Neuron Layers

CE/CZ4042 – Tutorial 4

1. Design a softmax layer of neurons to perform the following classification, given the inputs $x = (x_1, x_2)$ and target class labels d:

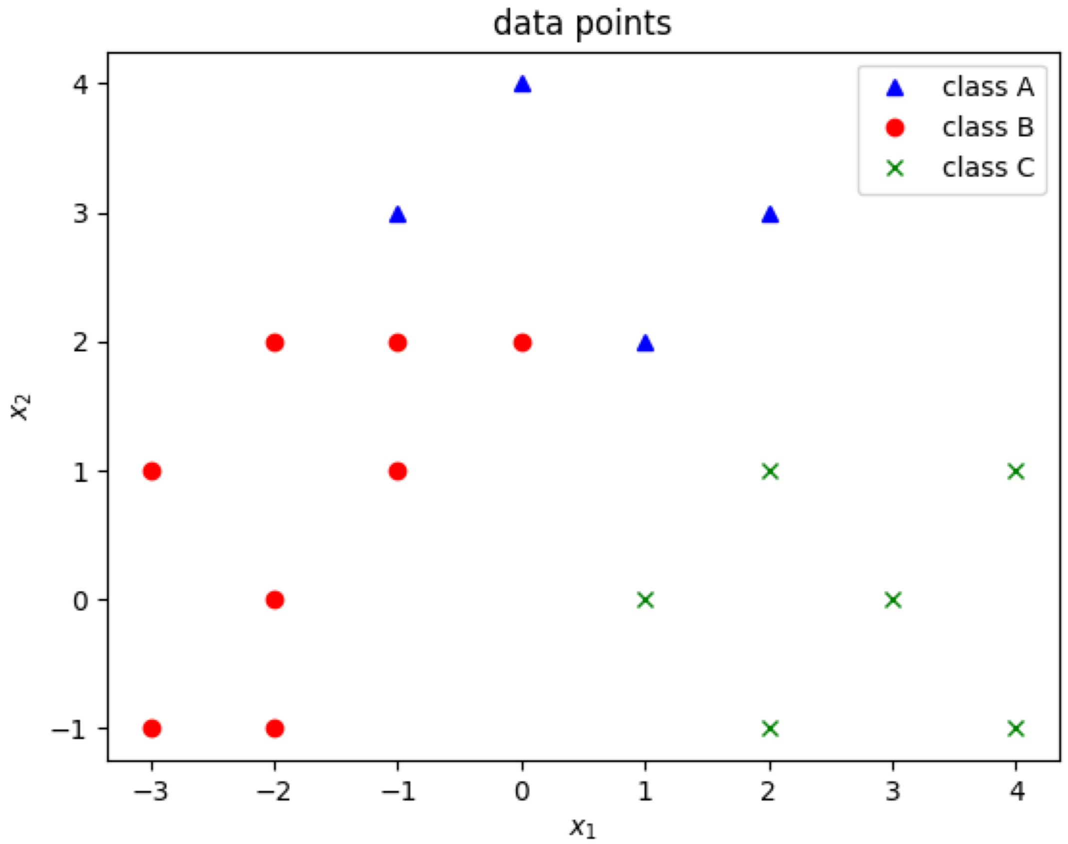
(x_1, x_2)	(0 4)	(-1 3)	(2 3)	(-2 2)	(0 2)	(1 2)	(-1 2)	(-3 1)	(-1 1)
d	A	A	A	B	B	A	B	B	B

(x_1, x_2)	(2 1)	(4 1)	(-2 0)	(1 0)	(3 0)	(-3 -1)	(-2 -1)	(2 -1)	(4 -1)
d	C	C	B	C	C	B	B	C	C

(a) Show one iteration of gradient descent learning at a learning factor 0.05.

(b) Find the weights and biases at convergence of learning

(c) Indicate the probabilities that the network predicts the classes of trained patterns.



data points
class A
class B
class C
x_1
x_2
−3
−2
−1
0
1
2
3
4
−1
0
1
2
3
4

GD for Softmax layer

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

Set learning rate α

Initialize $\boldsymbol{W}$ and $\boldsymbol{b}$

Iterate until convergence:

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

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

$$\nabla_U J = -\left(\boldsymbol{K} - f(\boldsymbol{U})\right)$$

$$\boldsymbol{W} \leftarrow \boldsymbol{W} + \alpha \boldsymbol{X}^T \nabla_U J$$

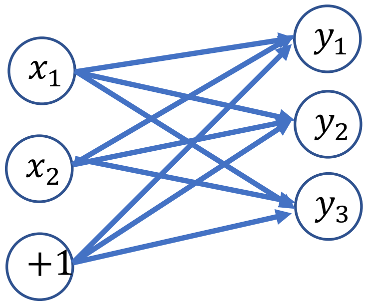
$$\boldsymbol{b} \leftarrow \boldsymbol{b} + \alpha (\nabla_U J)^T \boldsymbol{1}_P$$

Labels for classes:

$$Class\ A \rightarrow 0, Class\ B \rightarrow 1, Class\ C \rightarrow 2$$

The data matrix and target vector:

$$\boldsymbol{X} = \begin{pmatrix} 0 & 4 \\ -1 & 3 \\ 2 & 3 \\ -2 & 2 \\ \vdots & \vdots \\ 4 & -1 \end{pmatrix}, \quad \boldsymbol{d} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ \vdots \\ 2 \end{pmatrix}, \quad \boldsymbol{K} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 1 \end{pmatrix}$$



$$f(\boldsymbol{U}) = \frac{e^{\boldsymbol{U}}}{\sum_{k=1}^{K} e^{\boldsymbol{U_k}}}$$
$$\boldsymbol{Y} = \underset{k}{\operatorname{argmax}} f(\boldsymbol{U})$$

Learning rate $\alpha = 0.05$.

Initialize

Weights using truncated normal distribution with mean = 0 and s.d. = $1/\sqrt{2}$

Biases to zero.

$$\boldsymbol{W} = \begin{pmatrix} 0.88 & 0.08 & -0.34 \\ 0.68 & -0.39 & -0.19 \end{pmatrix} \text{ and } \boldsymbol{b} = \begin{pmatrix} 0.0 \\ 0.0 \\ 0.0 \end{pmatrix}$$

Iteration 1:

$$\boldsymbol{U} = \boldsymbol{XW} + \boldsymbol{B} = \begin{pmatrix} 0 & 4 \\ -1 & 3 \\ 2 & 3 \\ -2 & 2 \\ \vdots & \vdots \\ 4 & -1 \end{pmatrix} \begin{pmatrix} 0.88 & 0.08 & -0.34 \\ 0.68 & -0.39 & -0.19 \end{pmatrix} + \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ \vdots & \vdots & \vdots \\ 0.0 & 0.0 & 0.0 \end{pmatrix} = \begin{pmatrix} 2.72 & -1.54 & -0.75 \\ 1.17 & -1.23 & -0.23 \\ 3.8 & -1.0 & -1.23 \\ -0.39 & -0.93 & 0.30 \\ \vdots & \vdots & \vdots \\ 2.82 & 0.71 & -1.16 \end{pmatrix}$$

$$
\boldsymbol{U} = \begin{pmatrix} 2.72 & -1.54 & -0.75 \\ 1.17 & -1.23 & -0.23 \\ 3.8 & -1.0 & -1.23 \\ -0.39 & -0.93 & 0.30 \\ \vdots & \vdots & \vdots \\ 2.82 & 0.71 & -1.16 \end{pmatrix}
$$

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

$$
f(\boldsymbol{U}) == \begin{pmatrix} \frac{e^{2.72}}{e^{2.72}+e^{-1.54}+e^{-0.75}} & \frac{e^{-1.54}}{e^{2.72}+e^{-1.54}+e^{-0.75}} & \frac{e^{-0.75}}{e^{2.72}+e^{-1.54}+e^{-0.75}} \\ \frac{e^{1.17}}{e^{1.17}+e^{-1.23}+e^{-0.23}} & \frac{e^{-1.23}}{e^{1.17}+e^{-1.23}+e^{-0.23}} & \frac{e^{-0.23}}{e^{1.17}+e^{-1.23}+e^{-0.23}} \\ \vdots & \vdots & \vdots \\ \frac{e^{2.82}}{e^{2.82}+e^{0.71}+e^{-1.16}} & \frac{e^{0.71}}{e^{2.82}+e^{0.71}+e^{-1.16}} & \frac{e^{-1.16}}{e^{2.82}+e^{0.71}+e^{-1.16}} \end{pmatrix} = \begin{pmatrix} 0.96 & 0.01 & 0.03 \\ 0.75 & 0.07 & 0.18 \\ \vdots & \vdots & \vdots \\ 0.88 & 0.11 & 0.02 \end{pmatrix}
$$

$$f(\boldsymbol{U}) = \begin{pmatrix} 0.96 & 0.01 & 0.03 \\ 0.75 & 0.07 & 0.18 \\ \vdots & \vdots & \vdots \\ 0.88 & 0.11 & 0.02 \end{pmatrix}$$

$$\boldsymbol{y} = \underset{k}{\operatorname{argmax}} f(\boldsymbol{U}) = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

$$Cost = -\sum_{p=1}^{P} log\left(f\left(u_{pd_p}\right)\right) = -\log(0.96) - \log(0.75) \cdots - \log(0.02) = 34.36$$

$$Classification\ error = \sum_{p=1}^{P} 1(\boldsymbol{y} \neq \boldsymbol{d}) = 14$$

