[HW1_Prob2]_2-D_Perceptron_Training_with_Gradient_Descent.ipynb

In [19]:
```
### Training with manually updating W with "Backward" ###

import torch
#from torch.autograd import Variable
import torch.nn as nn
import torch.nn.functional as F

import torch.optim as optim




data = [(1.0,2.1,3.0), (2.0, 3.5, 6.0), (3.0, 3.0, 9.0), (4.0, 2.1, 12
x = torch.Tensor([d[:2] for d in data])
y = torch.Tensor([d[2] for d in data])

class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.fc1 = nn.Linear(2,5)
        self.fc2 = nn.Linear(5,10)
        self.fc3 = nn.Linear(10,1)

    def forward(self, x):
        x = self.fc1(x)
        x = F.relu(x)
        x = self.fc2(x)
        x = F.relu(x)
        x = self.fc3(x)
        x = F.relu(x)
        return x



net = Net()

print(net)
print(list(net.parameters()))

#input = torch.randn(1)
#out = net(input)

#def criterion(out, label):
#    return (label - out)**2
criterion = nn.MSELoss()


optimizer = optim.SGD(net.parameters(), lr=0.01, momentum=0.5)
```

```
#optimizer = optim.Adam(net.parameters(), lr=0.005)
for epoch in range(300):
    for i in range(0,len(x),len(x)//2):
        x_batch = x[i:i+len(x)//2]
        y_batch = y[i:i+len(x)//2]
        optimizer.zero_grad()
        out = net(x)
        loss = criterion(out, y)
        loss.backward()
        optimizer.step()
```

```
Net(
  (fc1): Linear(in_features=2, out_features=5, bias=True)
  (fc2): Linear(in_features=5, out_features=10, bias=True)
  (fc3): Linear(in_features=10, out_features=1, bias=True)
)
[Parameter containing:
tensor([[ 0.3118,  0.6851],
        [-0.3832,  0.1838],
        [-0.3732,  0.3696],
        [-0.0401, -0.0783],
        [ 0.1446,  0.1558]], requires_grad=True), Parameter containin
g:
tensor([-0.0506, -0.4546,  0.2436, -0.4431, -0.4523], requires_grad=T
rue), Parameter containing:
tensor([[ 0.4418, -0.2442, -0.4411, -0.0388, -0.1910],
        [ 0.2479,  0.2764,  0.4288, -0.4169,  0.1399],
        [-0.1059, -0.0392, -0.3316, -0.0323,  0.4375],
        [-0.1685, -0.1100,  0.1804, -0.2218, -0.3665],
        [ 0.3221,  0.1651, -0.3664, -0.0207, -0.0854],
        [ 0.3795, -0.4455,  0.1479, -0.1126, -0.2472],
        [ 0.1670, -0.1769,  0.4470, -0.2205, -0.3857],
        [-0.0336,  0.0752,  0.2979, -0.0482,  0.3269],
        [ 0.4138, -0.3692, -0.2594, -0.1371, -0.2753],
        [ 0.2758, -0.2787,  0.0206,  0.1032, -0.1971]], requires_grad
=True), Parameter containing:
tensor([-0.0317, -0.2902, -0.0321, -0.1589, -0.1082, -0.0515, -0.054
4, -0.2508,
        -0.3219,  0.1275], requires_grad=True), Parameter containing:
tensor([[-0.1136, -0.1010, -0.2541, -0.1794, -0.0092, -0.1849, -0.201
8, -0.2264,
          0.3065, -0.0363]], requires_grad=True), Parameter containin
g:
tensor([-0.0636], requires_grad=True)]
```

