Regression

CE/CZ4042 – Tutorial 2

1. Design a linear neuron to perform the following mapping:

$\boldsymbol{x} = (x_1, x_2, x_2)$	y
$(0.09 \quad -0.44 \quad -0.15)$	-2.57
$(0.69 \quad -0.99 \quad -0.76)$	-2.97
$(0.34 \quad 0.65 \quad -0.73)$	0.96
$(0.15 \quad 0.78 \quad -0.58)$	1.04
$(-0.63 \quad -0.78 \quad -0.56)$	-3.21
$(0.96 \quad 0.62 \quad -0.66)$	1.05
$(0.63 \quad -0.45 \quad -0.14)$	-2.39
$(0.88 \quad 0.64 \quad -0.33)$	0.66

Show one iteration of learning of the neuron with

(a) Stochastic gradient descent learning

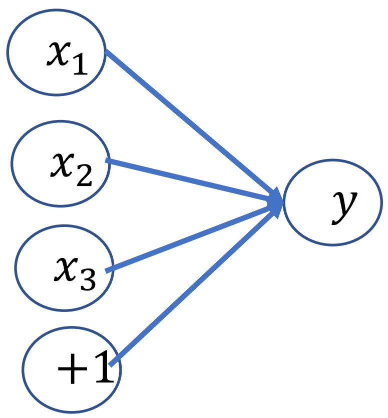
(b) Gradient descent learning

Initialize the weights randomly and biases to 0.0 and use a leaning factor $\alpha = 0.01$.

Plot the learning curves (mean square error vs. epochs) until convergence and find the learned weights and biases and the predicted values of y by the neuron.

Mapping:

$\boldsymbol{x} = (x_1, x_2, x_3)$	$\boldsymbol{d}$
$(0.09, -0.44, -0.15)$	-2.57
$(0.69, -0.99, -0.76)$	-2.97
$(0.34, 0.65, -0.73)$	0.96
$(0.15, , 0.78, -0.58)$	1.04
$(-0.63, -0.78, -0.56)$	-3.21
$(0.96, 0.62, -0.66)$	1.05
$(0.63, -0.45, -0.14)$	-2.39
$(0.88, 0.64, -0.33)$	0.66



$f(u) = u$

initial weights and biases: $\boldsymbol{w} = \begin{pmatrix} 0.77 \\ 0.02 \\ 0.63 \end{pmatrix}$, $b = 0.0$

Learning factor $\alpha = 0.01$

SGD for a linear neuron:

Given a training dataset $\{(\boldsymbol{x}_p, d_p)\}_{p=1}^{P}$

Set learning parameter α

Initialize $\boldsymbol{w}$ and b

Repeat until convergence:

For every training pattern $(\boldsymbol{x}_p, d_p)$:

Synaptic input $y_p = \boldsymbol{x}_p{}^T \boldsymbol{w} + b$

$\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha(d_p - y_p)\boldsymbol{x}_p$

$b \leftarrow b + \alpha(d_p - y_p)$

SGD:

learning factor $\alpha = 0.01$

Shuffle the inputs.

<u>Iteration 1</u>

Apply $\boldsymbol{x} = \begin{pmatrix} 0.34 \\ 0.65 \\ -0.73 \end{pmatrix}, d = 0.96$

$$y = \boldsymbol{x}^T \boldsymbol{w} + b = (0.34 \quad 0.65 \quad -0.15) \begin{pmatrix} 0.77 \\ 0.02 \\ 0.63 \end{pmatrix} + 0.0 = -0.19$$

$$s.e. = (d - y)^2 = (0.96 + 0.19)^2 = 1.32$$

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha(d - y)\boldsymbol{x} = \begin{pmatrix} 0.77 \\ 0.02 \\ 0.63 \end{pmatrix} + 0.01 \times (0.96 + 0.19) \begin{pmatrix} 0.34 \\ 0.65 \\ -0.73 \end{pmatrix} = \begin{pmatrix} 0.78 \\ 0.03 \\ 0.63 \end{pmatrix}$$

$$b = b + \alpha(d - y) = 0.0 + 0.01 \times (0.96 + 0.19) = 0.01$$

Apply $\boldsymbol{x} = \begin{pmatrix} 0.63 \\ -0.45 \\ -0.14 \end{pmatrix}, d = -2.39$

$$y = \boldsymbol{x}^T \boldsymbol{w} + b = (0.63 \quad -0.45 \quad -0.14) \begin{pmatrix} 0.78 \\ 0.03 \\ 0.63 \end{pmatrix} + 0.01 = 0.4$$

$$s.e. = (d - y)^2 = (-2.39 - 0.4)^2 = 7.78$$

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha(d - y)\boldsymbol{x} = \begin{pmatrix} 0.78 \\ 0.03 \\ 0.63 \end{pmatrix} + 0.01 \times (-2.39 - 0.4) \begin{pmatrix} 0.63 \\ -0.45 \\ -0.14 \end{pmatrix} = \begin{pmatrix} 0.76 \\ 0.04 \\ 0.63 \end{pmatrix}$$

$$b = b + \alpha(d - y) = 0.01 + 0.01 \times (-2.39 - 0.4) = -0.02$$

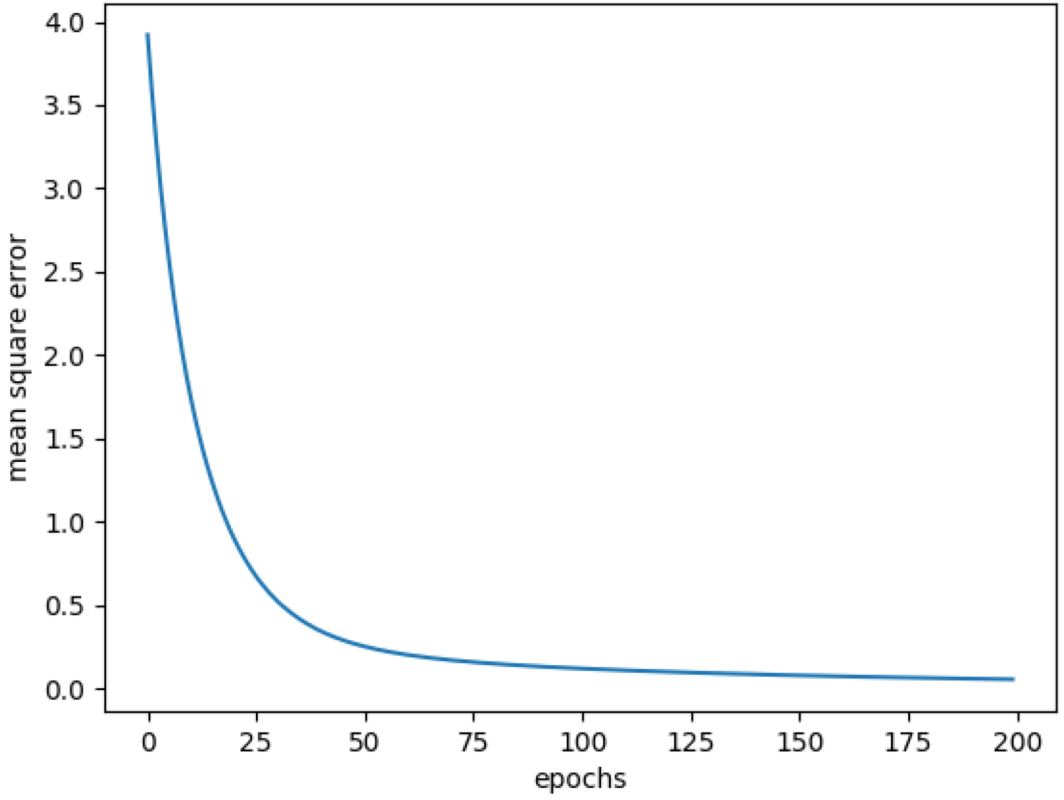
Continue apply other patterns

Then continue epochs 2, 3, …… until convergence.