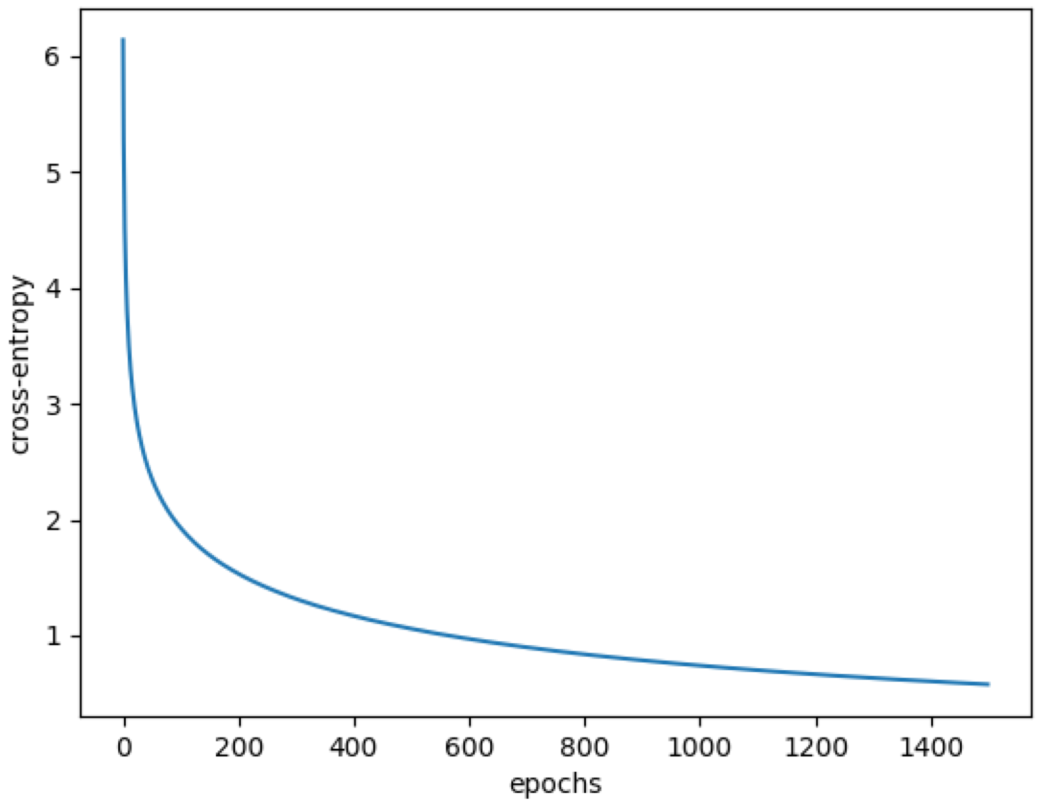
$$\nabla_U J = -\left(\boldsymbol{K} - f(\boldsymbol{U})\right) = -\left(\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 0.96 & 0.01 & 0.03 \\ 0.75 & 0.07 & 0.18 \\ \vdots & \vdots & \vdots \\ 0.88 & 0.11 & 0.02 \end{pmatrix}\right) = \begin{pmatrix} -0.04 & 0.01 & 0.03 \\ -0.25 & 0.07 & 0.18 \\ \vdots & \vdots & \vdots \\ 0.88 & 0.11 & -0.98 \end{pmatrix}$$

$$\boldsymbol{W} \leftarrow \boldsymbol{W} - \alpha \boldsymbol{X}^T \nabla_U J$$

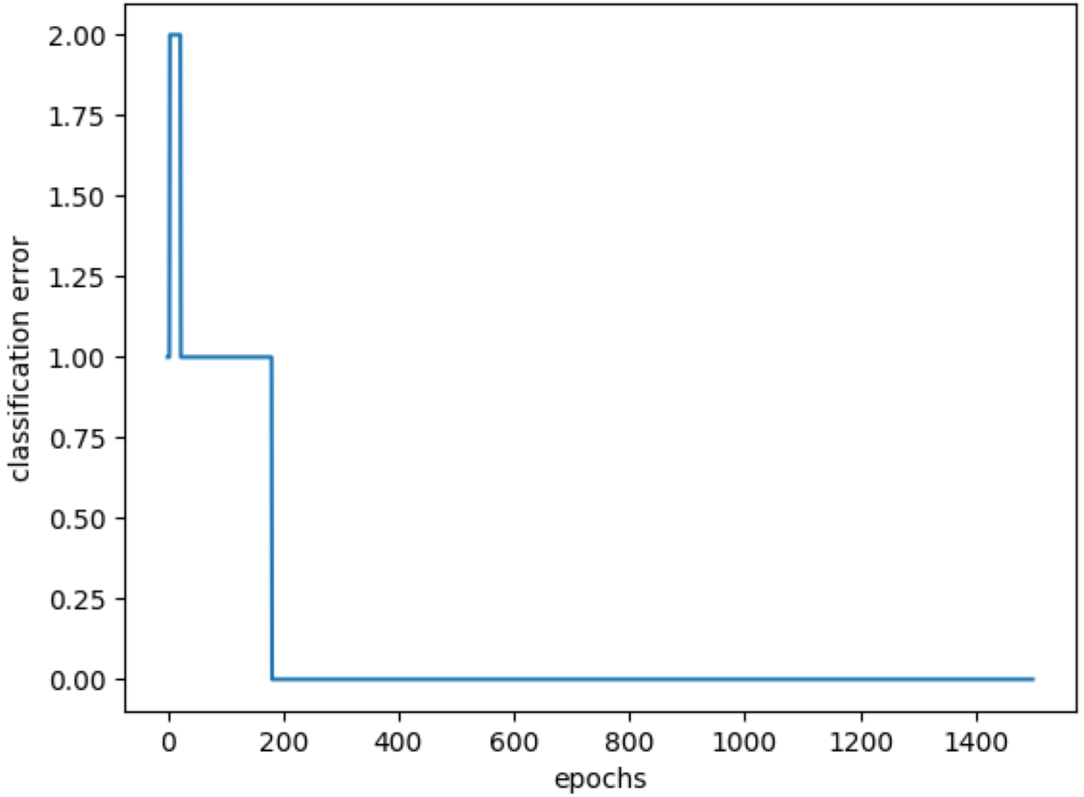
$$= \begin{pmatrix} 0.88 & 0.08 & -0.34 \\ 0.68 & -0.39 & -0.19 \end{pmatrix} + 0.05 \begin{pmatrix} 0 & -1 & \cdots & 4 \\ 4 & 3 & \cdots & -1 \end{pmatrix} \begin{pmatrix} 0.96 & 0.01 & 0.03 \\ 0.75 & 0.07 & 0.18 \\ \vdots & \vdots & \vdots \\ 0.88 & 0.11 & 0.02 \end{pmatrix}$$

$$= \begin{pmatrix} 0.28 & -0.54 & 0.89 \\ 0.54 & -0.12 & -0.31 \end{pmatrix}$$

$$\boldsymbol{b} \leftarrow \boldsymbol{b} - \alpha(\nabla_U J)^T \mathbf{1}_P = \begin{pmatrix} 0.0 \\ 0.0 \\ 0.0 \end{pmatrix} + 0.05 \begin{pmatrix} -0.04 & 0.01 & 0.03 \\ -0.25 & 0.07 & 0.18 \\ \vdots & \vdots & \vdots \\ 0.88 & 0.11 & -0.98 \end{pmatrix}^T \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} -0.32 \\ 0.27 \\ 0.06 \end{pmatrix}$$



cross-entropy
6
5
4
3
2
1
0
200
400
600
800
1000
1200
1400
epochs



2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
classification error
0
200
400
600
800
1000
1200
1400
epochs

At convergence:

$$\boldsymbol{W} = \begin{pmatrix} -0.15 & -3.41 & 4.18 \\ 5.27 & -1.02 & -4.15 \end{pmatrix} \text{ and } \boldsymbol{b} = \begin{pmatrix} -7.82 \\ 5.81 \\ 2.02 \end{pmatrix}$$

Entropy = 0.58

Error = 0

$$
\boldsymbol{X}=\begin{pmatrix}-1 & 2\\ 0 & 4\\ -1 & 3\\ 0 & 2\\ 3 & 0\\ -2 & -1\\ 4 & 1\\ 1 & 2\\ 2 & -1\\ 2 & 3\\ 2 & 1\\ -2 & 0\\ -3 & -1\\ 1 & 0\\ -1 & 1\\ 4 & -1\\ -3 & 1\\ -2 & 2\end{pmatrix}\quad f(\boldsymbol{U})=\begin{pmatrix}\color{red}{1.0} & 0.0 & 0.0\\ \color{red}{0.88} & 0.12 & 0.0\\ \color{red}{1.0} & 0.0 & 0.0\\ 0.0 & \color{red}{1.0} & 0.0\\ 0.26 & \color{red}{0.74} & 1.0\\ \color{red}{0.89} & 0.1 & 0.0\\ 0.01 & \color{red}{0.99} & 0.0\\ 0.0 & \color{red}{1.0} & 0.0\\ 0.0 & \color{red}{1.0} & 0.0\\ 0.0 & 0.0 & \color{red}{1.0}\\ 0.0 & 0.0 & \color{red}{1.0}\\ 0.0 & \color{red}{1.0} & 0.0\\ 0.0 & 0.02 & \color{red}{0.98}\\ 0.0 & 0.0 & \color{red}{1.0}\\ 0.0 & \color{red}{1.0} & 0.0\\ 0.0 & \color{red}{1.0} & 0.0\\ 0.0 & 0.0 & \color{red}{1.0}\\ 0.0 & 0.0 & \color{red}{1.0}\end{pmatrix},\boldsymbol{Y}=\begin{pmatrix}0\\0\\0\\1\\1\\0\\1\\1\\1\\2\\2\\1\\2\\2\\1\\1\\2\\2\end{pmatrix}
$$

Probabilities of input patterns, belonging to target classes are given in RED.

