbol{x} (\nabla_{\boldsymbol{z}} J)^T = \begin{pmatrix} 0.999 & 1.99 \\ -1.99 & 0.00 \end{pmatrix}$$

$$\boldsymbol{b} \leftarrow \boldsymbol{b} - \alpha \nabla_{\boldsymbol{z}} J = \begin{pmatrix} 3.00 \\ -1.00 \end{pmatrix}$$


SGD learning
mean square error
iterations
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
1000
2000
3000
4000
5000

2. A feedforward neural network with one hidden layer to perform the following classification:

class	inputs
A	$(1.0, 1.0), (0.0, 1.0)$
B	$(3.0, 4.0), (2.0, 2.0)$
C	$(2.0, -2.0), (-2.0, -3.0)$

The network has a hidden layer of three perceptrons and a softmax output layer.

Show one iteration of gradient descent learning and plot learning curves until convergence at a learning rate $\alpha = 0.1$.

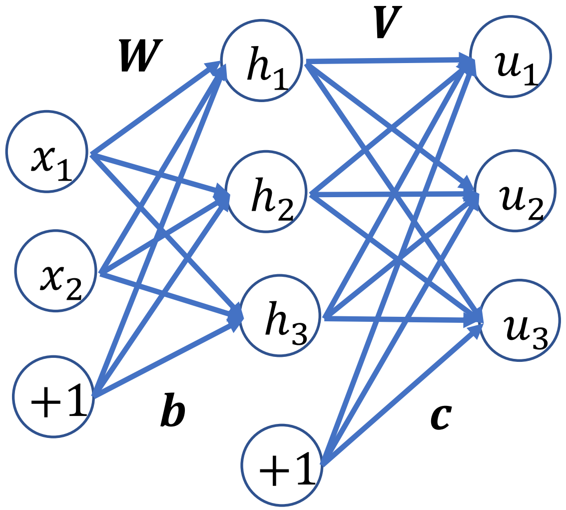
Determine the weights and biases at convergence.

class	*inputs*	*Label*
A	$(1.0, 1.0), (0.0, 1.0)$	0
B	$(3.0, 4.0), (2.0, 2.0)$	1
C	$(2.0, -2.0), (-2.0, -3.0)$	2

Feedforward network :

Perceptron hidden layer with 3 neurons

Softmax output layer with 3 neurons



GD for the feedforward network

Given a training dataset $(\boldsymbol{X}, \boldsymbol{D})$

Set learning parameter α

Initialize $\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{V}, \boldsymbol{c}$

Repeat until convergence:

$$\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{W} + \boldsymbol{B}$$
$$\boldsymbol{H} = f(\boldsymbol{Z})$$
$$\boldsymbol{U} = \boldsymbol{H}\boldsymbol{V} + \boldsymbol{C}$$
$$\boldsymbol{Y} = \arg\max_k g(\boldsymbol{U})$$

$$\nabla_{\boldsymbol{U}} J = -\big(\boldsymbol{K} - g(\boldsymbol{U})\big)$$
$$\nabla_{\boldsymbol{Z}} J = (\nabla_{\boldsymbol{U}} J)\boldsymbol{V}^T \cdot f'(\boldsymbol{Z})$$

$$\boldsymbol{V} \leftarrow \boldsymbol{V} - \alpha \boldsymbol{H}^T \, \nabla_{\boldsymbol{U}} J$$
$$\boldsymbol{c} \leftarrow \boldsymbol{c} - \alpha (\nabla_{\boldsymbol{U}} J)^T \boldsymbol{1}_P$$
$$\boldsymbol{W} \leftarrow \boldsymbol{W} - \alpha \boldsymbol{X}^T \, \nabla_{\boldsymbol{Z}} J$$
$$\boldsymbol{b} \leftarrow \boldsymbol{b} - \alpha (\nabla_{\boldsymbol{Z}} J)^T \boldsymbol{1}_P$$

Gradient Descent Learning

Iteration 1

$$\boldsymbol{X} = \begin{pmatrix} 1.0 & 1.0 \\ 0.0 & 1.0 \\ 3.0 & 4.0 \\ 2.0 & 2.0 \\ 2.0 & -2.0 \\ -2.0 & -3.0 \end{pmatrix} \text{ and } \boldsymbol{D} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 2 \\ 2 \end{pmatrix}$$

Targets as a one hot matrix:

$$\boldsymbol{K} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

Initialize weights and biases

To the hidden layer,

$$\boldsymbol{W} = \begin{pmatrix} -0.10 & 0.97 & 0.18 \\ -0.70 & 0.38 & 0.93 \end{pmatrix}, \boldsymbol{b} = \begin{pmatrix} 0.0 \\ 0.0 \\ 0.0 \end{pmatrix}$$