Iteration 1

$\boldsymbol{x}$	d	y	s.e.	$\boldsymbol{w}$	b
$\begin{pmatrix} 0.34 \\ 0.65 \\ -0.73 \end{pmatrix}$	0.96	-0.19	1.31	$\begin{pmatrix} 0.78 \\ 0.03 \\ 0.63 \end{pmatrix}$	0.01
$\begin{pmatrix} 0.63 \\ -0.45 \\ -0.14 \end{pmatrix}$	-2.39	0.4	7.78	$\begin{pmatrix} 0.76 \\ 0.04 \\ 0.63 \end{pmatrix}$	-0.02
$\begin{pmatrix} 0.88 \\ 0.64 \\ -0.33 \end{pmatrix}$	0.66	0.47	0.04	$\begin{pmatrix} 0.76 \\ 0.04 \\ 0.63 \end{pmatrix}$	-0.01
$\begin{pmatrix} 0.96 \\ 0.62 \\ -0.66 \end{pmatrix}$	1.05	0.33	0.52	$\begin{pmatrix} 0.77 \\ 0.05 \\ 0.62 \end{pmatrix}$	-0.01
$\begin{pmatrix} 0.09 \\ -0.44 \\ -0.15 \end{pmatrix}$	-2.57	-0.05	6.34	$\begin{pmatrix} 0.76 \\ 0.06 \\ 0.63 \end{pmatrix}$	-0.03
$\begin{pmatrix} 0.69 \\ -0.99 \\ -0.76 \end{pmatrix}$	-2.97	-0.04	8.59	$\begin{pmatrix} 0.74 \\ 0.09 \\ 0.65 \end{pmatrix}$	-0.06
$\begin{pmatrix} -0.63 \\ -0.78 \\ -0.56 \end{pmatrix}$	-3.21	-0.96	5.05	$\begin{pmatrix} 0.76 \\ 0.10 \\ 0.66 \end{pmatrix}$	-0.08
$\begin{pmatrix} 0.15 \\ 0.78 \\ -0.58 \end{pmatrix}$	1.04	-0.27	1.73	$\begin{pmatrix} 0.76 \\ 0.11 \\ 0.65 \end{pmatrix}$	-0.07

m.s.e = 3.92


4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
mean square error
0
25
50
75
100
125
150
175
200
epochs

At convergence:

$$\boldsymbol{w} = \begin{pmatrix} 0.37 \\ 2.57 \\ -0.21 \end{pmatrix}, b = -1.17$$

mse = 0.054

If $\boldsymbol{x} = (x_1, x_2, x_3)^T$, the learned function by the linear neuron:

$$y = \boldsymbol{w^T}\boldsymbol{x} + b$$

$$y = \begin{pmatrix} 0.37 & 2.57 & -0.21 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} - 1.17$$

$$y = 0.37x_1 + 2.57x_2 - 0.21x_3 - 1.17$$

Predicted values:

```
x: [-0.63 -0.78 -0.56], d: -3.21, y: -3.28035
x: [ 0.96  0.62 -0.66], d: 1.05, y: 0.919454
x: [ 0.09 -0.44 -0.15], d: -2.57, y: -2.22926
x: [ 0.88  0.64 -0.33], d: 0.66, y: 0.871378
x: [ 0.34  0.65 -0.73], d: 0.96, y: 0.782616
x: [ 0.15  0.78 -0.58], d: 1.04, y: 1.01435
x: [ 0.69 -0.99 -0.76], d: -2.97, y: -3.29005
x: [ 0.63 -0.45 -0.14], d: -2.39, y: -2.0579
```

GD for a linear neuron

Given a training dataset$(\boldsymbol{X}, \boldsymbol{d})$
Set the learning parameterα
Initialize $\boldsymbol{w}$ and b
Repeat until convergence:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w} + b\mathbf{1}_P$$

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha \boldsymbol{X}^T(\boldsymbol{d} - \boldsymbol{y})$$

$$b \leftarrow b + \alpha {\mathbf{1}_P}^T(\boldsymbol{d} - \boldsymbol{y})$$

GD for a linear neuron

$$\boldsymbol{X} = \begin{pmatrix} 0.09 & -0.44 & -0.15 \\ 0.69 & -0.99 & -0.76 \\ 0.34 & 0.65 & -0.73 \\ 0.15 & 0.78 & -0.58 \\ -0.63 & -0.78 & -0.56 \\ 0.96 & 0.62 & -0.66 \\ 0.63 & -0.45 & -0.14 \\ 0.88 & 0.64 & -0.33 \end{pmatrix}, \boldsymbol{d} = \begin{pmatrix} -2.57 \\ -2.97 \\ 0.96 \\ 1.04 \\ -3.21 \\ 1.05 \\ -2.39 \\ 0.66 \end{pmatrix}$$

initial weights and biases: $\boldsymbol{w} = \begin{pmatrix} 0.77 \\ 0.02 \\ 0.63 \end{pmatrix}, b = 0.0$

Learning factor $\alpha = 0.01$

$$\text{Output } \boldsymbol{y} = \boldsymbol{X}\boldsymbol{w} + b\mathbf{1}_P = \begin{pmatrix} 0.09 & -0.44 & -0.15 \\ 0.69 & -0.99 & -0.76 \\ 0.34 & 0.65 & -0.73 \\ 0.15 & 0.78 & -0.58 \\ -0.63 & -0.78 & -0.56 \\ 0.96 & 0.62 & -0.66 \\ 0.63 & -0.45 & -0.14 \\ 0.88 & 0.64 & -0.33 \end{pmatrix} \begin{pmatrix} 0.77 \\ 0.02 \\ 0.63 \end{pmatrix} - 0.0 \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -0.03 \\ 0.03 \\ -0.18 \\ -0.24 \\ -0.85 \\ 0.34 \\ 0.39 \\ 0.48 \end{pmatrix}$$

$$\text{m.s.e.} = \frac{1}{8}\sum_{p=1}^{8}\left(d_p - y_p\right)^2 = 4.02$$

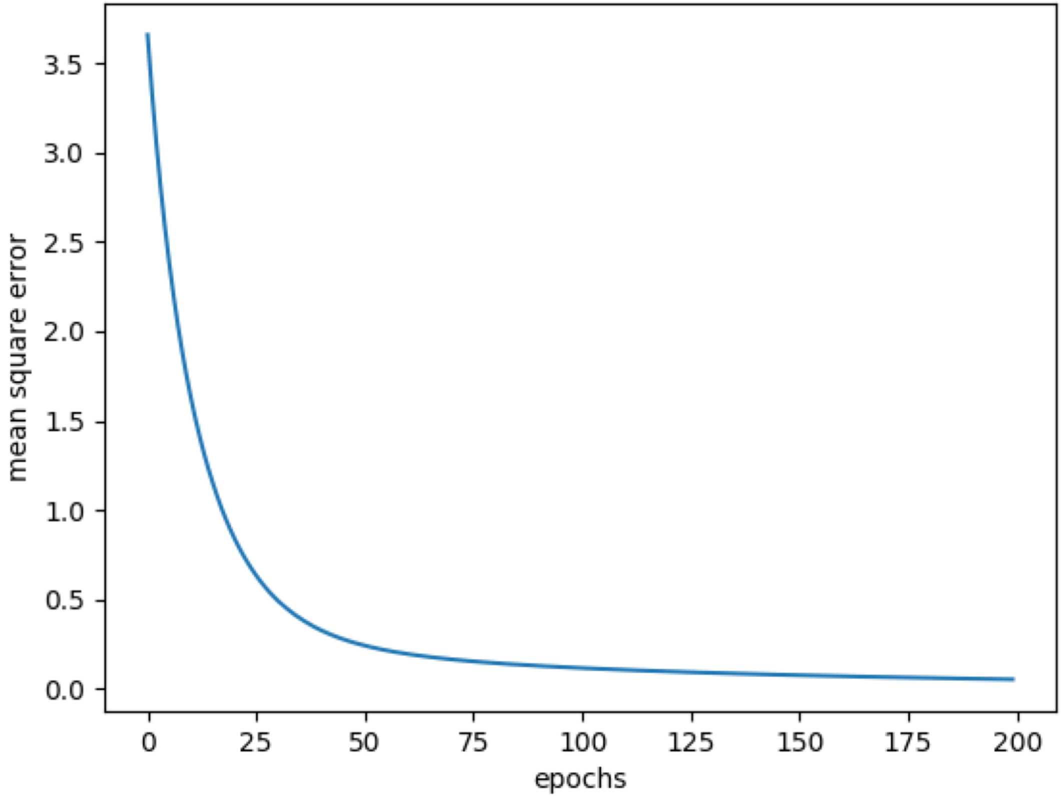
$$\boldsymbol{w} = \boldsymbol{w} + \alpha \boldsymbol{X}^{T}(\boldsymbol{d} - \boldsymbol{y})$$