In [18]:
```
list(net.parameters())
```

Out[18]:
```
[Parameter containing:
 tensor([[ 0.5507,  0.6671],
         [ 0.0057, -0.0594],
         [ 0.1171,  0.3885],
         [-0.4017,  0.3217],
         [ 0.3566,  0.3903]], requires_grad=True),
 Parameter containing:
 tensor([-0.2040, -0.5926,  0.1808,  0.2001, -0.0822], requires_grad=
True),
 Parameter containing:
 tensor([[-0.4307,  0.0538, -0.4318, -0.3961, -0.3446],
         [-0.1856,  0.0666, -0.2548,  0.0757,  0.3238],
         [-0.2463,  0.0719,  0.2261, -0.2641, -0.2735],
         [-0.3800,  0.1741,  0.1700,  0.3176, -0.2003],
         [ 0.2596, -0.0236,  0.2906, -0.0195,  0.1158],
         [-0.1210, -0.2117,  0.3186, -0.0444, -0.0286],
         [ 0.4398,  0.3341,  0.0049, -0.3943, -0.1575],
         [-0.1439,  0.1994, -0.0898, -0.0133, -0.4191],
         [ 0.1297,  0.2036,  0.2468, -0.2808,  0.2951],
         [ 0.3672,  0.1684,  0.4064,  0.2457,  0.3445]], requires_gra
d=True),
 Parameter containing:
 tensor([-0.0924,  0.0385, -0.0651,  0.1596,  0.0412, -0.1266,  0.058
9, -0.4239,
          0.3144,  0.2214], requires_grad=True),
 Parameter containing:
 tensor([[ 0.0098,  0.0273,  0.1927, -0.3146, -0.1369, -0.2542,  0.10
59,  0.1735,
           0.1179, -0.2736]], requires_grad=True),
 Parameter containing:
 tensor([-0.2404], requires_grad=True)]
```

In [13]:
```

```

Out[13]:
```
tensor([[ 1.0000,  2.1000],
        [ 2.0000,  3.5000],
        [ 3.0000,  3.0000],
        [ 4.0000,  2.1000],
        [ 5.0000,  7.2000],
        [ 6.0000, 10.1000]])
```

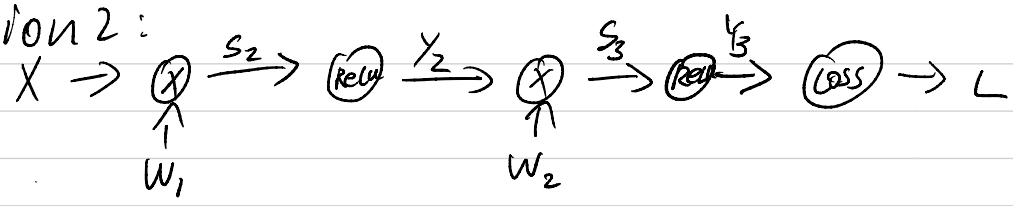
In [14]:
```
y
```

Out[14]:
```
tensor([ 3.,  6.,  9., 12., 15., 18.])
```

In [ ]:
```

```

hand-written answers to questions 2, 3, and 4

Question 2:



$$\frac{\partial L}{\partial Y_3} = \begin{bmatrix} 5 \\ 10 \end{bmatrix} = \begin{bmatrix} Y_{31} \\ Y_{32} \end{bmatrix}$$

From the forward there is no negative value so I have assum the relu layer is passthrough

$$Y_2 = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

$$Y_3 = [2\ 3] \begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix}$$

$$Y_3 = \begin{bmatrix} Y_{21} \cdot W_{211} + Y_{22} \cdot W_{221} \\ Y_{21} \cdot W_{12} + Y_{22} \cdot W_{222} \end{bmatrix}$$

$$Y_3 = [5\ \ 10]$$

$$\frac{\partial L}{\partial W} = \begin{bmatrix} \frac{\partial L}{\partial Y_{31}} \cdot Y_{21} & \frac{\partial L}{\partial Y_{32}} \cdot Y_{21} \\ \frac{\partial L}{\partial Y_{31}} \cdot Y_{22} & \frac{\partial L}{\partial Y_{32}} \cdot Y_{22} \end{bmatrix}$$