2. Use mini-batch gradient decent learning to train a softmax layer to classify Iris dataset (https://archive.ics.uci.edu/ml/datasets/Iris). The dataset contains 150 data points. Use 120 data points for training the classifier and test on the remaining 30 data points.
Set learning rate = 0.01 and batch size = 16.

You can use the following python commands to load Iris data:

```
from sklearn import datasets
iris = datasets.load_iris()
```

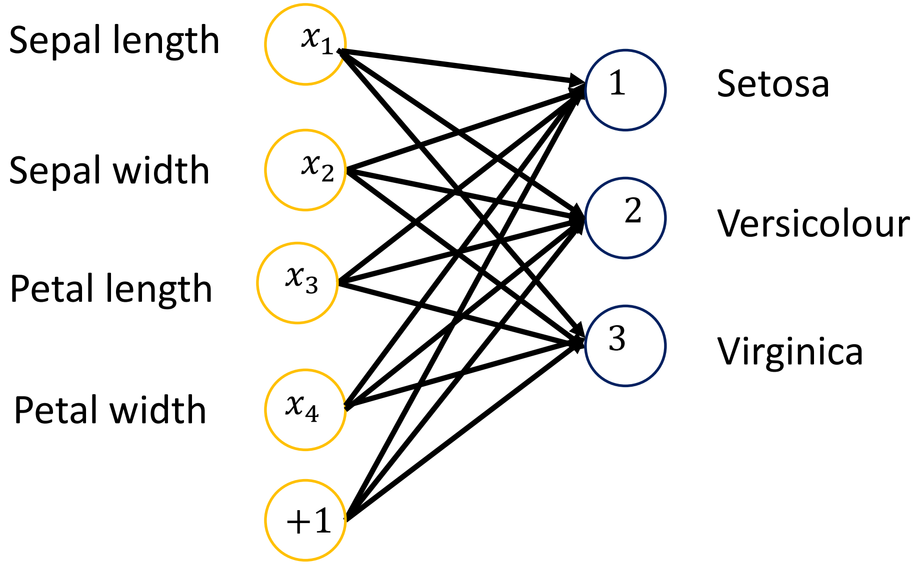
Repeat the classification with batch sizes = 2, 4, 8, 16, 24, 32, 48, and 64, and compare the accuracies and times taken for a weight update.

Iris dataset:
https://archive.ics.uci.edu/ml/datasets/Iris

Four features:
- Sepal length
- sepal width
- petal length
- petal width

Three classes:
- Iris Setosa
- Iris Versicolour
- Iris Virginica

150 data points, 50 for each class



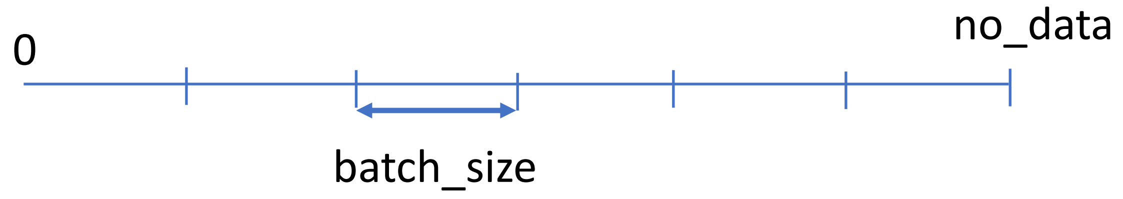
120 data points for training and **30** data points for testing