$$= \begin{pmatrix} 0.77 \\ 0.02 \\ 0.63 \end{pmatrix} + 0.01 \times \begin{pmatrix} 0.09 & 0.69 & 0.34 & 0.15 & -0.63 & 0.96 & 0.63 & 0.88 \\ -0.44 & -0.99 & 0.65 & 0.78 & -0.78 & 0.62 & -0.45 & 0.64 \\ -0.15 & -0.76 & -0.73 & -0.58 & -0.56 & -0.66 & -0.14 & -0.33 \end{pmatrix} \left(\begin{pmatrix} -2.57 \\ -2.97 \\ 0.96 \\ 1.04 \\ -3.21 \\ 1.05 \\ -2.39 \\ 0.66 \end{pmatrix} - \begin{pmatrix} -0.03 \\ 0.03 \\ -0.18 \\ -0.24 \\ -0.85 \\ 0.34 \\ 0.39 \\ 0.48 \end{pmatrix} \right)$$

$$= \begin{pmatrix} 0.76 \\ 0.12 \\ 0.66 \end{pmatrix}$$

$$b = b + \alpha \boldsymbol{1}_P{}^{T}(\boldsymbol{d} - \boldsymbol{y}) = 0.0 + 0.01 \times \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \left(\begin{pmatrix} -2.57 \\ -2.97 \\ 0.96 \\ 1.04 \\ -3.21 \\ 1.05 \\ -2.39 \\ 0.66 \end{pmatrix} - \begin{pmatrix} -0.03 \\ 0.03 \\ -0.18 \\ -0.24 \\ -0.85 \\ 0.34 \\ 0.39 \\ 0.48 \end{pmatrix} \right) = -0.07$$

iteration	$\boldsymbol{y}$	mse	$\boldsymbol{w}$	b
2	$\begin{pmatrix} -0.15 \\ -0.16 \\ -0.22 \\ -0.25 \\ -1.01 \\ 0.29 \\ 0.26 \\ 0.45 \end{pmatrix}$	3.36	$\begin{pmatrix} 0.75 \\ 0.21 \\ 0.67 \end{pmatrix}$	-0.14
200	$\begin{pmatrix} -2.23 \\ -3.29 \\ 0.78 \\ 1.01 \\ -3.28 \\ 0.92 \\ -2.06 \\ 0.87 \end{pmatrix}$	0.05	$\begin{pmatrix} 0.368 \\ 2.56 \\ -0.22 \end{pmatrix}$	-1.163



mean square error
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
25
50
75
100
125
150
175
200
epochs

At convergence:

$$\boldsymbol{w} = \begin{pmatrix} 0.37 \\ 2.57 \\ -0.21 \end{pmatrix}, b = -1.16$$

mse = 0.054

If $\boldsymbol{x} = (x_1, x_2, x_3)^T$, the learned function by the linear neuron:

$$y = \boldsymbol{w^T x} + b$$

$$y = \begin{pmatrix} 0.37 & 2.57 & -0.21 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} - 1.16$$

$$y = 0.37x_1 + 2.57x_2 - 0.21x_3 - 1.16$$

Predicted values

x	$\boldsymbol{d}$	SGD	GD
$(0.09, -0.44, -0.15)$	-2.57	-2.23	-2.23
$(0.69, -0.99, -0.76)$	-2.97	-3.29	-3.29
$(0.34, 0.65, -0.73)$	0.96	0.78	0.78
$(0.15, , 0.78, -0.58)$	1.04	1.01	1.01
$(-0.63, -0.78, -0.56)$	-3.21	-3.28	-3.28
$(0.96, 0.62, -0.66)$	1.05	0.92	0.92
$(0.63, -0.45, -0.14)$	-2.39	-2.06	-2.06
$(0.88, 0.64, -0.33)$	0.66	0.87	0.88
$\boldsymbol{w}$		$\begin{pmatrix} 0.369 \\ 2.566 \\ -0.212 \end{pmatrix}$	$\begin{pmatrix} 0.368 \\ 2.567 \\ -0.207 \end{pmatrix}$
b		-1.165	-1.163
m.s.e.		0.055	0.054

2. Design a perceptron to approximate the function y:

$$y = 0.5 + x_1 + 3{x_2}^2$$

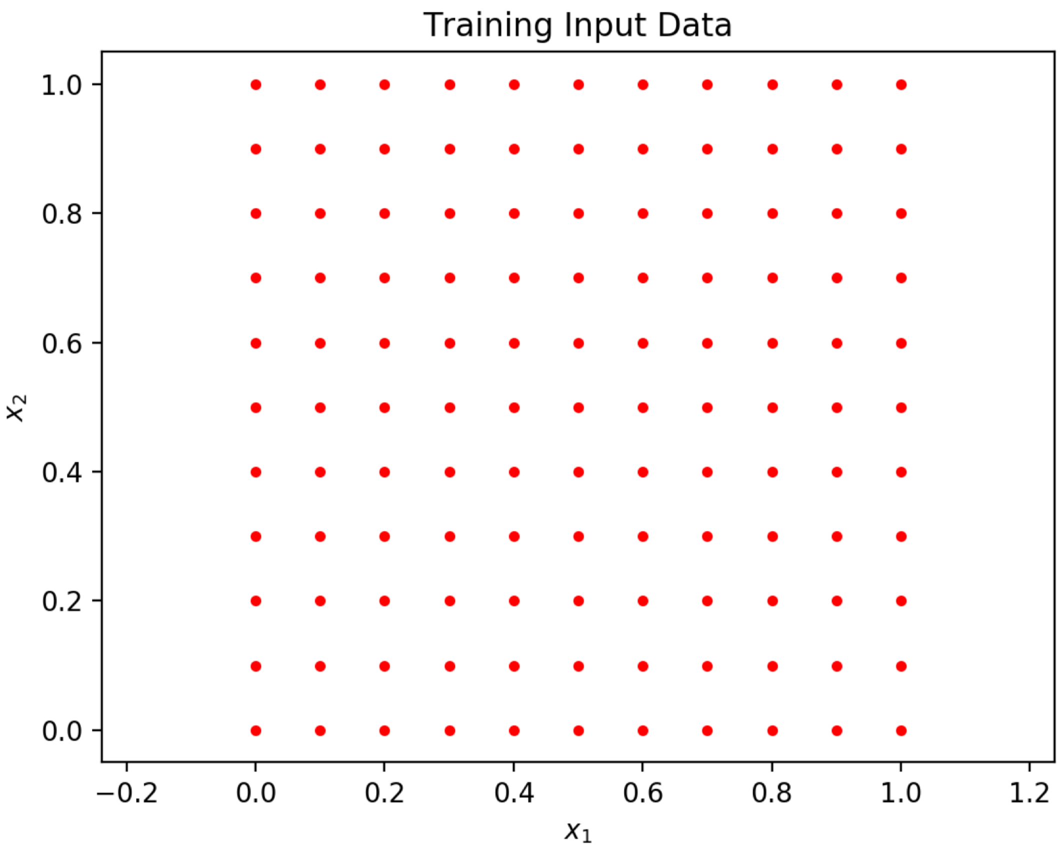
for inputs $0 \leq x_1, x_2 \leq 1$.

(a) Divide the input space into a grid of squares of size 0.1×0.1 and use the grid points as training data.
(b) Use gradient descent learning to train the perceptron with a learning rate $\alpha = 0.01$. Plot the mean square error against epochs until convergence
(c) Plot the targets and the predicted data points, and estimate the mean square error of the prediction.
(d) Repeat above (a)-(c) with stochastic gradient descent learning and compare the performance.