$$\frac{\partial L}{\partial W_2} = \begin{bmatrix} 5 \cdot 2 & 10 \cdot 2 \\ 5 \cdot 3 & 10 \cdot 3 \end{bmatrix}$$

$$\frac{\partial L}{\partial W_2} = \begin{bmatrix} 10 & 20 \\ 15 & 30 \end{bmatrix}$$

$$\frac{\partial L}{\partial Y_{31}} \cdot \frac{\partial Y_{31}}{\partial W_{ij}} + \frac{\partial L}{\partial Y_{32}} \cdot \frac{\partial Y_{32}}{\partial W_{ij}}$$

$$\frac{\partial L}{\partial W_{11}} = \frac{\partial L}{\partial Y_{31}} \cdot \frac{\partial Y_{31}}{\partial W_{11}} + \frac{\partial L}{\partial Y_{32}} \cdot \frac{\partial Y_{32}}{\partial W_{11}}$$

$$= \frac{\partial L}{\partial Y_{31}} \cdot Y_{21}$$

$$\frac{\partial L}{\partial W_{12}} = \frac{\partial L}{Y_{31}} \cdot \frac{\partial Y_{31}}{\partial W_{12}} + \frac{\partial L}{\partial Y_{32}} \cdot \frac{\partial Y_{32}}{\partial W_{12}}$$

$$= \frac{\partial L}{\partial Y_{32}} \cdot Y_{21}$$

$$\frac{\partial L}{\partial W_{21}} = \frac{\partial L}{Y_{31}} \cdot \frac{\partial Y_{31}}{\partial W_{21}} = \frac{\partial L}{Y_{31}} \cdot Y_{22}$$

$$\frac{\partial L}{\partial W_{22}} = \frac{\partial L}{Y_{32}} \cdot \frac{\partial Y_{32}}{\partial W_{22}} = \frac{\partial L}{Y_{32}} \cdot Y_{22}$$

$S_2 = X \cdot W_1 \qquad S_3 = S_2 \cdot W_2 \qquad W_1 = \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix} \qquad W_2 = \begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix}$

$\frac{\partial L}{Y_B} = \begin{bmatrix} 5 \\ 10 \end{bmatrix} \qquad S_2 = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$

$S_3 = \begin{bmatrix} S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} w_{211} & w_{212} \\ w_{221} & w_{222} \end{bmatrix} = \begin{bmatrix} S_{21} \cdot w_{211} + S_{22} w_{221} \\ S_{21} \cdot w_{212} + S_{22} w_{222} \end{bmatrix}$

$\frac{\partial L}{\partial Y_{31}} \cdot \frac{\partial Y_{31}}{\partial Y_{2i}} + \frac{\partial L}{\partial Y_{32}} \cdot \frac{\partial Y_{32}}{\partial Y_{2i}}$

$\frac{\partial L}{\partial Y_{31}} \cdot \frac{\partial Y_{31}}{\partial Y_{21}} + \frac{\partial L}{\partial Y_{32}} \cdot \frac{\partial Y_{32}}{\partial Y_{21}} = \frac{\partial L}{\partial Y_{31}} \cdot w_{211} + \frac{\partial L}{\partial Y_{32}} \cdot w_{212}$

$\frac{\partial L}{\partial Y_{31}} \cdot \frac{\partial Y_{31}}{\partial Y_{22}} + \frac{\partial L}{\partial Y_{32}} \cdot \frac{\partial Y_{32}}{\partial Y_{22}} = \frac{\partial L}{\partial Y_{31}} \cdot w_{221} + \frac{\partial L}{\partial Y_{32}} \cdot w_{222}$

$\frac{\partial L}{\partial Y_2} = \begin{bmatrix} 5 \cdot 1 + 10 \cdot 2 \\ 5 \cdot 1 + 10 \cdot 2 \end{bmatrix} = \begin{bmatrix} 25 \\ 25 \end{bmatrix}$