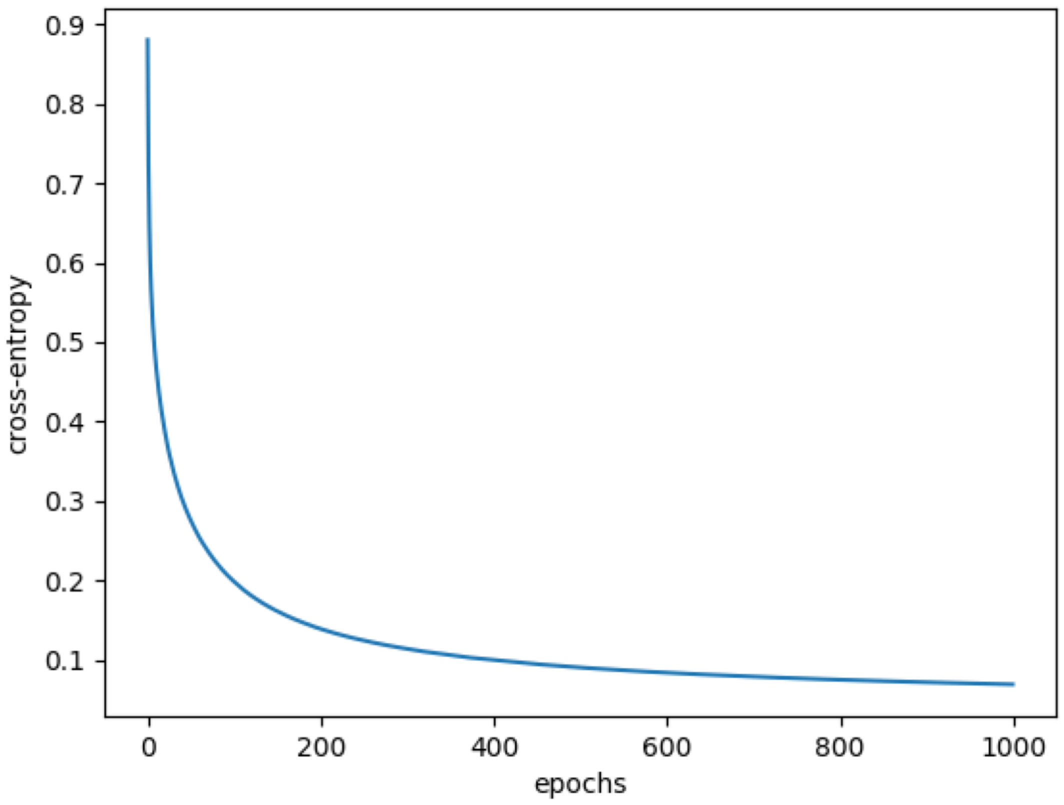
Mini-batch gradient descent

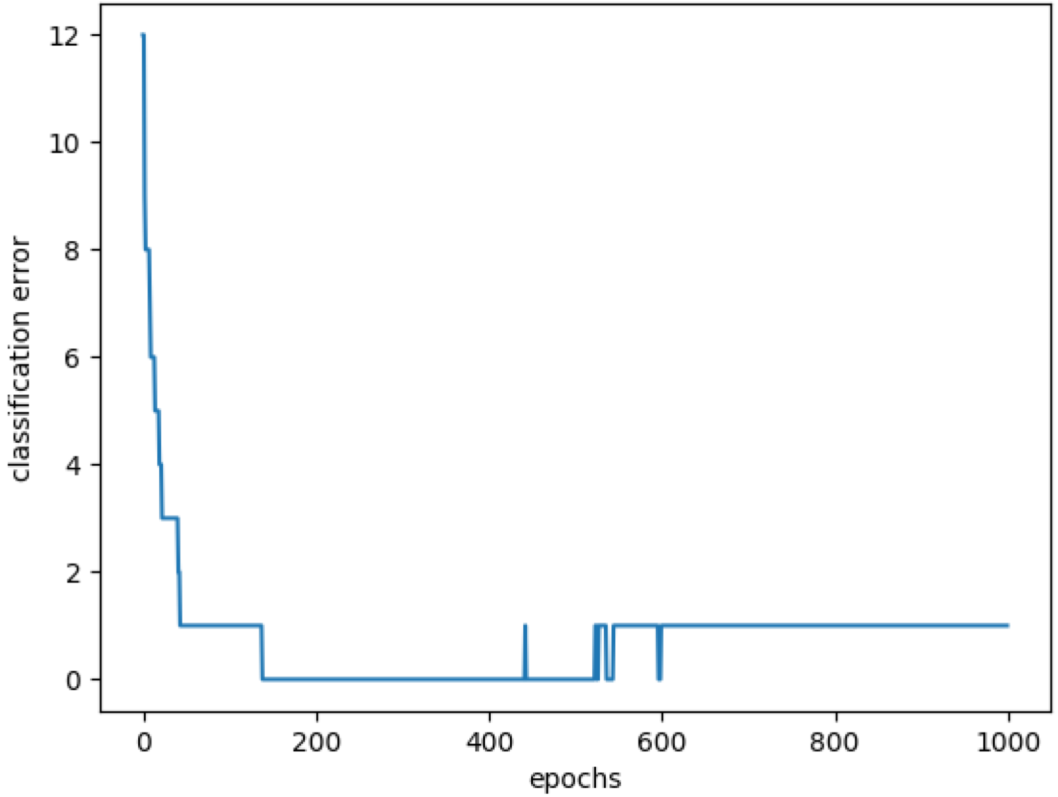
Batch size = 16, learning factor = 0.01

Implementing mini-batch training

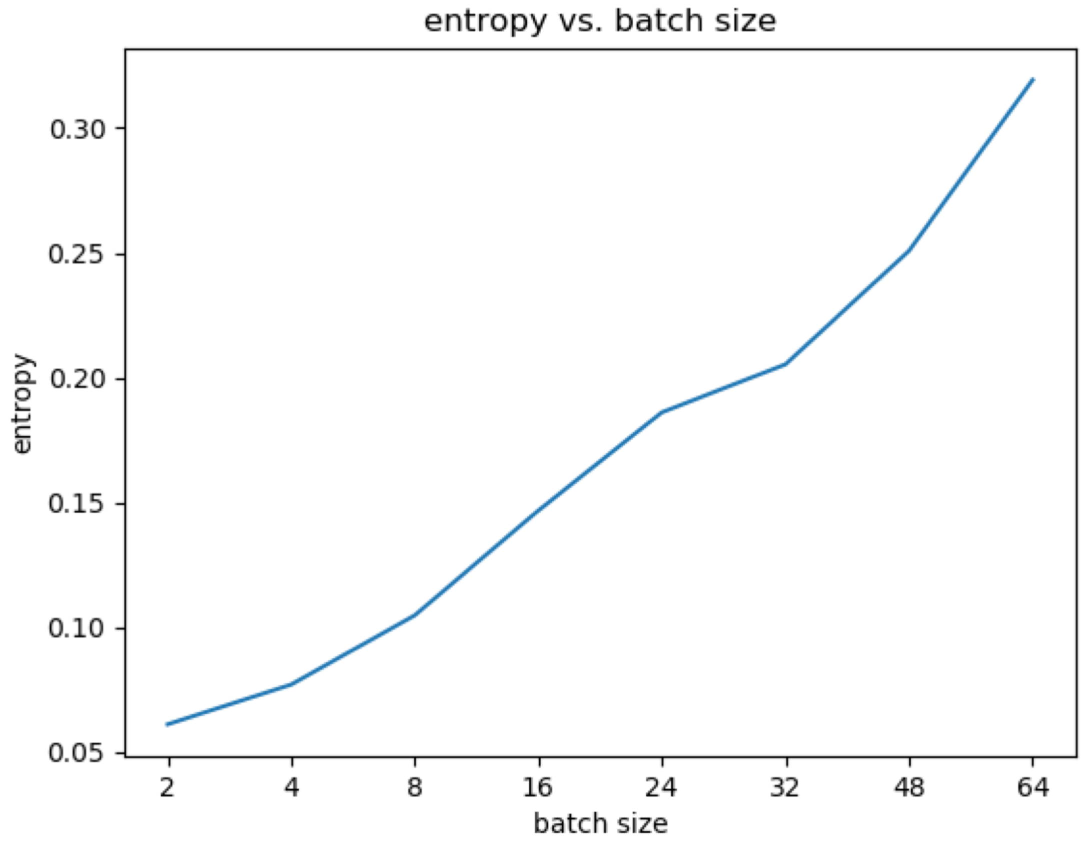
```
idx = np.arange(no_data)

for i in range(no_epochs):

    np.random.shuffle(idx)
    train_X, train_Y = trainX[idx], trainY[idx]

    for start, end in zip(range(0, no_data, batch_size), range(batch_size, no_data, batch_size)):
        train.run(feed_dict={x: train_X[start:end], y_: train_Y[start:end]})
```

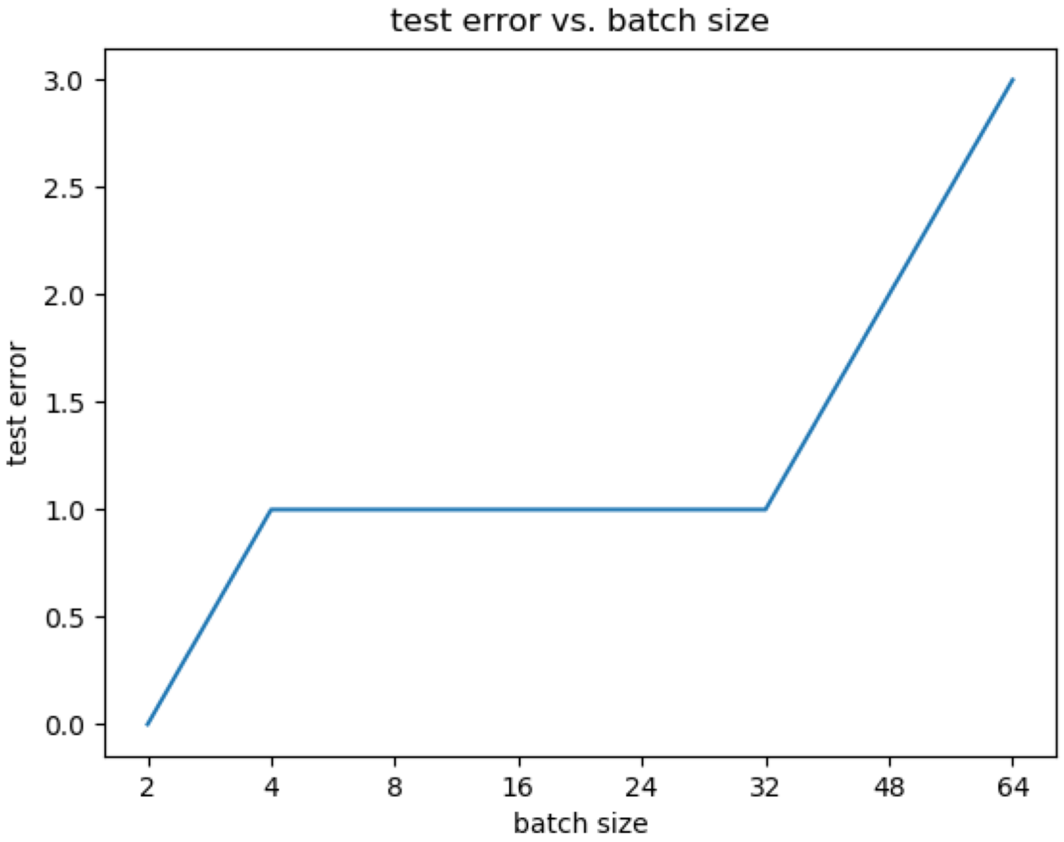


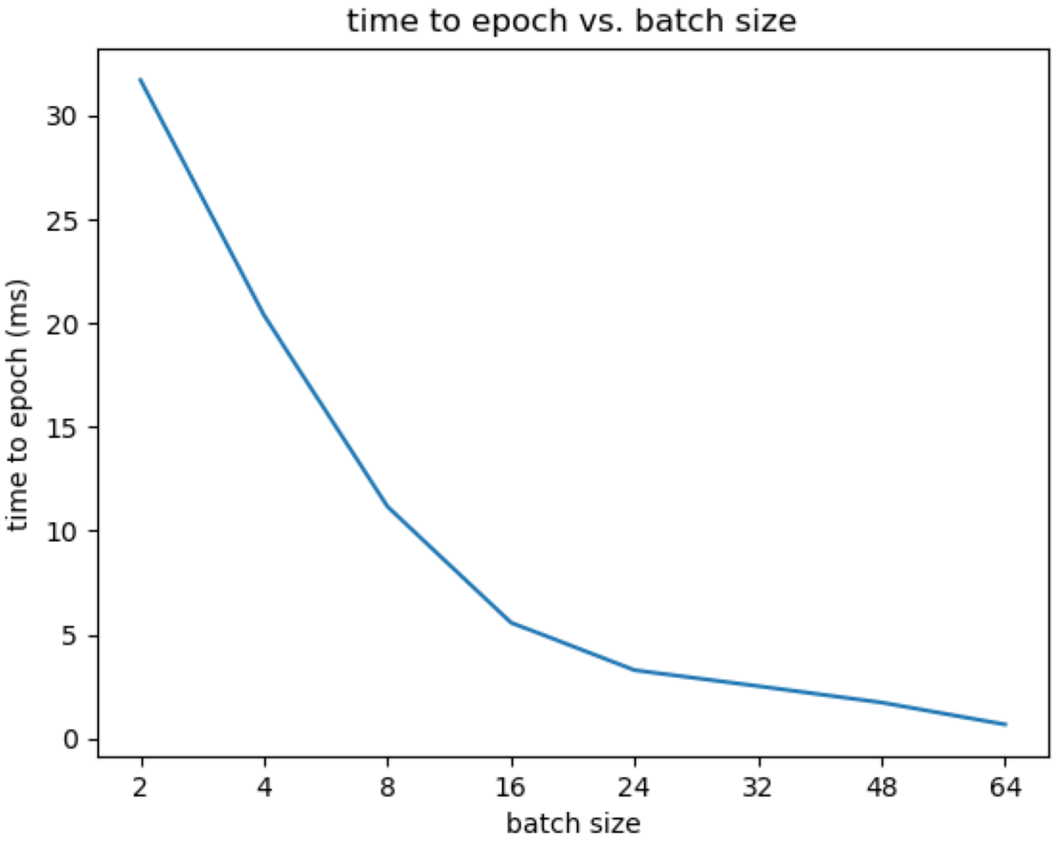

0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
cross-entropy
0
200
400
600
800
1000
epochs



classification error
12
10
8
6
4
2
0
0
200
400
600
800
1000
epochs


entropy vs. batch size
entropy
0.30
0.25
0.20
0.15
0.10
0.05
2
4
8
16
24
32
48
64
batch size


test error vs. batch size
3.0
2.5
2.0
1.5
1.0
0.5
0.0
test error
2
4
8
16
24
32
48
64
batch size



Optimal batch size is a compromise between the accuracy and running time.