Q3

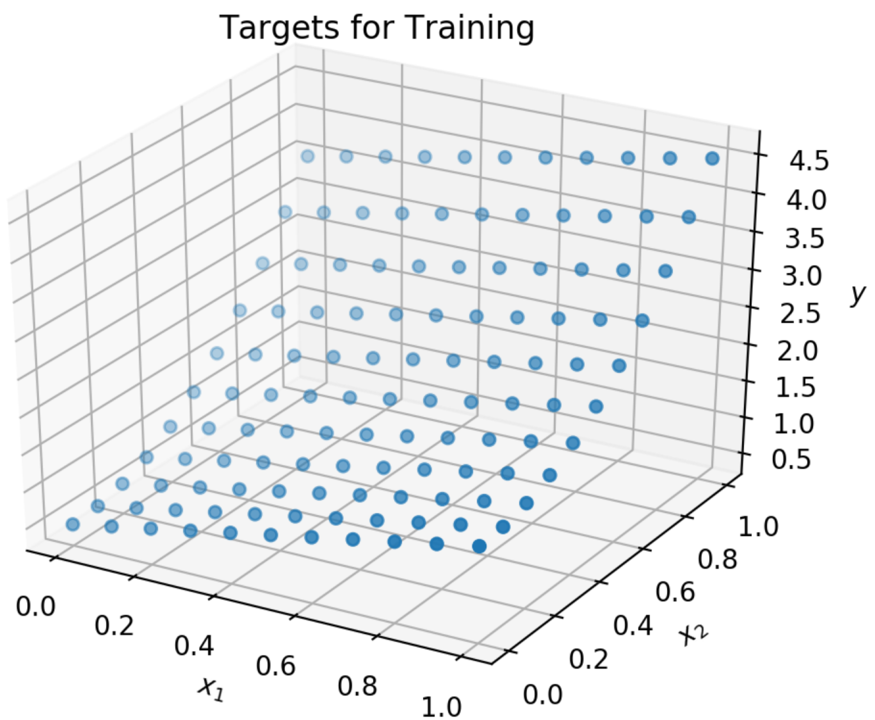
$y = 0.5 + x_1 + 3{x_2}^2$

Input space: $0 \leq x_1, x_2 \leq 1.$
Training Inputs are in a grid of size 0.1x0.1



Target outputs for training can be generated by substituting inputs (x_1, x_2) in

$$y = 0.5 + x_1 + 3{x_2}^2$$



$y = 0.5 + x_1 + 3{x_2}^2$

For input space: $0 \leq x_1, x_2 \leq 1$

In this space, the function is an increasing function and

$$\frac{\partial y}{\partial x_1} = 1, \frac{\partial y}{\partial x_2} = 6x_2$$

That is, minimum occurs at (0, 0) and maximum occurs at (1, 1):

$$y \in [0.5,\ 4.5]$$

So, the activation function $g(u)$ can be a sigmoidal function with amplitude 4.0 and shifted up by 0.5:

$$y = g(u) = 4f(u) + 0.5$$

where $f(u) = \frac{1}{1+e^{-u}}$:

$$y = g(u) = 4f(u) + 0.5$$

where $f(u) = \frac{1}{1+e^{-u}}$.

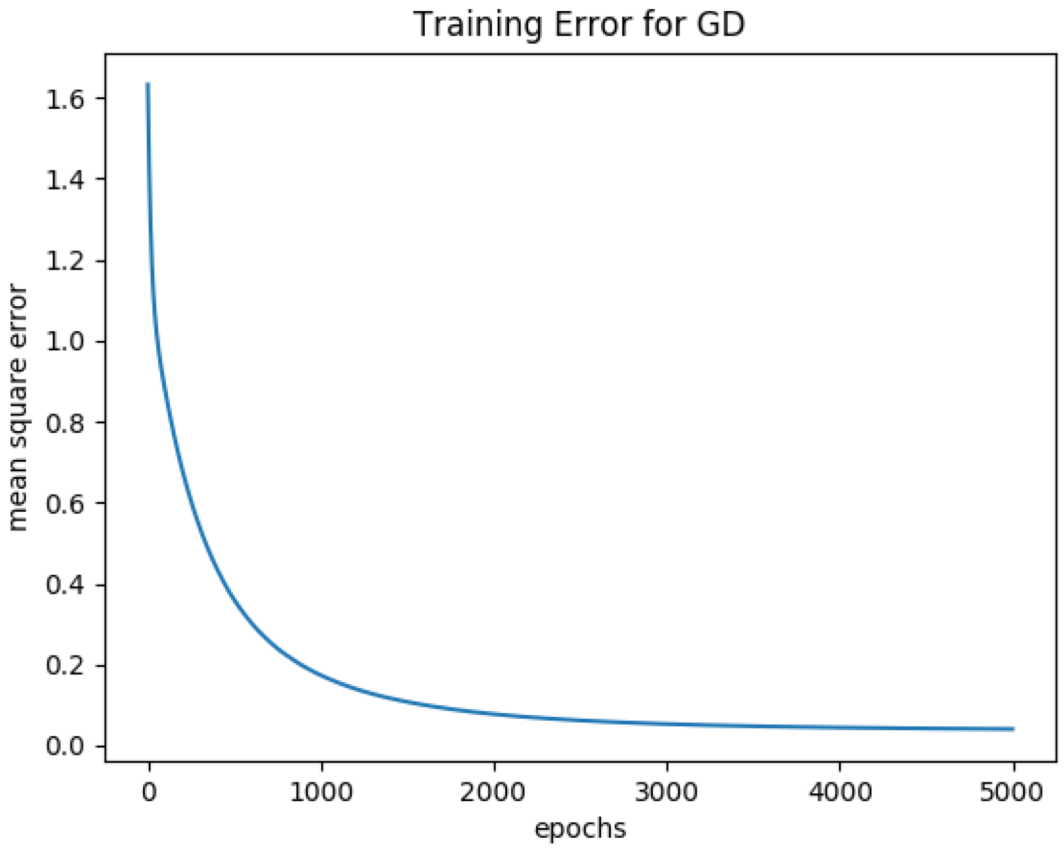
We need the gradient of the activation function in order to apply gradient descent learning:

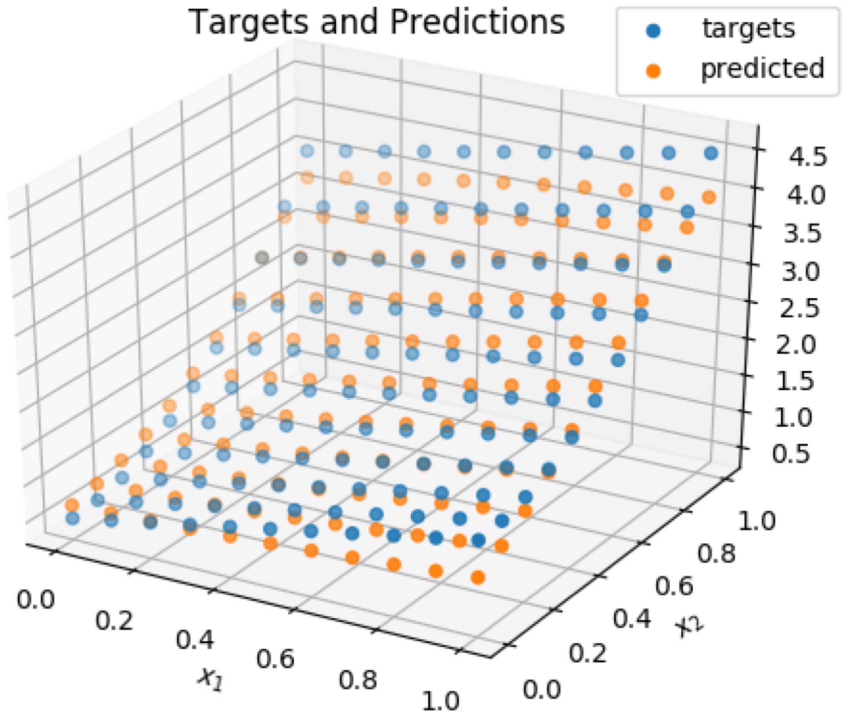
$$g'(u) = 4f'(u) = 4f(u)\big(1 - f(u)\big) = (y - 0.5)\left(1 - \frac{y - 0.5}{4}\right)$$

Gradient descent for perceptron

Given a training dataset$(\boldsymbol{X}, \boldsymbol{d})$
Set learning parameter α
Initialize $\boldsymbol{w}$ and b
Repeat until convergence:

$$\boldsymbol{u} = \boldsymbol{X}\boldsymbol{w} + b\boldsymbol{1}_P$$
$$\boldsymbol{y} = f(\boldsymbol{u})$$
$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha\boldsymbol{X}^T(\boldsymbol{d} - \boldsymbol{y}) \cdot f'(\boldsymbol{u})$$
$$b \leftarrow b + \alpha\boldsymbol{1}_P{}^T(\boldsymbol{d} - \boldsymbol{y}) \cdot f'(\boldsymbol{u})$$


Training Error for GD
mean square error
epochs
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
0
1000
2000
3000
4000
5000


Targets and Predictions
targets
predicted
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
y
0.0
0.2
0.4
0.6
0.8
1.0
x_1
0.0
0.2
0.4
0.6
0.8
1.0
x_2

SGD for perceptron

Given a training dataset $\{(\boldsymbol{x}_p, d_p)\}_{p=1}^{P}$

Set learning parameter α

Initialize $\boldsymbol{w}$ and b

Repeat until convergence:

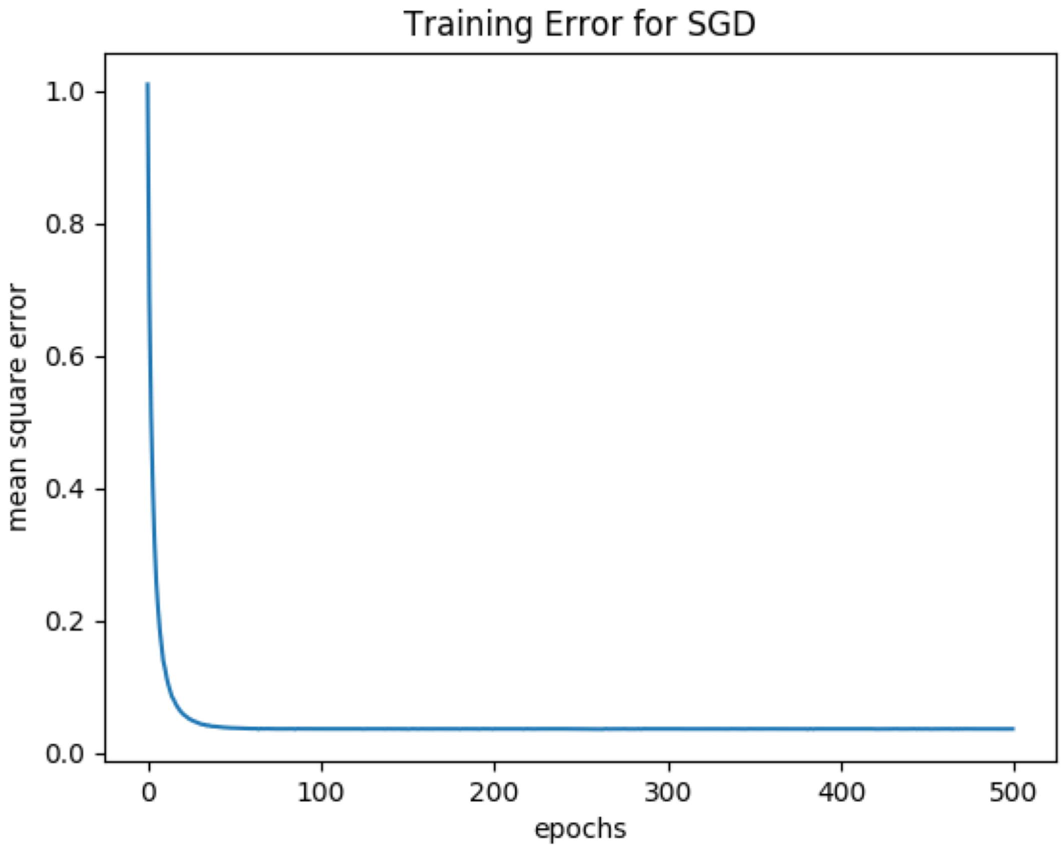
For every training pattern $(\boldsymbol{x}_p, d_p)$:

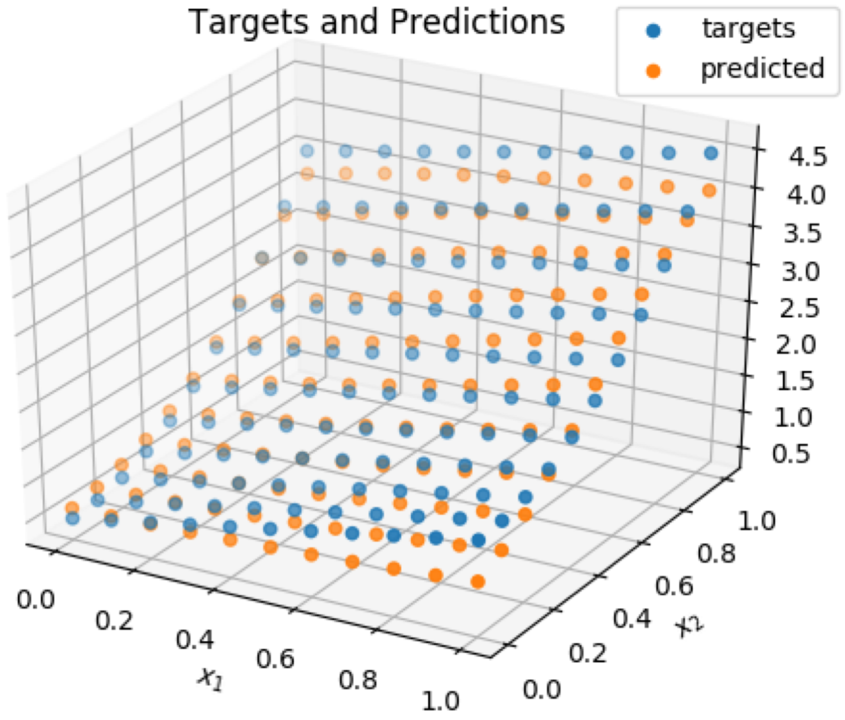
$$u_p = \boldsymbol{x}_p{}^T\boldsymbol{w} + b$$

$$y_p = f(\,u_p)$$

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha(d_p - y_p)f'(\,u_p)\boldsymbol{x}_p$$

$$b \leftarrow b + \alpha(d_p - y_p)f'(\,u_p)$$


Training Error for SGD
mean square error
epochs
1.0
0.8
0.6
0.4
0.2
0.0
0
100
200
300
400
500


Targets and Predictions
targets
predicted
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
y
0.0
0.2
0.4
0.6
0.8
1.0
x_1
0.0
0.2
0.4
0.6
0.8
1.0
x_2

GD vs. SGD

	GD	SGD
$\boldsymbol{w}$	$\begin{pmatrix}1.133\\3.724\end{pmatrix}$	$\begin{pmatrix}1.269\\4.050\end{pmatrix}$
b	-3.058	-3.328
m.s.e.	0.041	0.037

3. Train a linear neuron to learn the following function ϕ:

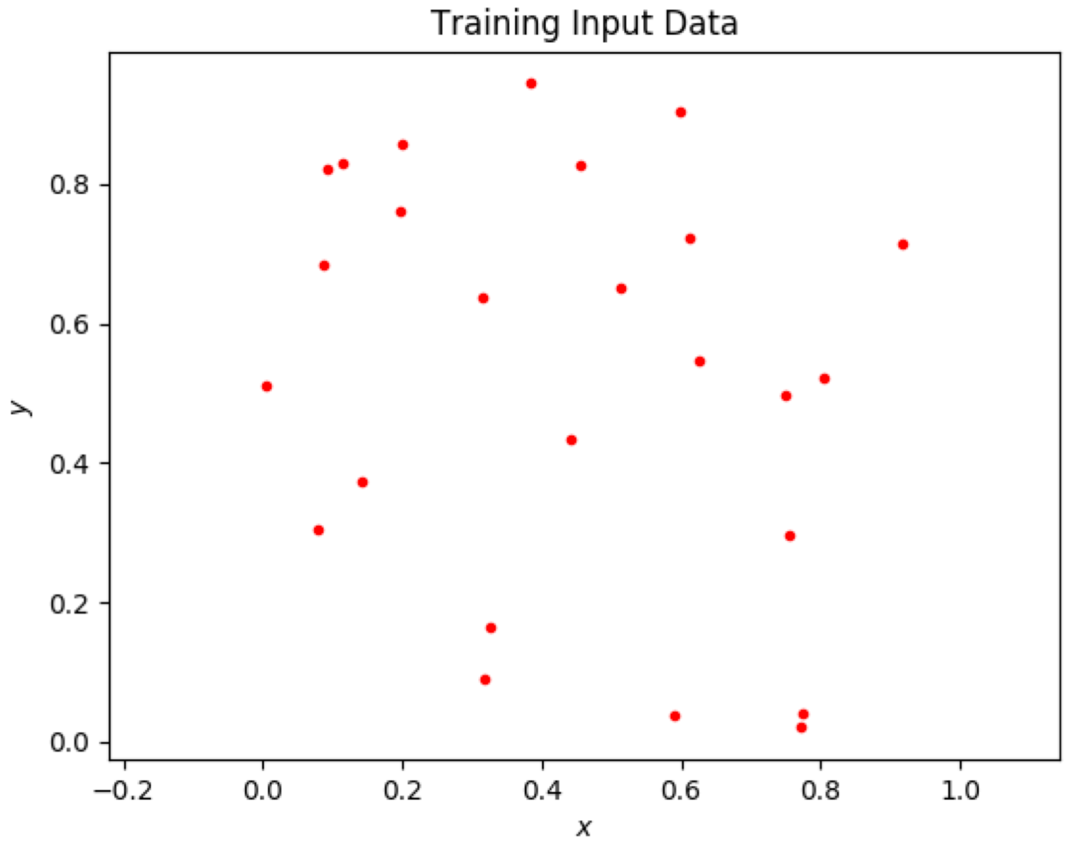
$$\phi(x, y) = 1.5 + 3.3x - 2.5y + 0.2xy$$