$\frac{\partial L}{\partial S_2} =$

$\frac{\partial L}{\partial W_1} = \frac{\partial L}{\partial S_2} \cdot \frac{\partial S_2}{\partial W_1} \qquad X = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$

$S_2 = X^T W_1 = \begin{bmatrix} 1 & 2 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix}$

$S_2 = \begin{bmatrix} X_1 & X_2 \end{bmatrix} \begin{bmatrix} w_{111} & w_{112} \\ w_{121} & w_{122} \end{bmatrix}$

$S_2 \begin{bmatrix} X_1 w_{111} + X_2 w_{121} \\ X_1 w_{112} + X_2 w_{122} \end{bmatrix}$

$\frac{\partial L}{\partial S_{21}} \frac{\partial S_{21}}{\partial w_{1ij}} + \frac{\partial L}{\partial S_{22}} \cdot \frac{\partial S_{22}}{\partial w_{1ij}}$

$w_{111} = \frac{\partial L}{\partial S_{21}} \cdot \frac{\partial S_{21}}{\partial w_{111}} = \frac{\partial L}{\partial S_{21}} \cdot X_1$

$w_{112} = \frac{\partial L}{\partial S_{21}} \cdot \frac{\partial S_{21}}{\partial w_{112}}^{0} + \frac{\partial L}{\partial S_{22}} \cdot \frac{\partial S_{22}}{\partial w_{112}} = \frac{\partial L}{\partial S_{22}} \cdot X_1$

$\frac{\partial L}{\partial W_1} = \begin{bmatrix} 25 \cdot 1 & 25 \cdot 1 \\ 25 \cdot 2 & 25 \cdot 2 \end{bmatrix} = \begin{bmatrix} 25 & 25 \\ 50 & 50 \end{bmatrix}$

$$w_{121} = \frac{\partial L}{\partial s_{21}} \cdot \frac{\partial s_{21}}{\partial w_{121}} = \frac{\partial L}{\partial s_{21}} \cdot x_2$$

$$w_{122} = \frac{\partial L}{\partial s_{22}} \cdot \frac{\partial s_{22}}{\partial w_{122}} = \frac{\partial L}{\partial s_{22}} \cdot x_2$$

Problem 3.

$S_2 = X W_1 \quad Y_2 = \text{relu}(S_2) \quad S_3 = Y_2 W_2 \quad Y_3 = \text{relu}(S_3) \quad \text{Loss} = \frac{1}{N}\sum (\hat{y}-y)^2$

$X = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \quad W_1 = \begin{bmatrix} 2 & 1 \\ 0 & -1 \end{bmatrix} \quad W_2 = \begin{bmatrix} -1 & 2 \\ 1 & 2 \end{bmatrix}$

Foward: $S_2 = X W_1$

$S_2 = [1\ 2]\begin{bmatrix} 2 & 1 \\ 0 & -1 \end{bmatrix} = [2\ -1] = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$

$Y_2 = \begin{bmatrix} 2 \\ -1 \end{bmatrix}[1\ 0] = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$

$S_3 = [2\ 0]\begin{bmatrix} -1 & 2 \\ 1 & 2 \end{bmatrix} = [-2\ \ 4] = \begin{bmatrix} -2 \\ 4 \end{bmatrix} = [$

$Y_3 = \begin{bmatrix} -2 \\ 4 \end{bmatrix}[0\ \ 1] = \begin{bmatrix} S_{31}\cdot r_1 \\ S_{32}\cdot r_2 \end{bmatrix} = \begin{bmatrix} -2\cdot 0 \\ 4\cdot 1 \end{bmatrix} = \begin{bmatrix} -2 \\ 4 \end{bmatrix})$

$\text{Loss} = \frac{1}{2}(0)^2 + \frac{1}{2}(4)^2 = 0 + \frac{16}{2} = 8$

$\frac{\partial L}{\partial Y_3} = \frac{2}{2}(Y_3 - Y)\cdot(1) = (Y_3 - Y) \qquad \frac{\partial L}{\partial Y_3} = \begin{bmatrix} 0 \\ 4 \end{bmatrix}$