To the output-layer

$$\boldsymbol{V} = \begin{pmatrix} 1.01 & 0.09 & -0.39 \\ 0.79 & -0.45 & -0.22 \\ 0.28 & 0.96 & -0.07 \end{pmatrix}, \boldsymbol{c} = \begin{pmatrix} 0.0 \\ 0.0 \\ 0.0 \end{pmatrix}$$

Hidden layer is a continuous perceptron layer. The activation function:

$$f(\boldsymbol{Z}) = \frac{1}{1 + e^{-\boldsymbol{Z}}}$$

Output layer is a softmax layer. The activation function:

$$g(\boldsymbol{U}) = \frac{e^{\boldsymbol{U}}}{\sum_{k=1}^{K} e^{\boldsymbol{U_k}}}$$

Synaptic input to hidden-layer,

$$\boldsymbol{Z} = \boldsymbol{XW} + \boldsymbol{B} = \begin{pmatrix} 1.0 & 1.0 \\ 0.0 & 1.0 \\ 3.0 & 4.0 \\ 2.0 & 2.0 \\ 2.0 & -2.0 \\ -2.0 & -3.0 \end{pmatrix} \begin{pmatrix} -0.10 & 0.97 & 0.18 \\ -0.70 & 0.38 & 0.93 \end{pmatrix} + \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix}$$

$$= \begin{pmatrix} -0.80 & 1.35 & 1.10 \\ -0.70 & 0.38 & 0.93 \\ -3.08 & 4.44 & 4.23 \\ -1.59 & 2.70 & 2.21 \\ 1.20 & 1.18 & -1.50 \\ 2.29 & -3.08 & -3.13 \end{pmatrix}$$

Output of the hidden layer, $\boldsymbol{H} = f(\boldsymbol{Z}) = \frac{\mathbf{1}}{\mathbf{1}+\boldsymbol{e}^{-\boldsymbol{Z}}} = \begin{pmatrix} 0.31 & 0.79 & 0.75 \\ 0.33 & 0.59 & 0.72 \\ 0.04 & 0.99 & 0.99 \\ 0.17 & 0.94 & 0.90 \\ 0.77 & 0.77 & 0.18 \\ 0.91 & 0.04 & 0.04 \end{pmatrix}$

Synaptic input to output-layer,

$$\boldsymbol{U} = \boldsymbol{HV} + \boldsymbol{C} = \begin{pmatrix} 0.31 & 0.79 & 0.75 \\ 0.33 & 0.59 & 0.72 \\ 0.04 & 0.99 & 0.99 \\ 0.17 & 0.94 & 0.90 \\ 0.77 & 0.77 & 0.18 \\ 0.91 & 0.04 & 0.04 \end{pmatrix} \begin{pmatrix} 1.01 & 0.09 & -0.39 \\ 0.79 & -0.45 & -0.22 \\ 0.28 & 0.96 & -0.07 \end{pmatrix} + \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix} = \begin{pmatrix} 1.15 & 0.40 & -0.34 \\ 1.01 & 0.46 & -0.31 \\ 1.10 & 0.51 & -0.30 \\ 1.16 & 0.47 & -0.33 \\ 1.43 & -0.09 & -0.48 \\ 0.96 & 0.11 & -0.36 \end{pmatrix}$$

Output layer activation $g(\boldsymbol{U}) = \frac{e^{\boldsymbol{U}}}{\sum_{k=1}^{K} e^{\boldsymbol{U_k}}} = \begin{pmatrix} 0.59 & 0.28 & 0.13 \\ 0.54 & 0.31 & 0.15 \\ 0.56 & 0.31 & 0.14 \\ 0.58 & 0.29 & 0.13 \\ 0.73 & 0.16 & 0.11 \\ 0.59 & 0.25 & 0.16 \end{pmatrix}$

Output $\boldsymbol{Y} = \arg\max_k g(\boldsymbol{U}) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$

$$\boldsymbol{D} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 2 \\ 2 \end{pmatrix}, \qquad \boldsymbol{Y} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \qquad g(\boldsymbol{U}) = \begin{pmatrix} 0.59 & 0.28 & 0.13 \\ 0.54 & 0.31 & 0.15 \\ 0.56 & 0.31 & 0.14 \\ 0.58 & 0.29 & 0.13 \\ 0.73 & 0.16 & 0.11 \\ 0.59 & 0.25 & 0.16 \end{pmatrix}$$