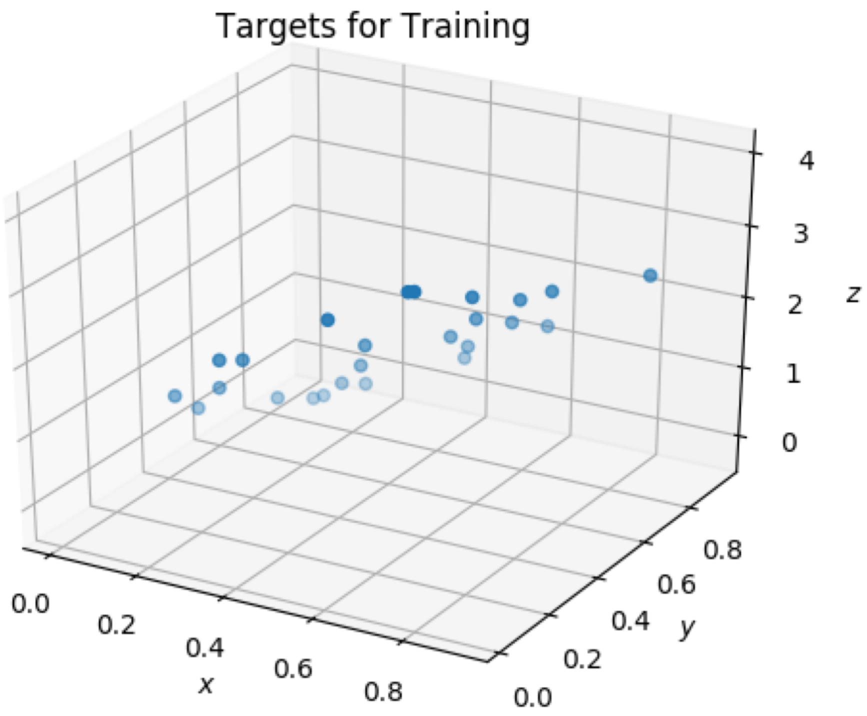
for $0 \leq x, y \leq 1$.

(a) Sample 25 data points randomly from the input space for training.
(b) Use the gradient descent algorithm to train a linear neuron
(c) Compute the training error and plot the function approximated by the linear neuron.
(d) Repeat (b) and (c) for a perceptron and compare the results with those of the linear neuron.

$$z = \phi(x, y) = 1.5 + 3.3x - 2.5y + 0.2xy$$

$$0 \le x, y \le 1$$



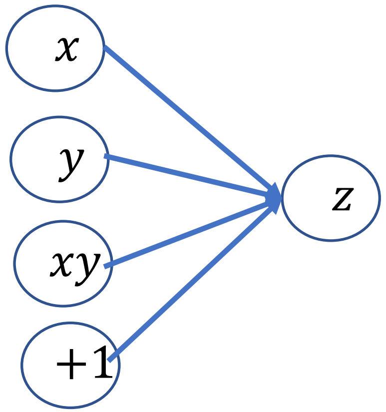


$$z = 1.5 + 3.3x - 2.5y + 0.2xy$$

Linear neuron learns a linear function. The above equation can be written as a linear equation:

$$z = 1.5 + 3.3x_1 - 2.5x_2 + 0.2x_3$$

where the linear neuron receives 3 inputs: $x_1 = x$, $x_2 = y$, and $x_3 = xy$.



$$f(u) = u$$

GD for a linear neuron

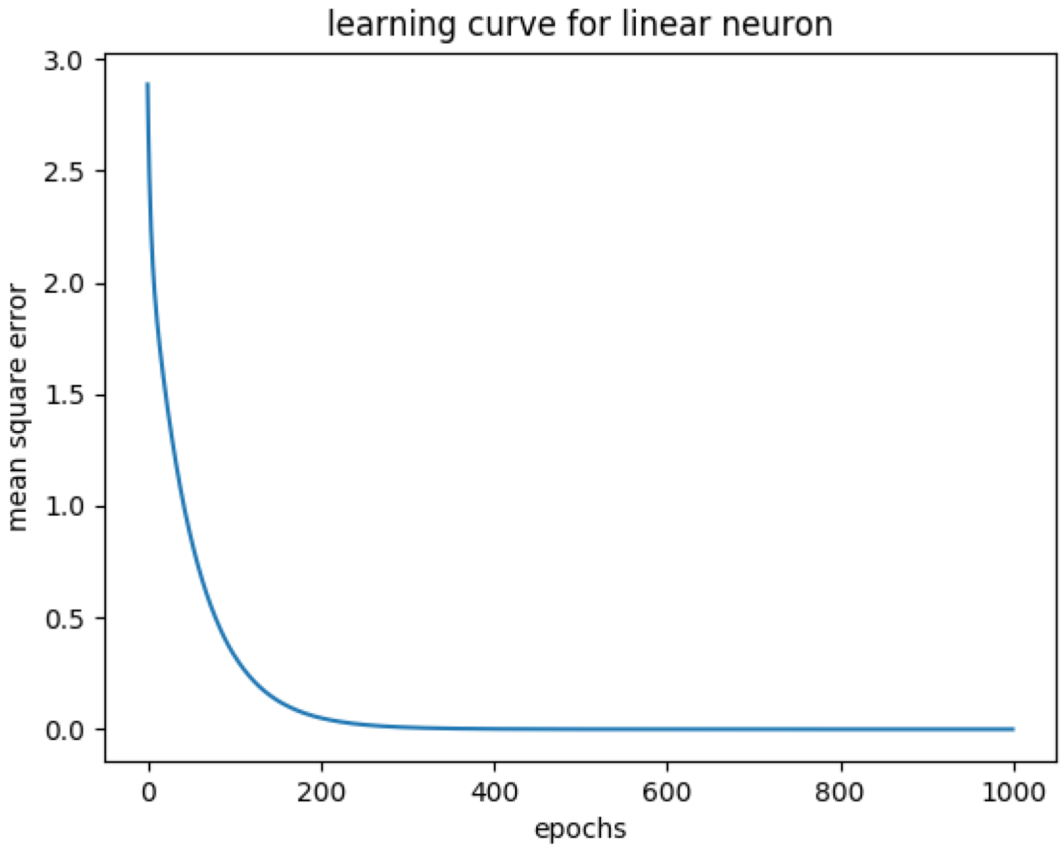
Given a training dataset$(\boldsymbol{X}, \boldsymbol{d})$

Set the learning parameterα

Initialize $\boldsymbol{w}$ and b

Repeat until convergence:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w} + b\boldsymbol{1}_P$$
$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha\boldsymbol{X}^T(\boldsymbol{d} - \boldsymbol{y})$$
$$b \leftarrow b + \alpha\boldsymbol{1}_P{}^T(\boldsymbol{d} - \boldsymbol{y})$$


learning curve for linear neuron
mean square error
epochs
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
200
400
600
800
1000