$\frac{\partial L}{\partial S_3} = \frac{\partial L}{\partial Y_3}\cdot\frac{\partial Y_3}{\partial S_3} = \frac{\partial L}{\partial Y_{31}}\cdot\frac{\partial Y_{31}}{\partial S_{3i}} + \frac{\partial L}{\partial Y_{32}}\cdot\frac{\partial Y_{32}}{\partial S_i}$

$\text{for } S_{31} = \frac{\partial L}{\partial Y_{31}}\cdot\frac{\partial Y_{31}}{\partial S_{31}} + \frac{\partial L}{\partial Y_{32}}\cdot\frac{\partial Y_{32}}{\partial S_{31}} = \frac{\partial L}{\partial Y_{31}}\cdot 0 + \frac{\partial}{\partial Y_{32}}\cdot 0 = 0$

$\text{for } S_{32} = \frac{\partial L}{\partial Y_{31}}\cdot\frac{\partial Y_{31}}{\partial S_{32}} + \frac{\partial L}{\partial Y_{32}}\cdot\frac{\partial Y_{32}}{\partial S_{32}} = \frac{\partial L}{\partial Y_{32}}\cdot 1 = \frac{\partial L}{\partial Y_{32}} = 4$

$\frac{\partial L}{\partial S_3} = \begin{bmatrix} 0 \\ 4 \end{bmatrix}$

$\frac{\partial L}{\partial W_2} = \frac{\partial L}{\partial S_3}\cdot\frac{\partial S_3}{\partial W_2} \qquad S_3 = Y_2 W_2$

$S_3 = [Y_{21}\ \ Y_{22}]\begin{bmatrix} W_{211} & W_{212} \\ W_{221} & W_{222} \end{bmatrix}$

$S_3 = \begin{bmatrix} Y_{21}\cdot W_{211} + Y_{22}\cdot W_{221} \\ Y_{21}\cdot W_{212} + Y_{22}\cdot W_{222} \end{bmatrix}$

$$\frac{\partial L}{\partial W_2} = \frac{\partial L}{\partial S_{31}} \cdot \frac{\partial S_{31}}{\partial W_{2ij}} + \frac{\partial L}{\partial S_{32}} \cdot \frac{\partial S_{32}}{\partial W_{2ij}} = \begin{bmatrix} 0 & 4\cdot 2 \\ 0 & 4\cdot 0 \end{bmatrix} = \begin{bmatrix} 0 & 8 \\ 0 & 0 \end{bmatrix}$$

$$W_{211} = \frac{\partial L}{\partial S_{31}} \cdot \frac{\partial S_{31}}{\partial W_{211}} = \frac{\partial L}{\partial S_{31}} \cdot Y_{21}$$

$$W_{212} = \frac{\partial L}{\partial S_{32}} \cdot \frac{\partial S_{32}}{\partial W_{212}} = \frac{\partial L}{\partial S_{32}} \cdot Y_{21}$$

$$W_{221} = \frac{\partial L}{\partial S_{31}} \cdot \frac{\partial S_{31}}{\partial W_{221}} = \frac{\partial L}{\partial S_{31}} \cdot Y_{22}$$

$$W_{222} = \frac{\partial L}{\partial S_{32}} \frac{\partial S_{32}}{\partial W_{222}} = \frac{\partial L}{\partial S_{32}} \cdot Y_{22}$$

$$\frac{\partial L}{\partial S_3} = \begin{bmatrix} 0 \\ 4 \end{bmatrix}$$

$$W_2 = \begin{bmatrix} -1 & 2 \\ 1 & 2 \end{bmatrix}$$

$$\frac{\partial L}{\partial Y_2} = \frac{\partial L}{\partial S_3} \cdot \frac{\partial S_3}{\partial Y_2} = \frac{\partial L}{\partial S_{31}} \cdot \frac{\partial S_{31}}{\partial Y_{21}} + \frac{\partial L}{\partial S_{32}} \cdot \frac{\partial S_{32}}{\partial Y_{21}}, \quad \frac{\partial L}{\partial S_{31}} \cdot \frac{\partial S_{31}}{\partial Y_{22}} + \frac{\partial L}{\partial S_{32}} \cdot \frac{\partial S_{32}}{\partial Y_{22}}$$