3. Design a perceptron layer to perform the following mapping:

Inputs	Outputs
$(0.50 \quad 0.23)$	$(0.16 \quad 0.74)$
$(0.20 \quad 0.76)$	$(0.49 \quad 0.97)$
$(0.17 \quad 0.09)$	$(0.01 \quad 0.26)$
$(0.69 \quad 0.95)$	$(1.19 \quad 1.70)$
$(0.00 \quad 0.51)$	$(0.13 \quad 0.52)$
$(0.81 \quad 0.61)$	$(0.77 \quad 1.48)$
$(0.72 \quad 0.29)$	$(0.40 \quad 1.04)$
$(0.92 \quad 0.72)$	$(1.14 \quad 1.7)$

Train the perceptron layer with (a) GD and (b) SGD. Show one iteration of learning and plot learning curves and predicted and target outputs. Set the learning factor $\alpha = 0.05$.

Inputs	Outputs
$(0.50 \quad 0.23)$	$(0.16 \quad 0.74)$
$(0.20 \quad 0.76)$	$(0.49 \quad 0.97)$
$(0.17 \quad 0.09)$	$(0.01 \quad 0.26)$
$(0.69 \quad 0.95)$	$(1.19 \quad 1.70)$
$(0.00 \quad 0.51)$	$(0.13 \quad 0.52)$
$(0.81 \quad 0.61)$	$(0.77 \quad 1.48)$
$(0.72 \quad 0.29)$	$(0.40 \quad 1.04)$
$(0.92 \quad 0.72)$	$(1.14 \quad 1.7)$

Let $\boldsymbol{y} = (y_1, y_2)$.

$$y_1, y_2 \in [0.0, 2.0]$$

Activation function $y = f(u) = \frac{2}{1+e^{-u}} = 2f_0(u)$

where $f_0(u)$ is the sigmoid function.

GD for a perceptron layer:

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

Set learning parameter α

Initialize $\boldsymbol{W}$ and $\boldsymbol{b}$

Repeat until convergence:

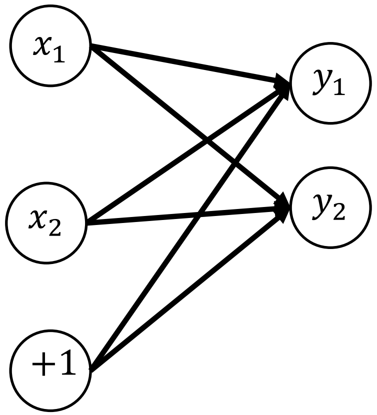
$$\boldsymbol{U} = \boldsymbol{X}\boldsymbol{W} + \boldsymbol{B}$$

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

$$\nabla_{U} J = -(\boldsymbol{D} - \boldsymbol{Y}) \cdot f'(\boldsymbol{U})$$

$$\boldsymbol{W} \leftarrow \boldsymbol{W} - \alpha \boldsymbol{X}^{T} \nabla_{U} J$$

$$\boldsymbol{b} \leftarrow \boldsymbol{b} - \alpha (\nabla_{U} J)^{T} \mathbf{1}_{P}$$



Initialize:

Weights using truncated normal distribution: $\boldsymbol{W} = \begin{pmatrix} 1.24 & 0.11 \\ -0.48 & 0.97 \end{pmatrix}$

Biases to zero $\boldsymbol{b} = \begin{pmatrix} 0.0 \\ 0.0 \end{pmatrix}$

$\alpha = 0.05$

Activation function $\boldsymbol{y} = f(\boldsymbol{u}) = \frac{2}{1+e^{-u}} = 2f_0(\boldsymbol{u})$
where $f_0(\boldsymbol{u})$ is the sigmoid function.

$$f'(\boldsymbol{u}) = 2f_0'(\boldsymbol{u}) = 2f_0(\boldsymbol{u})\big(1 - f_0(\boldsymbol{u})\big) = \boldsymbol{y}\left(1 - \frac{\boldsymbol{y}}{2}\right)$$

Iteration 1:

$$\text{Input } \boldsymbol{X} = \begin{pmatrix} 0.5 & 0.23 \\ 0.2 & 0.76 \\ 0.17 & 0.09 \\ 0.69 & 0.95 \\ 0,0 & 0.51 \\ 0.81 & 0.61 \\ 0.72 & 0.29 \\ 0.92 & 0.72 \end{pmatrix}, \text{ Targets } \boldsymbol{D} = \begin{pmatrix} 0.16 & 0.74 \\ 0.49 & 0.97 \\ 0.01 & 0.26 \\ 1.19 & 1.7 \\ 0.13 & 0.52 \\ 0.77 & 1.48 \\ 0.4 & 1.04 \\ 1.14 & 1.7 \end{pmatrix}$$

$$\boldsymbol{U} = \boldsymbol{X}\boldsymbol{W} + \boldsymbol{B} = \begin{pmatrix} 0.5 & 0.23 \\ 0.2 & 0.76 \\ 0.17 & 0.09 \\ 0.69 & 0.95 \\ 0,0 & 0.51 \\ 0.81 & 0.61 \\ 0.72 & 0.29 \\ 0.92 & 0.72 \end{pmatrix} \begin{pmatrix} 1.24 & 0.11 \\ -0.48 & 0.97 \end{pmatrix} + \begin{pmatrix} 0.0 & 0.0 \\ 0.0 & 0.0 \\ 0.0 & 0.0 \\ 0.0 & 0.0 \\ 0.0 & 0.0 \\ 0.0 & 0.0 \\ 0.0 & 0.0 \\ 0.0 & 0.0 \end{pmatrix} = \begin{pmatrix} 0.51 & 0.28 \\ -0.11 & 0.76 \\ 0.17 & 0.11 \\ 0.40 & 1.0 \\ -0.24 & 0.49 \\ 0.71 & 0.68 \\ 0.75 & 0.36 \\ 0.80 & 0.80 \end{pmatrix}$$

$$\boldsymbol{Y} = f(\boldsymbol{U}) = \frac{2}{1+e^{-\boldsymbol{U}}} = \begin{pmatrix} 1.25 & 1.14 \\ 0.94 & 1.36 \\ 1.08 & 1.05 \\ 1.20 & 1.46 \\ 0.88 & 1.24 \\ 1.34 & 1.33 \\ 1.36 & 1.18 \\ 1.38 & 1.38 \end{pmatrix},$$

Cost = $J(\boldsymbol{W}, \boldsymbol{b}\,) = \frac{1}{8}\sum_{p=1}^{8}\sum_{k=1}^{2}\left(d_{pk} - y_{pk}\right)^2 = 0.76$

$$f'(\boldsymbol{U}) = \boldsymbol{Y} \cdot \left(1 - \frac{\boldsymbol{Y}}{2}\right) = \begin{pmatrix} 0.47 & 0.49 \\ 0.50 & 0.43 \\ 0.50 & 0.50 \\ 0.48 & 0.39 \\ 0.49 & 0.47 \\ 0.44 & 0.45 \\ 0.44 & 0.48 \\ 0.43 & 0.43 \end{pmatrix}$$

$$\nabla_U J = -(\boldsymbol{D} - \boldsymbol{Y}) \cdot f'\,(\boldsymbol{U})$$

$$= -\left(\begin{pmatrix} 0.16 & 0.74 \\ 0.49 & 0.97 \\ 0.01 & 0.26 \\ 1.19 & 1.7 \\ 0.13 & 0.52 \\ 0.77 & 1.48 \\ 0.4 & 1.04 \\ 1.14 & 1.7 \end{pmatrix} - \begin{pmatrix} 1.25 & 1.14 \\ 0.94 & 1.36 \\ 1.08 & 1.05 \\ 1.20 & 1.46 \\ 0.88 & 1.24 \\ 1.34 & 1.33 \\ 1.36 & 1.18 \\ 1.38 & 1.38 \end{pmatrix}\right) \cdot \begin{pmatrix} 0.47 & 0.49 \\ 0.50 & 0.43 \\ 0.50 & 0.50 \\ 0.48 & 0.39 \\ 0.49 & 0.47 \\ 0.44 & 0.45 \\ 0.44 & 0.48 \\ 0.43 & 0.43 \end{pmatrix}$$