At convergence,

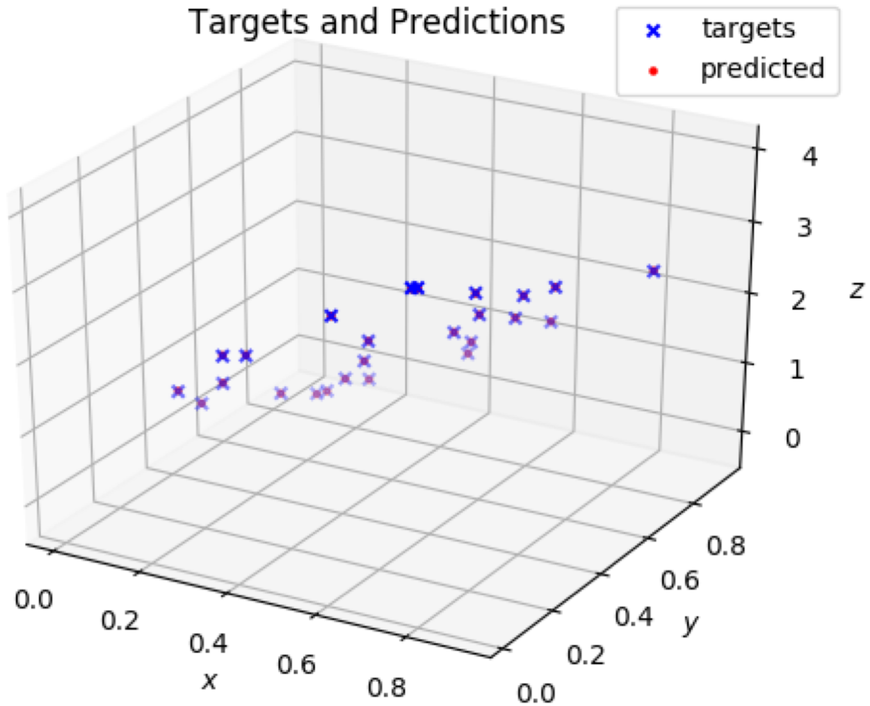
Weights $w = \begin{pmatrix} 3.25 \\ -2.55 \\ 0.29 \end{pmatrix}$ and bias $b = 1.53$

Mean square error = 3.97 x 10^{-5}

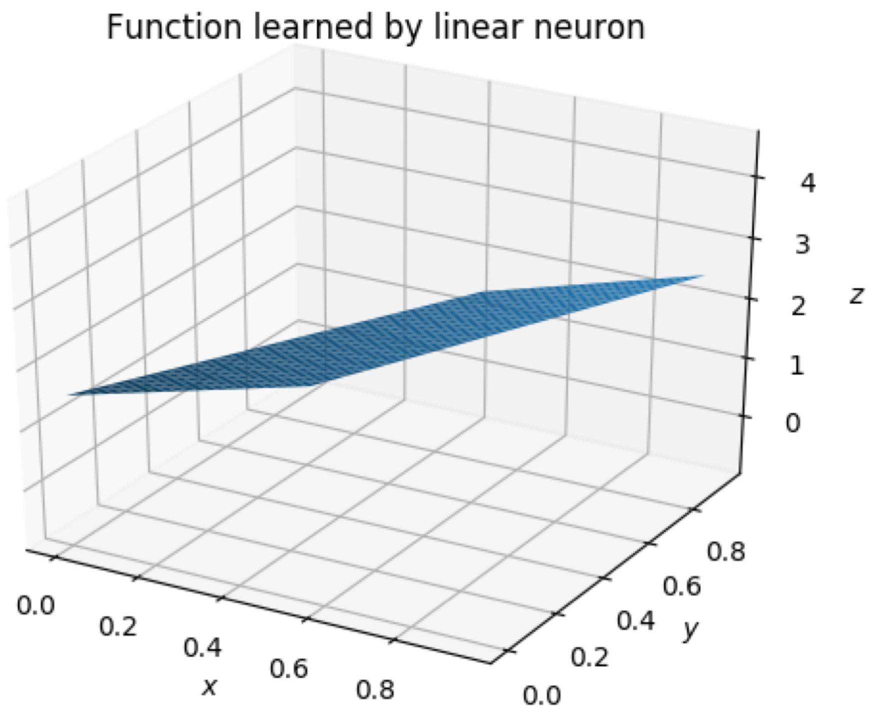
The function learned by the linear neuron:

$$z = w_1 x_1 + w_2 x_2 + w_3 x_3 + b$$

$$z = 3.25x - 2.55y + 0.29xy + 1.53$$

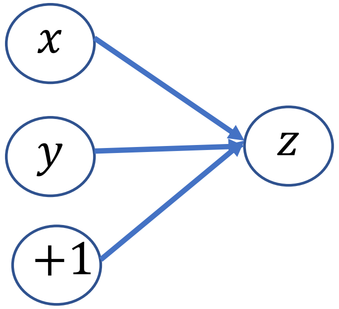


Targets and Predictions
targets
predicted
0.0
0.2
0.4
0.6
0.8
x
0.0
0.2
0.4
0.6
0.8
y
0
1
2
3
4
z



Function learned by linear neuron
0
1
2
3
4
z
0.0
0.2
0.4
0.6
0.8
x
0.0
0.2
0.4
0.6
0.8
y

For perceptron learning, the activation function is a sigmoid function and the range of output should be known.



$$z = \phi(x, y) = 1.5 + 3.3x - 2.5y + 0.2xy$$

where $0 \le x, y \le 1$

Differentiating the function to find the maximum and minimum points:

.

$$\frac{\partial z}{\partial x} = 3.3 + 0.2y = 0 \ \rightarrow y = -16.5$$
$$\frac{\partial z}{\partial y} = -2.5 + 0.2x = 0 \ \rightarrow x = 12.5$$

That is, maximum and minimum occur at boundary points.

$(\mathbf{0}, \mathbf{0})$	$(\mathbf{1}, \mathbf{0})$	$(\mathbf{0}, \mathbf{1})$	$(\mathbf{1}, \mathbf{1})$
1.5	4.8	-1.0	2.5

Therefore, $z \in [-1.0, 4.8]$

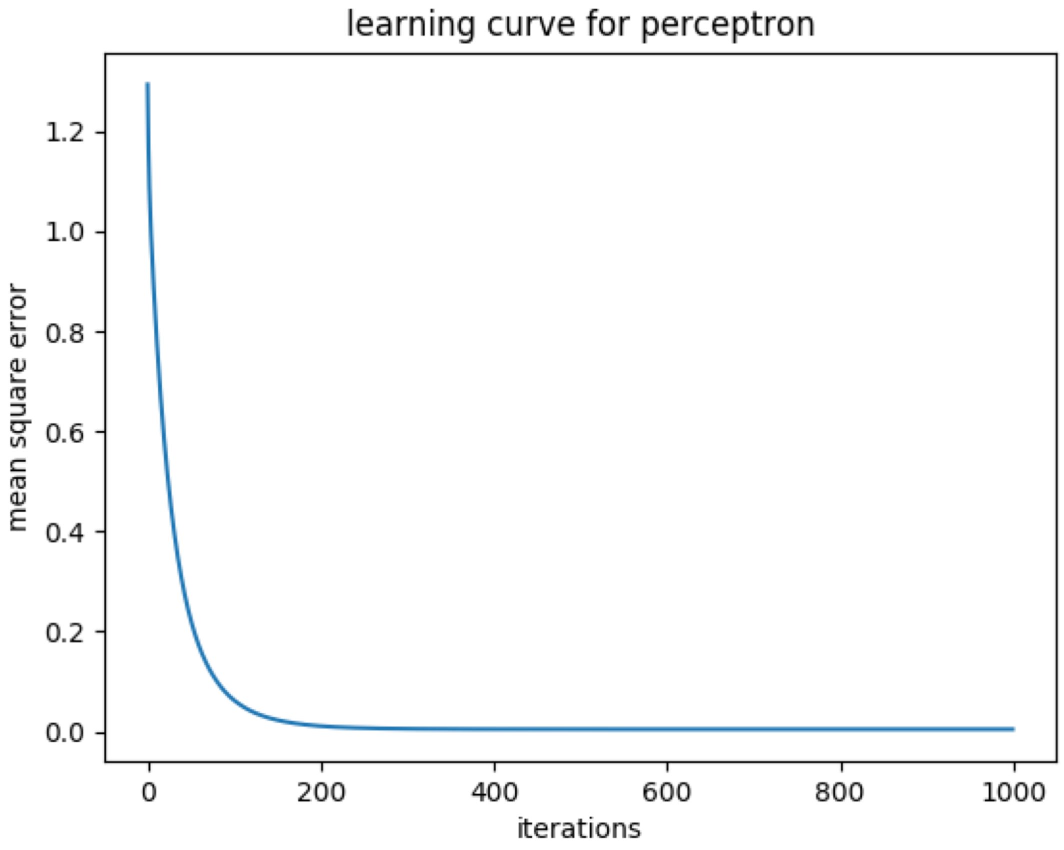
For the perceptron, activation function:

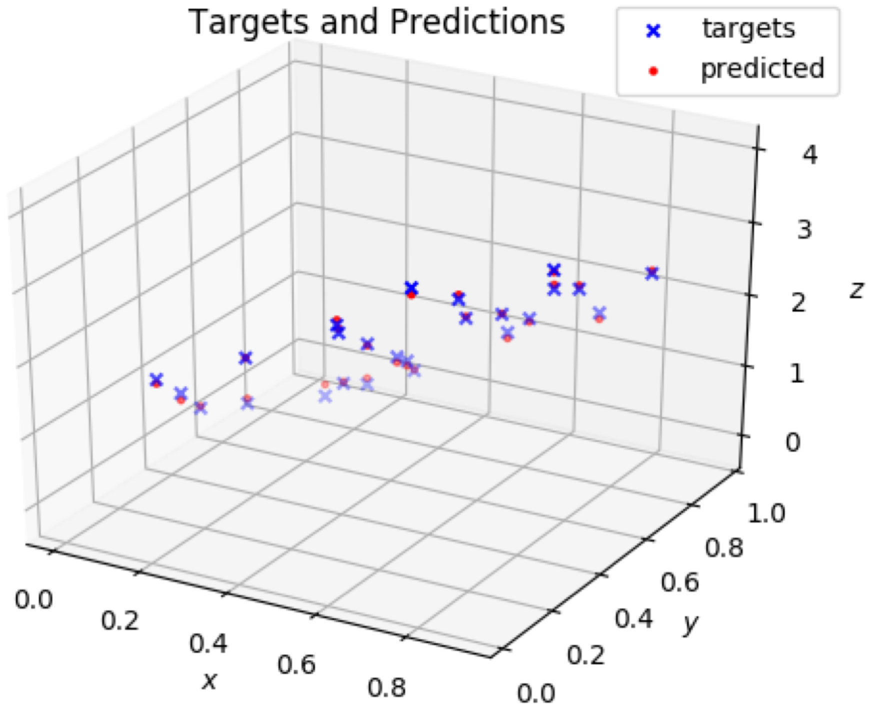
$$f(u) = \frac{5.8}{1 + e^{-u}} - 1.0$$

Gradient descent for perceptron

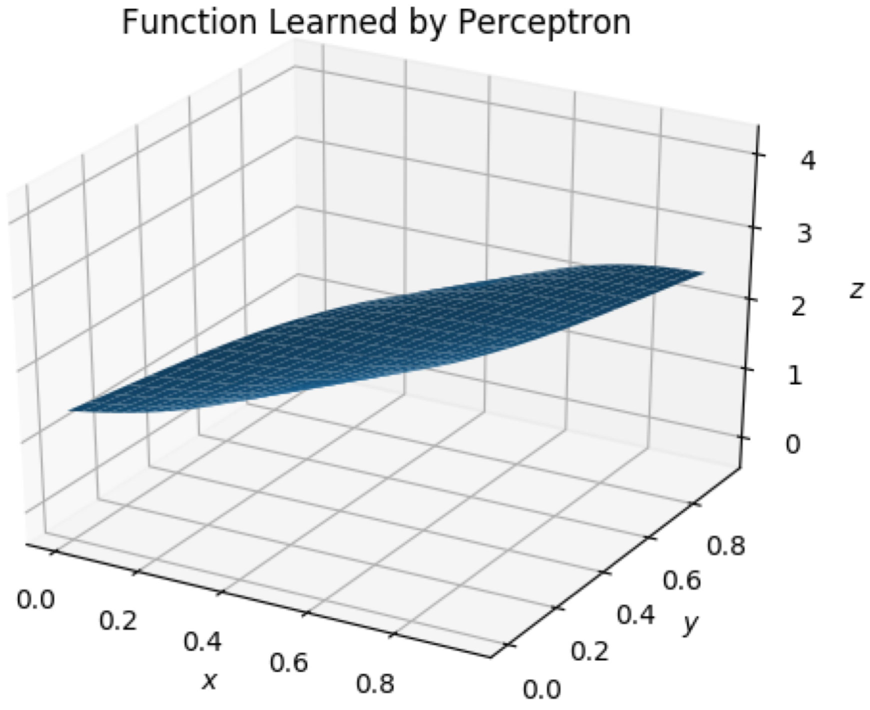
Given a training dataset$(\boldsymbol{X}, \boldsymbol{d})$
Set learning parameter α
Initialize $\boldsymbol{w}$ and b
Repeat until convergence:

$$\boldsymbol{u} = \boldsymbol{X}\boldsymbol{w} + b\mathbf{1}_P$$
$$\boldsymbol{y} = f(\boldsymbol{u})$$
$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha\boldsymbol{X}^T(\boldsymbol{d} - \boldsymbol{y}) \cdot f'(\boldsymbol{u})$$
$$b \leftarrow b + \alpha\mathbf{1}_P{}^T(\boldsymbol{d} - \boldsymbol{y}) \cdot f'(\boldsymbol{u})$$


learning curve for perceptron
mean square error
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
200
400
600
800
1000
iterations



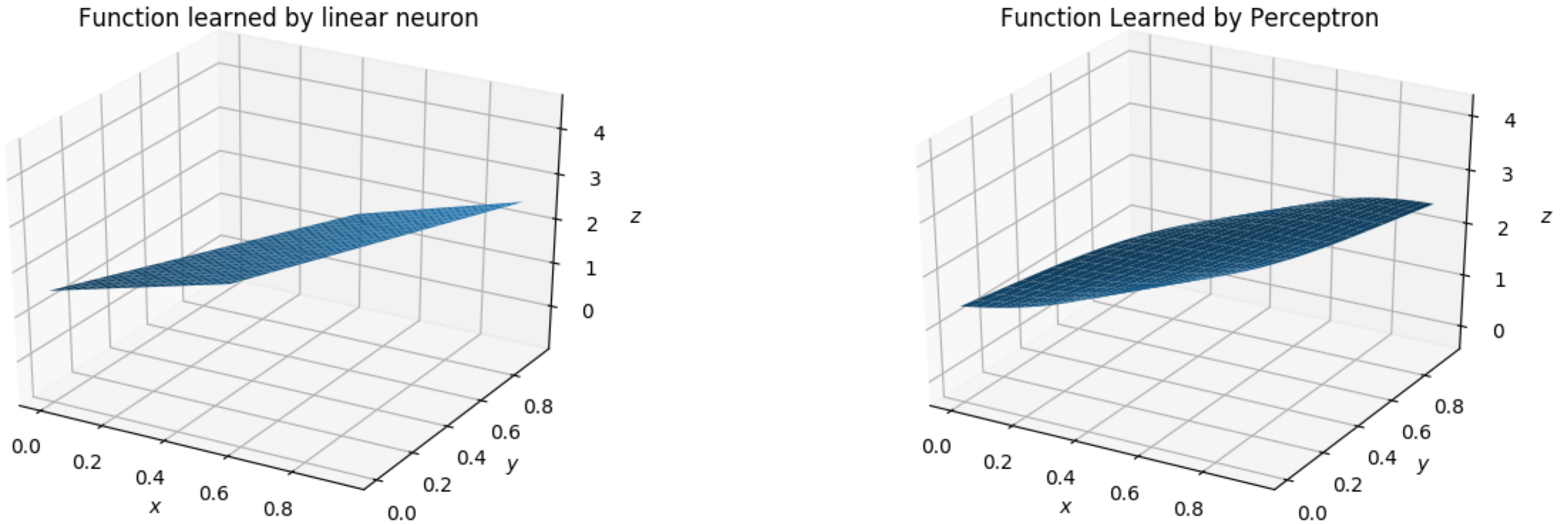
Targets and Predictions
targets
predicted
x
y
z
0.0
0.2
0.4
0.6
0.8
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2
3
4



Function Learned by Perceptron
0.0
0.2
0.4
0.6
0.8
x
0.0
0.2
0.4
0.6
0.8
y
0
1
2
3
4
z

$z = 1.5 + 3.3x - 2.5y + 0.2xy$ for $0 \leq x, y \leq 1$

	Linear Neuron	Perceptron
$\boldsymbol{w}$	$\begin{pmatrix} 3.25 \\ -2.55 \\ 0.29 \end{pmatrix}$	$\begin{pmatrix} 2.537 \\ -1.801 \end{pmatrix}$
b	1.53	-0.354
m.s.e.	3.97 x 10^{-5}	0.004
function	$z = 3.25x - 2.55y + 0.29xy + 1.53$	$z = \dfrac{5.8}{1 + e^{0.354 - 2.537x + 1.801y}} - 1.0$


Function learned by linear neuron
Function Learned by Perceptron
x
y
z
0.0
0.2
0.4
0.6
0.8
0
1
2
3
4