$$Y_{21} = \frac{\partial L}{\partial S_{31}} \cdot W_{211} + \frac{\partial L}{\partial S_{32}} \cdot W_{212} \qquad Y_{22} = \frac{\partial L}{\partial S_{31}} \cdot W_{221} + \frac{\partial L}{\partial S_{32}} \cdot W_{222}$$

$$\frac{\partial L}{\partial Y_2} = \begin{bmatrix} 0 + 4\cdot 2 \\ 0 + 4\cdot 2 \end{bmatrix} = \begin{bmatrix} 8 \\ 8 \end{bmatrix}$$

$$\begin{bmatrix} S_{21} \\ S_{22} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ r_1 & r_2 \end{bmatrix} = \begin{bmatrix} S_{21} r_1 \\ S_{22} r_2 \end{bmatrix}$$

$$\frac{\partial L}{\partial S_2} = \frac{\partial L}{\partial Y_2} \frac{\partial Y_2}{\partial S_2} = \frac{\partial L}{\partial Y_{21}} \cdot \frac{\partial Y_{21}}{\partial S_{2i}} + \frac{\partial L}{\partial Y_{22}} \cdot \frac{\partial Y_{22}}{\partial S_{2i}} = \begin{bmatrix} 8 \\ 0 \end{bmatrix} = \frac{\partial L}{\partial S_2}$$

for $S_{21} = \frac{\partial L}{\partial Y_{21}} \cdot r_1 + \frac{\partial L}{\partial Y_{22}} \cdot 0 = 8$

$S_{22} = \frac{\partial L}{\partial Y_{21}} \cdot 0 + \frac{\partial L}{\partial Y_{22}} \cdot r_2 = 0$

$S_2 = X W_1 \quad Y_2 = \text{relu}(S_2) \quad S_3 = Y_2 W_2 \quad Y_3 = \text{relu}(S_3)$

$X = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \quad W_1 = \begin{bmatrix} 2 & 1 \\ 0 & -1 \end{bmatrix} \quad W_2 = \begin{bmatrix} -1 & 2 \\ 1 & 2 \end{bmatrix} \quad \frac{\partial L}{\partial S_2} = \begin{bmatrix} 8 \\ 0 \end{bmatrix}$

$$\frac{\partial L}{\partial W_1} = \frac{\partial L}{\partial S_2} \cdot \frac{\partial S_2}{\partial W_1} = \frac{\partial L}{\partial S_{21}} \cdot \frac{\partial S_{21}}{\partial W_{ij}} + \frac{\partial L}{\partial S_{22}} \cdot \frac{\partial S_{22}}{\partial W_{ij}}$$

$S_2 = X W_1$

$$S_2 = X^T W_1 = \begin{matrix} x_1 & x_2 \end{matrix} \begin{bmatrix} 1 & 2 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 0 & -1 \end{bmatrix} \begin{matrix} w_{111} & w_{112} \\ w_{121} & w_{122} \end{matrix}$$

$$S_2 = \begin{bmatrix} x_1 w_{111} + x_2 w_{121} \\ x_1 w_{112} + x_2 w_{122} \end{bmatrix}$$