Classification error = $\sum 1(\boldsymbol{D} \neq \boldsymbol{Y}) = 4$

Entropy $J = -\sum_{p=1}^{P} log\left(g\left(u_{pd_p}\right)\right)$

$$= -\big(log(0.59) + log(54) + log(0.31) + log(0.29) + log(0.11) + log(0.16)\big)$$
$$= 7.63$$

$$\nabla_{\boldsymbol{U}} J = -\left(\boldsymbol{K} - g(\boldsymbol{U})\right) = -\left(\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 0.59 & 0.28 & 0.13 \\ 0.54 & 0.31 & 0.15 \\ 0.56 & 0.31 & 0.14 \\ 0.58 & 0.29 & 0.13 \\ 0.73 & 0.16 & 0.11 \\ 0.59 & 0.25 & 0.16 \end{pmatrix}\right) = \begin{pmatrix} -0.41 & 0.28 & 0.13 \\ -0.46 & 0.31 & 0.15 \\ 0.56 & -0.69 & 0.14 \\ 0.58 & -0.71 & 0.13 \\ 0.73 & 0.16 & -0.89 \\ 0.59 & 0.25 & -0.84 \end{pmatrix}$$

$$f'(\boldsymbol{Z}) = \boldsymbol{H} \cdot (\boldsymbol{1} - \boldsymbol{H}) = \begin{pmatrix} 0.21 & 0.16 & 0.19 \\ 0.22 & 0.24 & 0.20 \\ 0.04 & 0.01 & 0.01 \\ 0.14 & 0.06 & 0.09 \\ 0.18 & 0.18 & 0.15 \\ 0.08 & 0.04 & 0.04 \end{pmatrix}$$

$$\nabla_{\boldsymbol{Z}} J = (\nabla_{\boldsymbol{U}} J)\boldsymbol{V}^T \cdot f'(\boldsymbol{Z}) = \begin{pmatrix} -0.41 & 0.28 & 0.13 \\ -0.46 & 0.31 & 0.15 \\ 0.56 & -0.69 & 0.14 \\ 0.58 & -0.71 & 0.13 \\ 0.73 & 0.16 & -0.89 \\ 0.59 & 0.25 & -0.84 \end{pmatrix} \begin{pmatrix} 1.01 & 0.09 & -0.39 \\ 0.79 & -0.45 & -0.22 \\ 0.28 & 0.96 & -0.07 \end{pmatrix}^T \cdot \begin{pmatrix} 0.21 & 0.16 & 0.19 \\ 0.22 & 0.24 & 0.20 \\ 0.04 & 0.01 & 0.01 \\ 0.14 & 0.06 & 0.09 \\ 0.18 & 0.18 & 0.15 \\ 0.08 & 0.04 & 0.04 \end{pmatrix} = \begin{pmatrix} -0.09 & -0.08 & 0.03 \\ -0.11 & -0.13 & 0.03 \\ 0.02 & 0.01 & -0.01 \\ 0.07 & 0.04 & -0.05 \\ 0.20 & 0.13 & 0.06 \\ 0.08 & 0.02 & 0.02 \end{pmatrix}$$