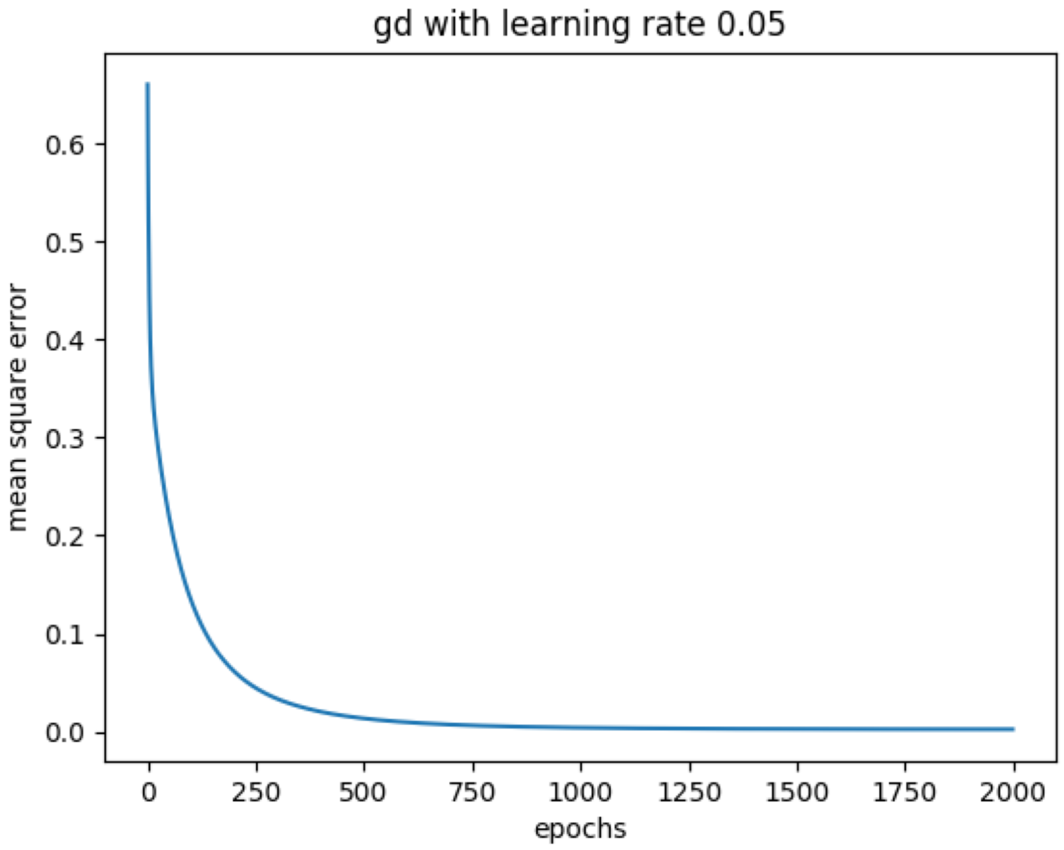
$$= \begin{pmatrix} 0.51 & 0.20 \\ 0.23 & 0.17 \\ 0.53 & 0.40 \\ 0.00 & -0.09 \\ 0.37 & 0.34 \\ 0.25 & -0.07 \\ 0.42 & 0.07 \\ 0.10 & -0.14 \end{pmatrix}$$

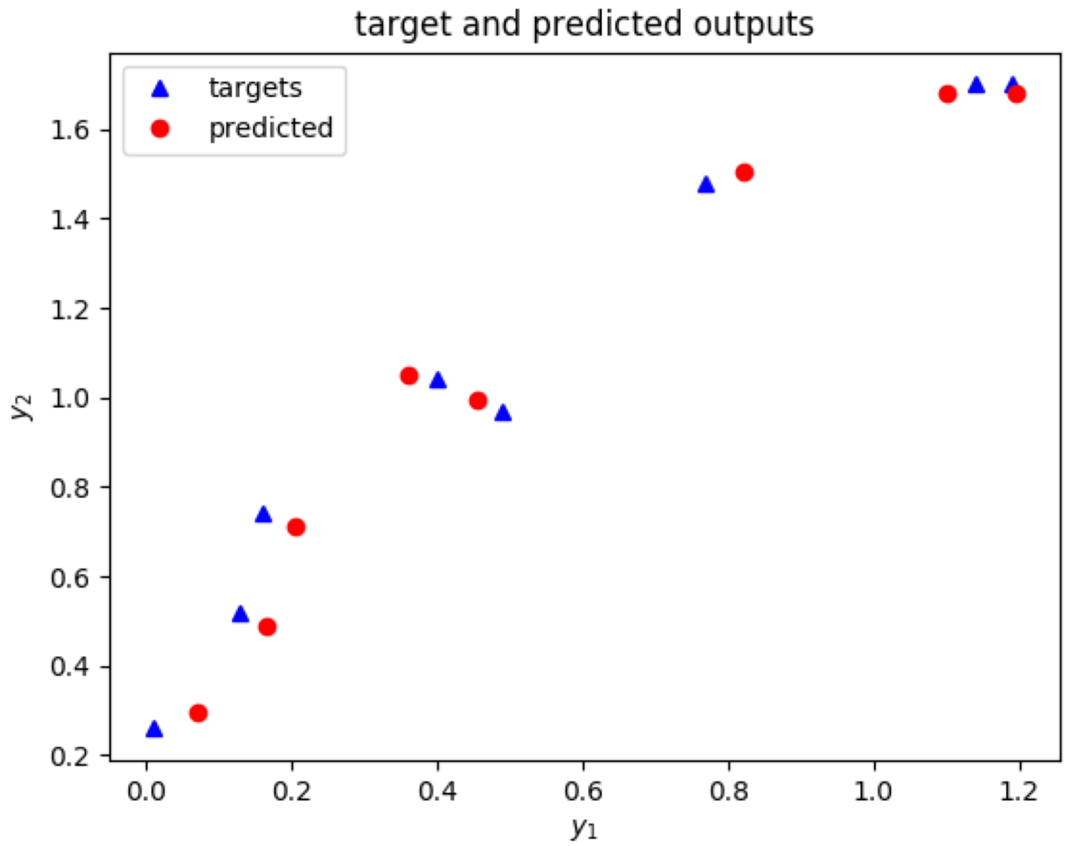
$$\boldsymbol{W} \leftarrow \boldsymbol{W} - \alpha \boldsymbol{X}^T \nabla_U J$$

$$\boldsymbol{W} = \begin{pmatrix} 1.24 & 0.11 \\ -0.48 & 0.97 \end{pmatrix} - 0.05 \begin{pmatrix} 0.5 & 0.2 & 0.17 & 0.69 & 0.0 & 0.81 & 0.72 & 0.92 \\ 0.23 & 0.76 & 0.09 & 0.95 & 0.51 & 0.61 & 0.29 & 0.72 \end{pmatrix} \begin{pmatrix} 0.51 & 0.20 \\ 0.23 & 0.17 \\ 0.53 & 0.40 \\ 0.00 & -0.09 \\ 0.37 & 0.34 \\ 0.25 & -0.07 \\ 0.42 & 0.07 \\ 0.10 & -0.14 \end{pmatrix} = \begin{pmatrix} 1.19 & 0.11 \\ -0.52 & 0.96 \end{pmatrix}$$

$$\boldsymbol{b} \leftarrow \boldsymbol{b} - \alpha (\nabla_U J)^T \mathbf{1}_P$$

$$\boldsymbol{b} = \begin{pmatrix} 0.0 \\ 0.0 \end{pmatrix} - 0.05 \begin{pmatrix} 0.51 & 0.23 & 0.53 & 0.0 & 0.37 & 0.25 & 0.42 & 0.10 \\ 0.20 & 0.17 & 0.40 & -0.09 & 0.34 & -0.07 & 0.07 & -0.14 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -0.12 \\ -0.04 \end{pmatrix}$$


gd with learning rate 0.05
mean square error
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
250
500
750
1000
1250
1500
1750
2000
epochs


target and predicted outputs
targets
predicted
y_1
y_2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6

SGD for a perceptron layer:

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

Set learning parameter α

Initialize $\boldsymbol{W}$ and $\boldsymbol{b}$

Repeat until convergence:

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

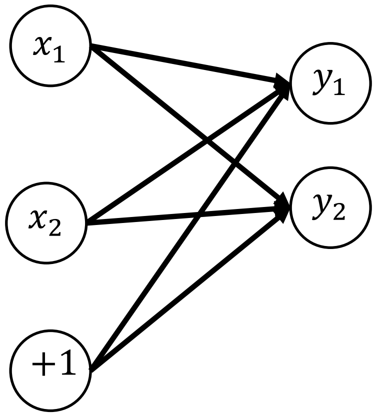
$$\boldsymbol{u} = \boldsymbol{W}^T \boldsymbol{x} + \boldsymbol{b}$$

$$\boldsymbol{y} = f(\boldsymbol{u})$$

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

$$\boldsymbol{W} \leftarrow \boldsymbol{W} - \alpha \boldsymbol{x} (\nabla_{\boldsymbol{u}} J)^T$$

$$\boldsymbol{b} \leftarrow \boldsymbol{b} - \alpha \nabla_{\boldsymbol{u}} J$$



Initialize:

Weights using truncated normal distribution: $\boldsymbol{W} = \begin{pmatrix} 1.24 & 0.11 \\ -0.48 & 0.97 \end{pmatrix}$

Biases to zero $\boldsymbol{b} = \begin{pmatrix} 0.0 \\ 0.0 \end{pmatrix}$

$\alpha = 0.05$

Activation function $y = f(u) = \frac{2}{1+e^{-u}} = 2f_0(u)$
where $f_0(u)$ is the sigmoid function.

$$f'(u) = 2f_0'(u) = 2f_0(u)\big(1 - f_0(u)\big) = y\left(1 - \frac{y}{2}\right)$$

Iteration 1:

Apply patterns one by one in random order:

Let first pattern, $\boldsymbol{x} = \begin{pmatrix} 0.17 \\ 0.09 \end{pmatrix}$, $\boldsymbol{d} = \begin{pmatrix} 0.01 \\ 0.26 \end{pmatrix}$

$$\boldsymbol{u} = \boldsymbol{W}^T \boldsymbol{x} + \boldsymbol{b} = \begin{pmatrix} 1.24 & -0.48 \\ 0.11 & 0.96 \end{pmatrix} \begin{pmatrix} 0.17 \\ 0.09 \end{pmatrix} + \begin{pmatrix} 0.0 \\ 0.0 \end{pmatrix} = \begin{pmatrix} 0.17 \\ 0.11 \end{pmatrix}$$

$$\boldsymbol{y} = f(\boldsymbol{u}) = \frac{2}{1 + e^{-\boldsymbol{u}}} = \begin{pmatrix} 1.08 \\ 1.05 \end{pmatrix}$$

$$Square\ error = 1.78$$

$$f'(\boldsymbol{u}) = \boldsymbol{y}(1 - \boldsymbol{y}/2) = \begin{pmatrix} 1.08 \\ 1.05 \end{pmatrix} . \left(\begin{pmatrix} 1.0 \\ 1.0 \end{pmatrix} - 0.5 \begin{pmatrix} 1.08 \\ 1.05 \end{pmatrix} \right) = \begin{pmatrix} 0.50 \\ 0.50 \end{pmatrix}$$

$$\nabla_{\boldsymbol{u}} J = -(\boldsymbol{d} - \boldsymbol{y}) \cdot f'(\boldsymbol{u}) = \left(\begin{pmatrix} 0.01 \\ 0.26 \end{pmatrix} - \begin{pmatrix} 1.08 \\ 1.05 \end{pmatrix} \right) \cdot \begin{pmatrix} 0.50 \\ 0.50 \end{pmatrix} = \begin{pmatrix} 0.53 \\ 0.40 \end{pmatrix}$$

$$\boldsymbol{W} = \boldsymbol{W} - \alpha \boldsymbol{x} (\nabla_{\boldsymbol{u}} J)^T = \begin{pmatrix} 1.24 & 0.11 \\ -0.48 & 0.97 \end{pmatrix} - 0.05 \begin{pmatrix} 0.17 \\ 0.09 \end{pmatrix} (0.53 \quad 0.40) = \begin{pmatrix} 1.23 & 0.11 \\ -0.48 & 0.96 \end{pmatrix}$$

$$\boldsymbol{b} = \boldsymbol{b} - \alpha \nabla_{\boldsymbol{u}} J = \begin{pmatrix} 0.0 \\ 0.0 \end{pmatrix} - 0.05 \begin{pmatrix} 0.53 \\ 0.40 \end{pmatrix} = \begin{pmatrix} -0.03 \\ -0.02 \end{pmatrix}$$

Apply second pattern, $\boldsymbol{x} = \begin{pmatrix} 0.69 \\ 0.95 \end{pmatrix}$, $\boldsymbol{d} = \begin{pmatrix} 1.19 \\ 1.7 \end{pmatrix}$

$$\boldsymbol{u} = \boldsymbol{W}^T\boldsymbol{x} + \boldsymbol{b} = \begin{pmatrix} 1.23 & -0.48 \\ 0.11 & 0.96 \end{pmatrix}\begin{pmatrix} 0.69 \\ 0.95 \end{pmatrix} + \begin{pmatrix} -0.03 \\ -0.02 \end{pmatrix} = \begin{pmatrix} 0.37 \\ 0.97 \end{pmatrix}$$

$$\boldsymbol{y} = f(\boldsymbol{u}) = \frac{2}{1 + e^{-\boldsymbol{u}}} = \begin{pmatrix} 1.18 \\ 1.45 \end{pmatrix}$$

$$Square\ error = \sum_{k=1}^{2}(d_k - y_k)^2 = 0.06$$

$$f'(\boldsymbol{u}) = \boldsymbol{y}(1 - \boldsymbol{y}/2) = \begin{pmatrix} 1.18 \\ 1.45 \end{pmatrix} \cdot \left(\begin{pmatrix} 1.0 \\ 1.0 \end{pmatrix} - 0.5\begin{pmatrix} 1.18 \\ 1.45 \end{pmatrix}\right) = \begin{pmatrix} 0.48 \\ 0.40 \end{pmatrix}$$

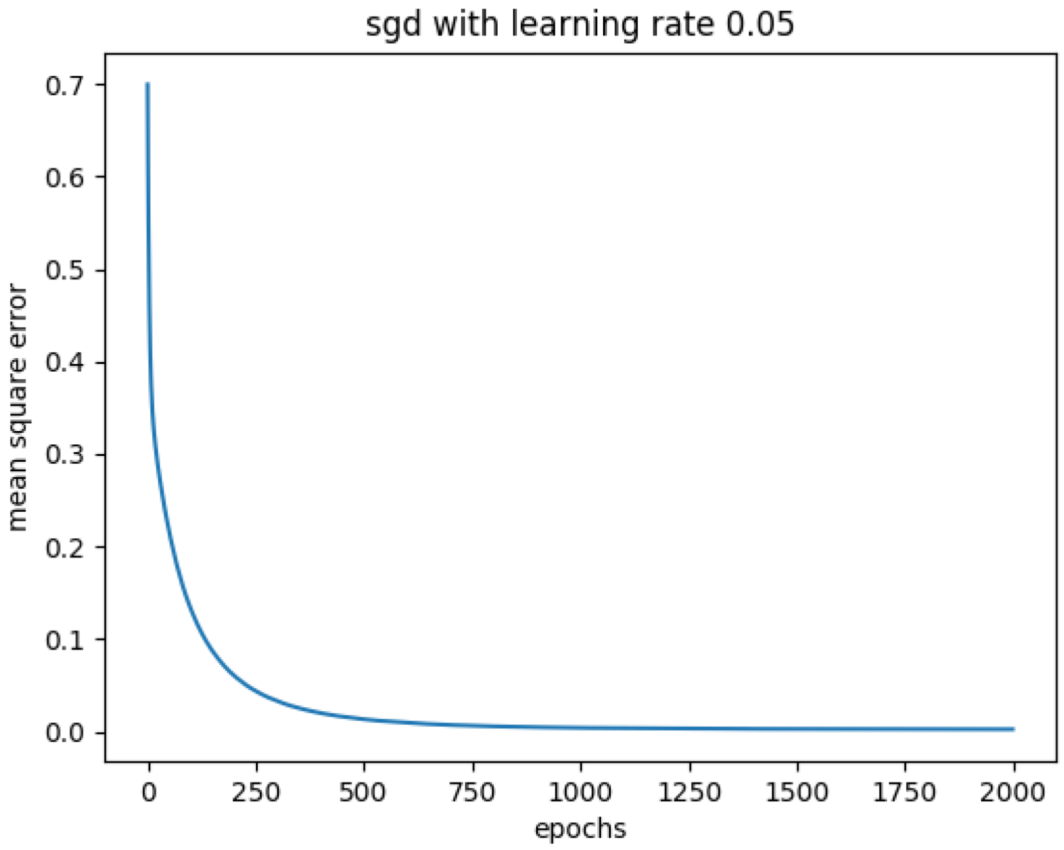
$$\nabla_{\boldsymbol{u}}J = -(\boldsymbol{d} - \boldsymbol{y}) \cdot f'(\boldsymbol{u}) = \left(\begin{pmatrix} 1.19 \\ 1.7 \end{pmatrix} - \begin{pmatrix} 1.18 \\ 1.45 \end{pmatrix}\right) \cdot \begin{pmatrix} 0.48 \\ 0.40 \end{pmatrix} = \begin{pmatrix} 0.00 \\ -0.10 \end{pmatrix}$$

$$\boldsymbol{W} = \boldsymbol{W} - \alpha\boldsymbol{x}(\nabla_{\boldsymbol{u}}J)^T = \begin{pmatrix} 1.23 & 0.11 \\ -0.48 & 0.96 \end{pmatrix} - 0.05\begin{pmatrix} 0.69 \\ 0.95 \end{pmatrix}\begin{pmatrix} 0.00 & -0.10 \end{pmatrix} = \begin{pmatrix} 1.23 & 0.11 \\ -0.48 & 0.97 \end{pmatrix}$$