$w_{111} = \frac{\partial L}{\partial S_{21}} \cdot x_1$

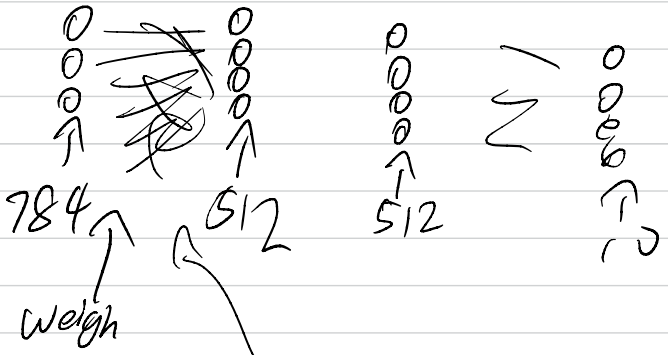
$w_{121} = \frac{\partial L}{\partial S_{21}} \cdot x_2$

$w_{112} = \frac{\partial L}{\partial S_{22}} \cdot x_1$

$w_{122} = \frac{\partial L}{\partial S_{22}} \cdot x_2$

$$\frac{\partial L}{\partial W_1} = \begin{bmatrix} 8 \cdot 1 & 0 \\ 8 \cdot 2 & 0 \end{bmatrix} = \begin{bmatrix} 8 & 0 \\ 16 & 0 \end{bmatrix}$$

Question 4:



$(784 \cdot 512) + 512 = 401\,920$

$(512 \cdot 512) + 512 = 262,656$ $+ = 669,706$ total parameter

$(512 \cdot 10) + 10 = 5130$

1 image = 784 input $\quad y = xw + b$

784 node (512 mutiply) + 512 node (784 addition) + 512 relu

512 node (512 mutiply) + 512 node (512 addition) + 512 relu

512 node (10 mutiply) + 10 node (512 addition) + 10 relu

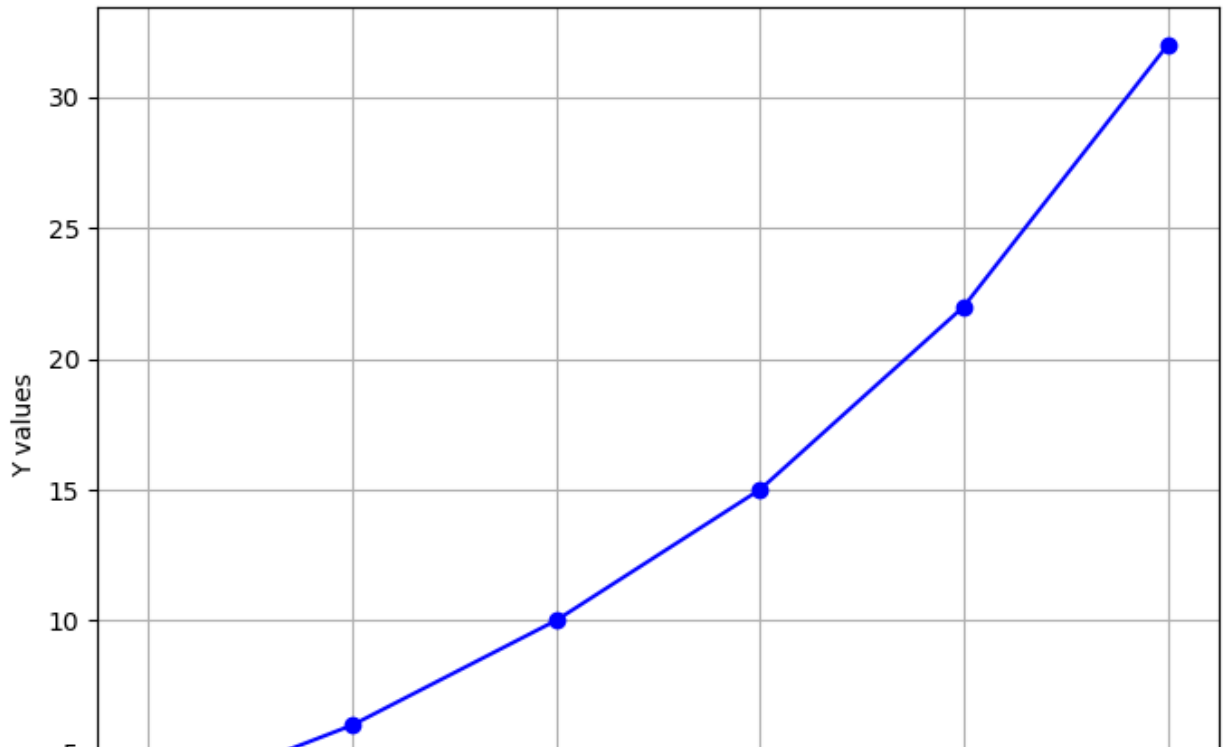
$= 1,338,378$ total operation in 1 forward pass