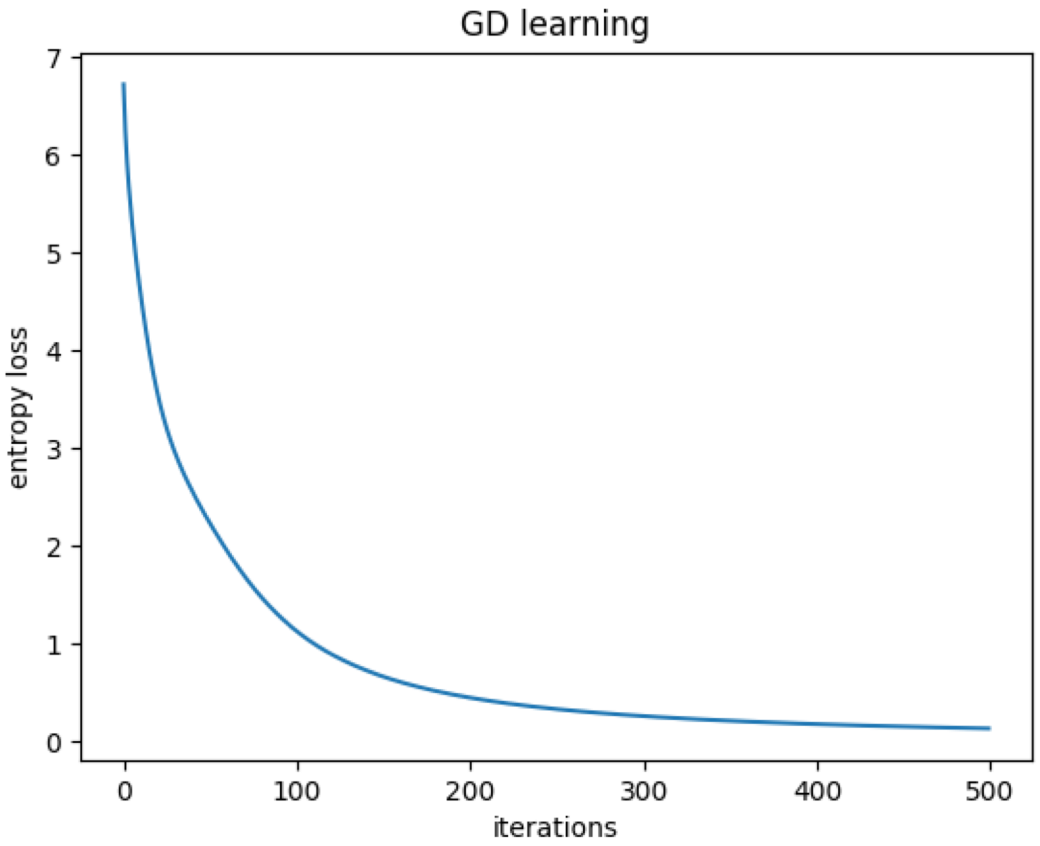
Learning rate $\alpha = 0.1$

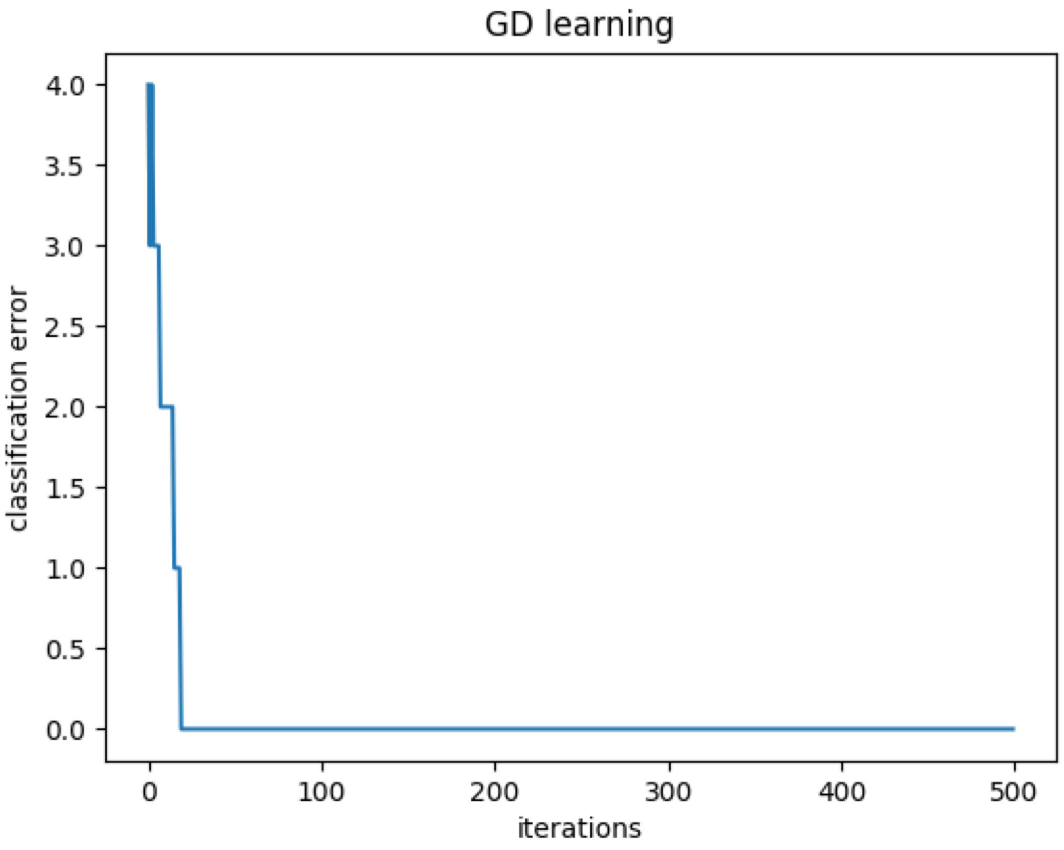
$$\boldsymbol{V} \leftarrow \boldsymbol{V} - \alpha \boldsymbol{H}^T \, \nabla_{\boldsymbol{U}} J = \begin{pmatrix} 0.92 & 0.05 & -0.26 \\ 0.68 & -0.36 & -0.19 \\ 0.22 & 1.05 & -0.10 \end{pmatrix}$$

$$\boldsymbol{c} \leftarrow \boldsymbol{c} - \alpha (\nabla_{\boldsymbol{U}} J)^T \mathbf{1}_P = \begin{pmatrix} -0.16 \\ 0.04 \\ 0.12 \end{pmatrix}$$

$$\boldsymbol{W} \leftarrow \boldsymbol{W} - \alpha \boldsymbol{X}^T \, \nabla_{\boldsymbol{Z}} J = \begin{pmatrix} -0.13 & 0.95 & 0.18 \\ -0.63 & 0.42 & 0.95 \end{pmatrix}$$

$$\boldsymbol{b} \leftarrow \boldsymbol{b} - \alpha (\nabla_{\boldsymbol{Z}} J)^T \mathbf{1}_P = \begin{pmatrix} -0.02 \\ 0.00 \\ -0.01 \end{pmatrix}$$