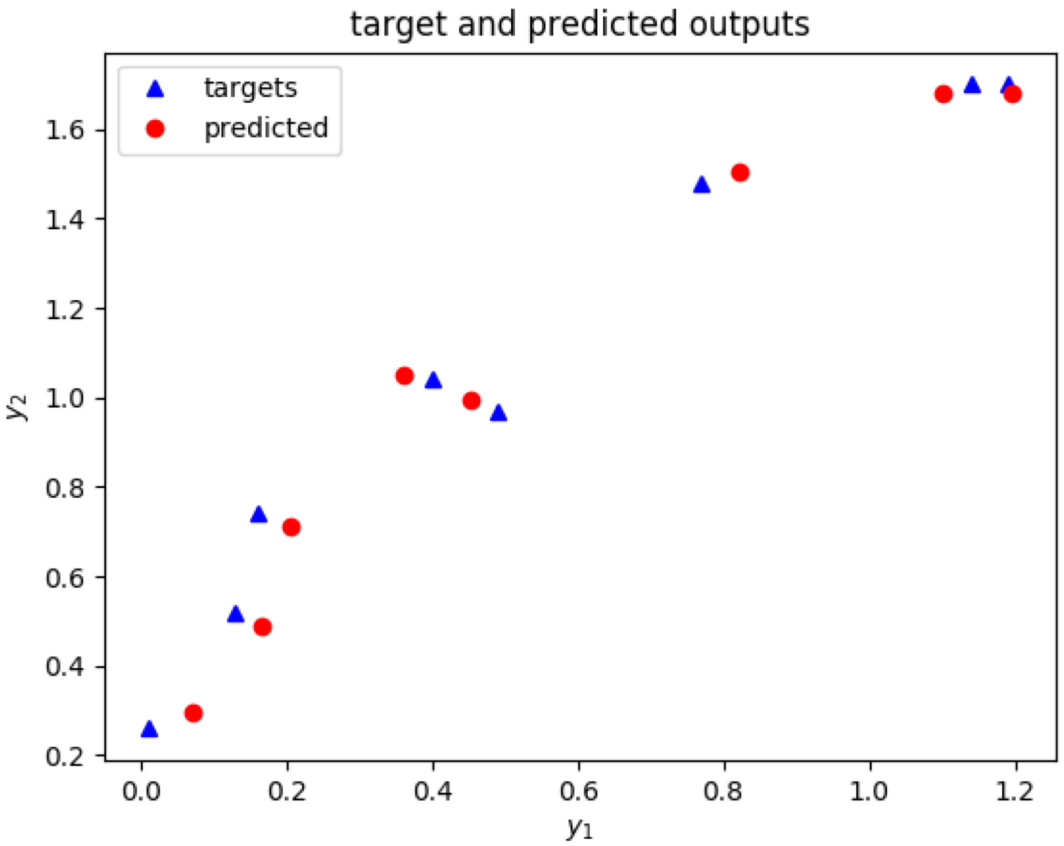
$$\boldsymbol{b} = \boldsymbol{b} - \alpha\nabla_{\boldsymbol{u}}J = \begin{pmatrix} -0.03 \\ -0.02 \end{pmatrix} - 0.05\begin{pmatrix} 0.00 \\ -0.10 \end{pmatrix} = \begin{pmatrix} -0.03 \\ -0.01 \end{pmatrix}$$

Iteration 1:

$\boldsymbol{x}$	$\boldsymbol{d}$	$\boldsymbol{u}$	$\boldsymbol{y}$	s.e.	$\nabla_u J$	$\boldsymbol{w}$	$\boldsymbol{b}$
$\begin{pmatrix}0.17\\0.09\end{pmatrix}$	$\begin{pmatrix}0.01\\0.26\end{pmatrix}$	$\begin{pmatrix}0.17\\0.11\end{pmatrix}$	$\begin{pmatrix}1.08\\1.05\end{pmatrix}$	1.78	$\begin{pmatrix}0.53\\0.40\end{pmatrix}$	$\begin{pmatrix}1.23 & 0.11\\-0.48 & 0.96\end{pmatrix}$	$\begin{pmatrix}-0.03\\-0.02\end{pmatrix}$
$\begin{pmatrix}0.69\\0.95\end{pmatrix}$	$\begin{pmatrix}1.19\\1.7\end{pmatrix}$	$\begin{pmatrix}0.37\\0.97\end{pmatrix}$	$\begin{pmatrix}1.18\\1.45\end{pmatrix}$	0.06	$\begin{pmatrix}0.00\\-0.10\end{pmatrix}$	$\begin{pmatrix}1.23 & 0.11\\-0.48 & 0.97\end{pmatrix}$	$\begin{pmatrix}-0.03\\-0.01\end{pmatrix}$
$\begin{pmatrix}0.72\\0.29\end{pmatrix}$	$\begin{pmatrix}0.4\\1.04\end{pmatrix}$	$\begin{pmatrix}0.72\\0.35\end{pmatrix}$	$\begin{pmatrix}1.35\\1.17\end{pmatrix}$	0.91	$\begin{pmatrix}0.42\\0.06\end{pmatrix}$	$\begin{pmatrix}1.22 & 0.11\\-0.48 & 0.97\end{pmatrix}$	$\begin{pmatrix}-0.05\\-0.02\end{pmatrix}$
$\begin{pmatrix}0.92\\0.72\end{pmatrix}$	$\begin{pmatrix}1.14\\1.7\end{pmatrix}$	$\begin{pmatrix}0.73\\0.78\end{pmatrix}$	$\begin{pmatrix}1.35\\1.37\end{pmatrix}$	0.15	$\begin{pmatrix}0.09\\-0.14\end{pmatrix}$	$\begin{pmatrix}1.21 & 0.12\\-0.49 & 0.97\end{pmatrix}$	$\begin{pmatrix}-0.05\\-0.01\end{pmatrix}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\vdots$

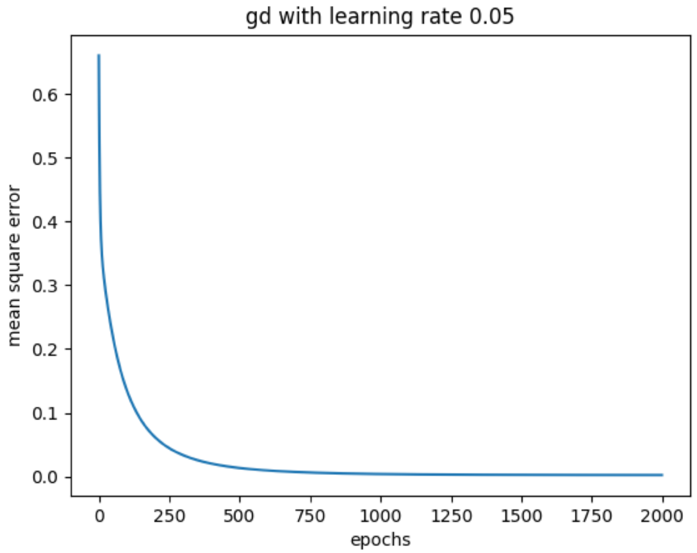


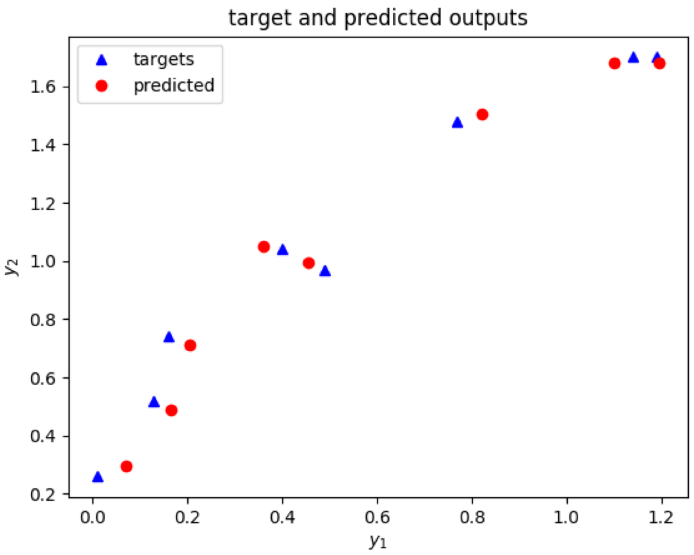
sgd with learning rate 0.05
mean square error
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
250
500
750
1000
1250
1500
1750
2000
epochs



target and predicted outputs
targets
predicted
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
y_2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
y_1

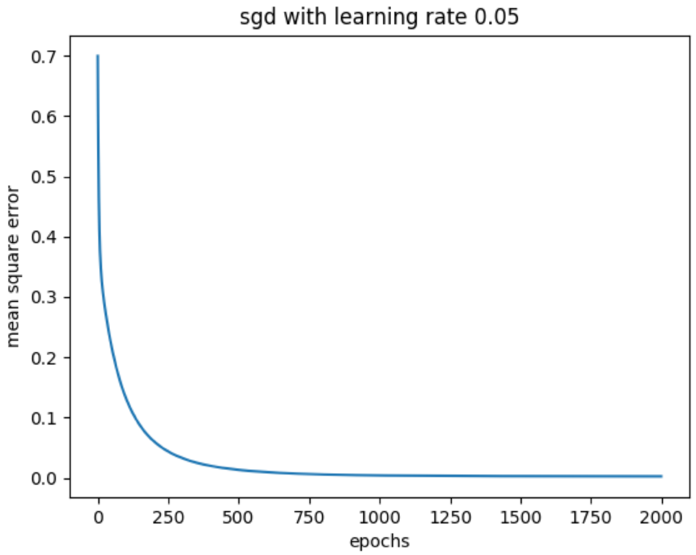
GD

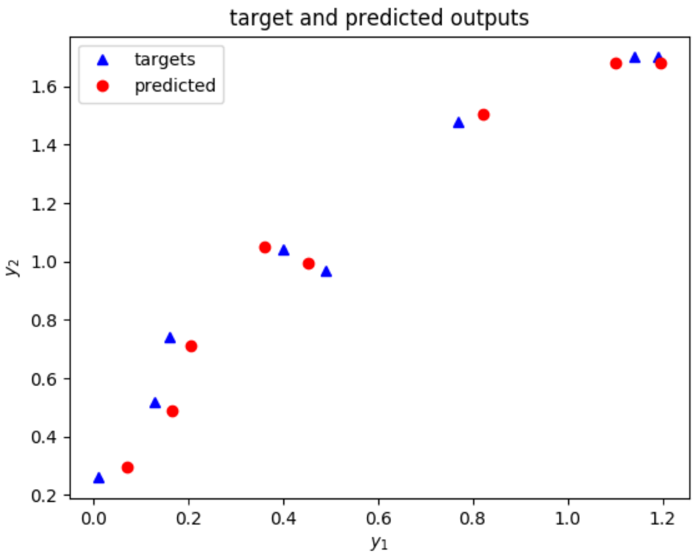




m.s.e = 0.002

SGD





m.s.e = 0.002