GD learning
entropy loss
iterations
0
1
2
3
4
5
6
7
0
100
200
300
400
500


GD learning
classification error
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
100
200
300
400
500
iterations

At convergence:

$$\boldsymbol{V} = \begin{pmatrix} 2.93 & -5.33 & 3.12 \\ 2.80 & 1.20 & -3.87 \\ 0.09 & 4.55 & -3.47 \end{pmatrix}, \qquad \boldsymbol{c} = \begin{pmatrix} -1.94 \\ -0.06 \\ 2.01 \end{pmatrix}$$

$$\boldsymbol{W} = \begin{pmatrix} -1.81 & 0.32 & 0.08 \\ -1.40 & 2.92 & 1.91 \end{pmatrix}, \qquad \boldsymbol{b} = \begin{pmatrix} 4.36 \\ 0.73 \\ -1.71 \end{pmatrix}$$

$$\boldsymbol{Y} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 2 \\ 2 \end{pmatrix}$$

Entropy = 0.138

Error = 0

3. Design a feedforward neural network consisting of two-hidden layers to approximate the following function:

$$\phi(x,y) = 0.8x^2 - y^3 + 2.5xy$$

for $-1.0 \leq x,y \leq 1.0$.

Use three ReLU neurons at each hidden layer and a linear neuron at the output layer.

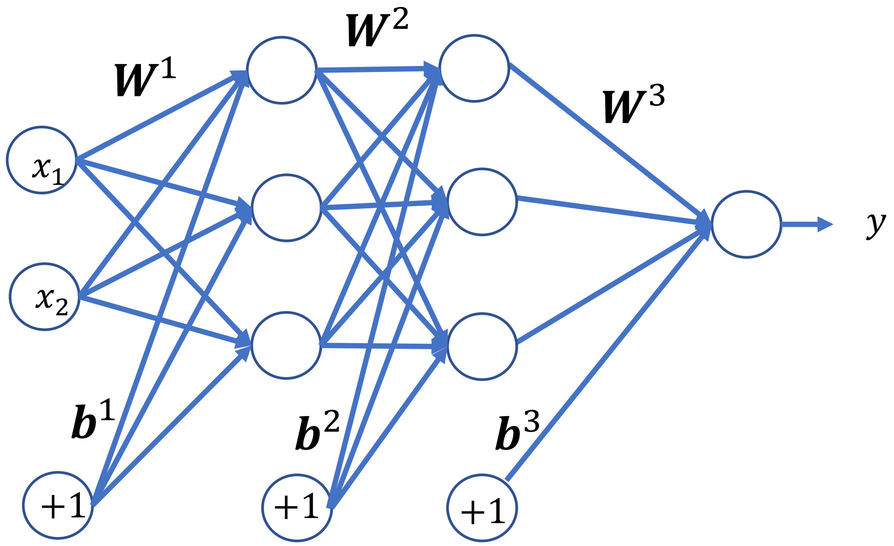
(a) Divide the input space equally into square regions of size 0.25×0.25 and use grid points as data to learn the function ϕ.

(b) Train the network using gradient decent learning at learning rate $\alpha = 0.01$ and plot the learning curve (mean square error vs. iterations) and the predicted data points.

(c) Compare the learning curves when learning the function at learning rates $\alpha = 0.005, 0.01, 0.05, \text{and } 0.1$.

$$\phi(x, y) = 0.8x^2 - x^3 + 2.5xy \quad \text{for } -1.0 \leq x, y \leq 1.0$$

Feedforward neural network with two hidden layers:

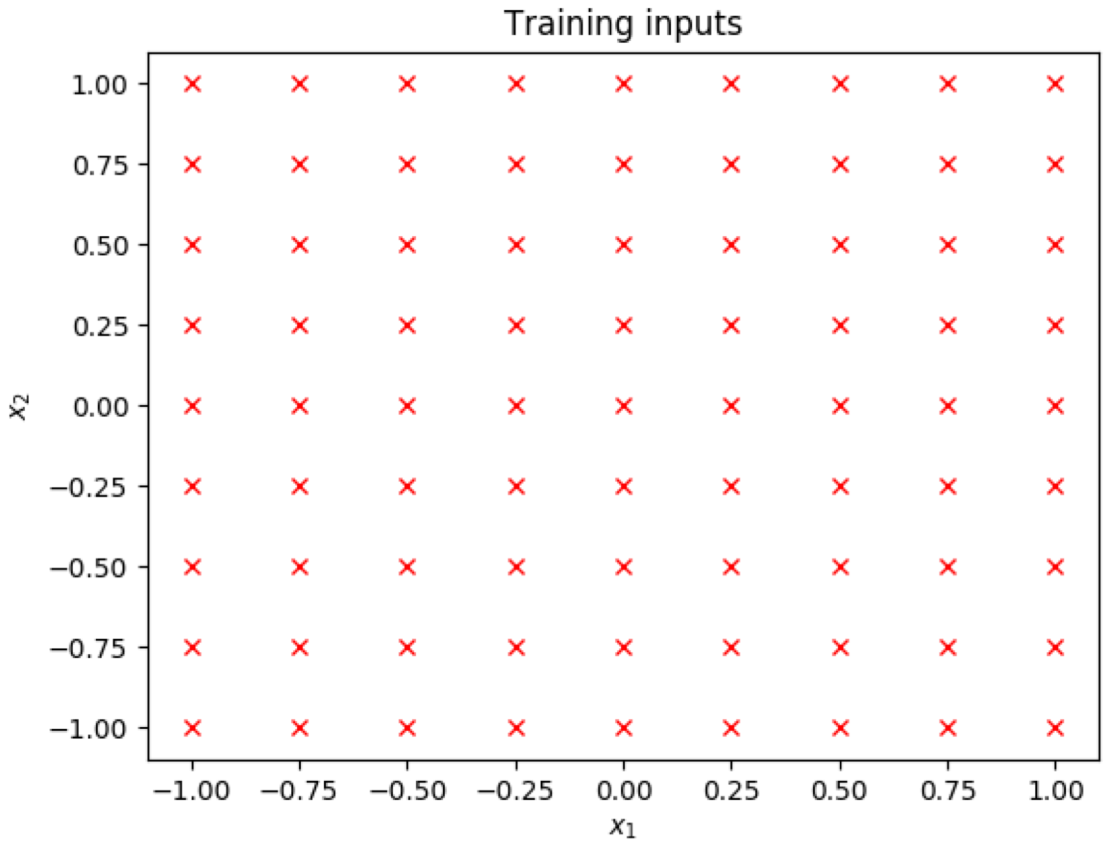


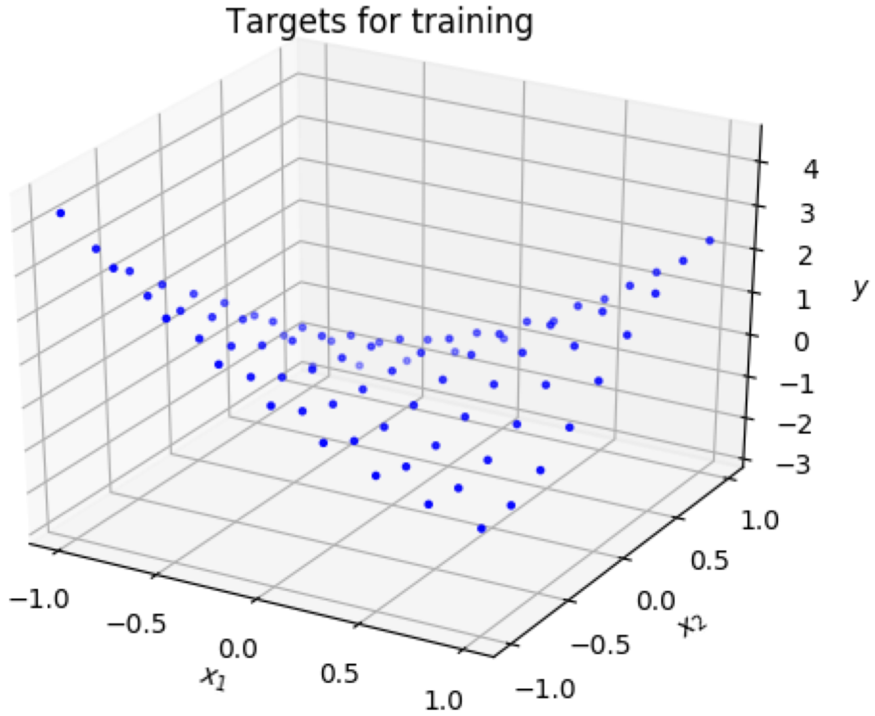
If input is $\boldsymbol{x} = (x_1, x_2)$, the output

$$y = 0.8{x_1}^2 - {x_2}^3 + 2.5x_1x_2$$

$$y = 0.8{x_1}^2 - {x_2}^3 + 2.5x_1x_2 \text{ for } -1.0 \le x_1, x_2 \le 1.0$$

Data points in a grid of size 0.25x0.25:





Forward propagation:

Input $(\boldsymbol{X}, \boldsymbol{D})$

$\boldsymbol{U}^1 = \boldsymbol{X}\boldsymbol{W}^1 + \boldsymbol{B}^1$

For layers $l = 1, 2, \cdots, L-1$:

$$\boldsymbol{H}^l = f^l(\boldsymbol{U}^l)$$

$$\boldsymbol{U}^{l+1} = \boldsymbol{H}^l\boldsymbol{W}^{l+1} + \boldsymbol{B}^{l+1}$$

$\boldsymbol{Y} = f^L(\boldsymbol{U}^L)$

Backward propagation:

If $l = L$:

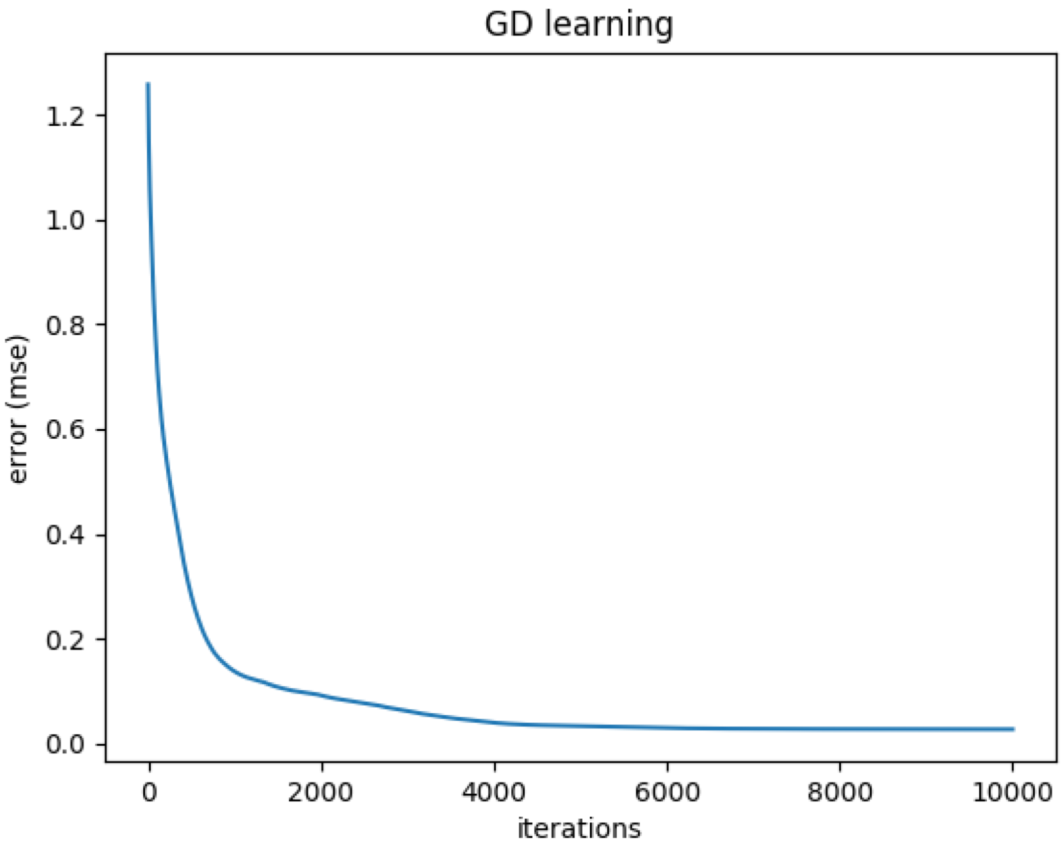
$$\nabla_{\boldsymbol{U}^l} J = -(\boldsymbol{D} - \boldsymbol{Y})$$

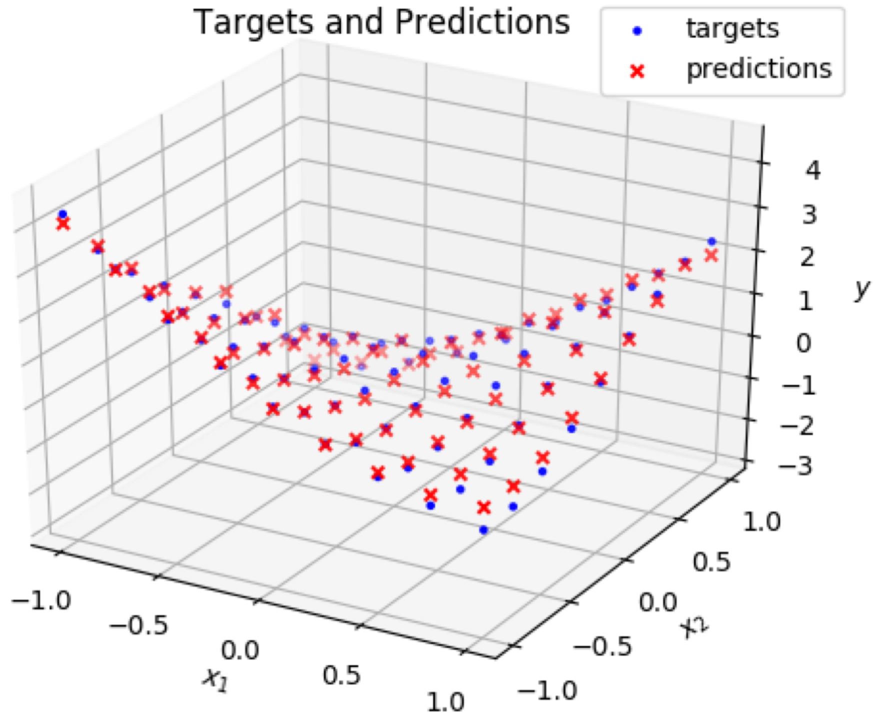
Else:

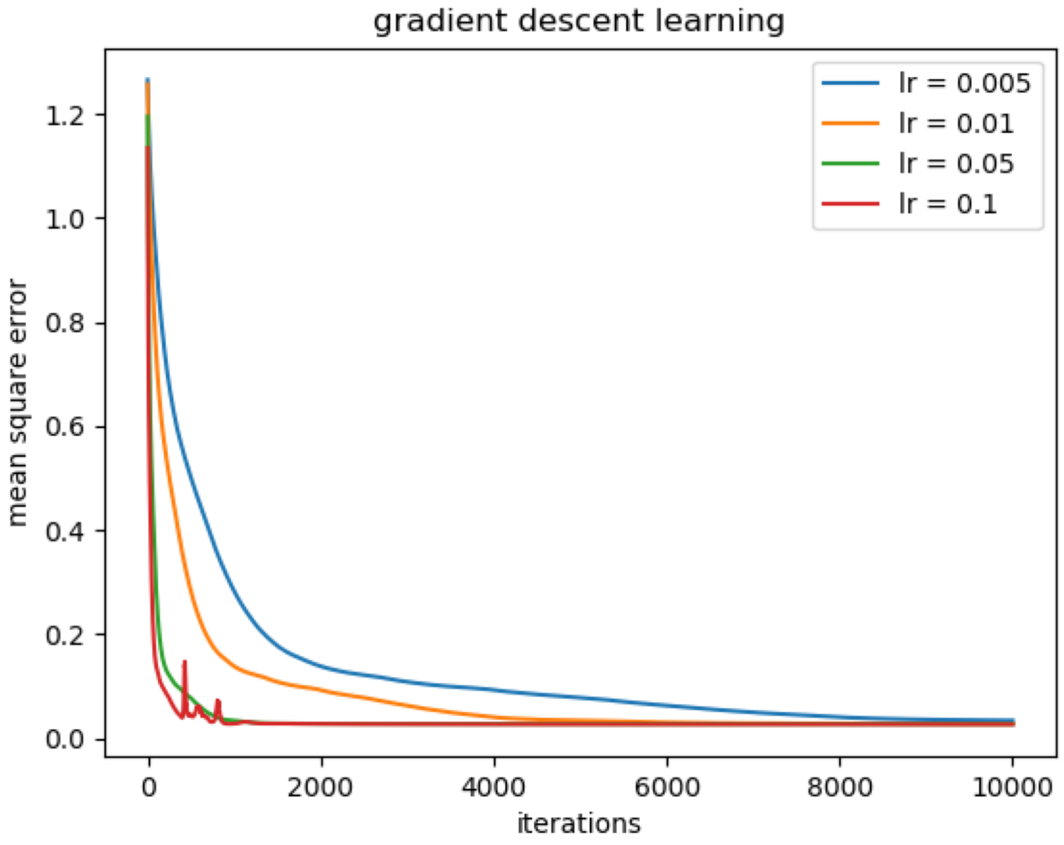
$$\nabla_{\boldsymbol{U}^l} J = \left(\nabla_{\boldsymbol{U}^{l+1}} J\right) {\boldsymbol{W}^{l+1}}^T \cdot f^{l\prime}\left(\boldsymbol{U}^l\right)$$

$$\nabla_{\boldsymbol{W}^l} J = {\boldsymbol{H}^{l-1}}^T \left(\nabla_{\boldsymbol{U}^l} J\right)$$

$$\nabla_{\boldsymbol{b}^l} J = \left(\nabla_{\boldsymbol{U}^l} J\right)^T \boldsymbol{1}_P$$


GD learning
error (mse)
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
2000
4000
6000
8000
10000
iterations


Targets and Predictions
targets
predictions
4
3
2
1
0
−1
−2
−3
y
−1.0
−0.5
0.0
0.5
1.0
x_1
1.0
0.5
0.0
−0.5
−1.0
x_2


gradient descent learning
lr = 0.005
lr = 0.01
lr = 0.05
lr = 0.1
mean square error
iterations
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
2000
4000
6000
8000
10000