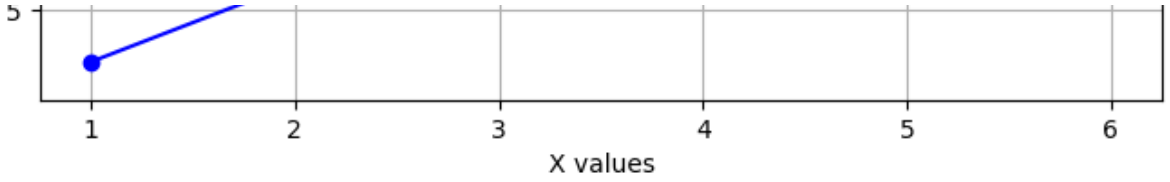
[W2_hw]_Mult-layer_Perceptron_Non_linear_Training.ipynb

In [96]:
```
### Training with fancier version ###

import torch
import torch.nn as nn
import torch.nn.functional as F

import torch.optim as optim
import matplotlib.pyplot as plt

class Net(nn.Module): ## nn.Module class is used
    def __init__(self):
        super(Net, self).__init__()
        self.fc1 = nn.Linear(1,4096,bias=False)  # in dim, out dim
        self.fc2 = nn.Linear(4096,2048,bias=False)
        self.fc3 = nn.Linear(2048,1,bias=False)

    def forward(self, x):
        x = self.fc1(x)
        x = F.sigmoid(x)
        x = self.fc2(x)
        x = F.sigmoid(x)
        x = self.fc3(x)
        return x

net = Net()

print(net)
print(list(net.parameters())) # parameters are randomized


#def criterion(out, label):
#    return (label - out)**2
criterion = nn.MSELoss()


# optimizer = optim.SGD(net.parameters(), lr=0.01, momentum=0.9)
optimizer = optim.Adam(net.parameters(), lr=0.00007,)


data = [(1.0,3.0), (2.0,6.0), (3.0,10.0), (4.0,15.0), (5.0,22.0), (6.0

# Split data into x and y
x_values, y_values = zip(*data)

# Plotting
plt.figure(figsize=(8, 6))
plt.plot(x_values, y_values, marker='o', linestyle='-', color='b')
plt.xlabel('X values')
```

```
plt.ylabel('Y values')
plt.grid(True)
plt.show()
```

```
Net(
  (fc1): Linear(in_features=1, out_features=4096, bias=False)
  (fc2): Linear(in_features=4096, out_features=2048, bias=False)
  (fc3): Linear(in_features=2048, out_features=1, bias=False)
)
[Parameter containing:
tensor([[ 0.2647],
        [ 0.6125],
        [-0.9204],
        ...,
        [ 0.0500],
        [-0.9578],
        [ 0.2392]], requires_grad=True), Parameter containing:
tensor([[-0.0024, -0.0130, -0.0089,  ...,  0.0104, -0.0093,  0.0004],
        [-0.0113,  0.0020, -0.0090,  ..., -0.0022,  0.0090,  0.0082],
        [ 0.0095, -0.0065,  0.0018,  ..., -0.0002, -0.0011, -0.0053],
        ...,
        [ 0.0075, -0.0020, -0.0113,  ...,  0.0152, -0.0037, -0.0074],
        [ 0.0107,  0.0123,  0.0095,  ...,  0.0040, -0.0150,  0.0105],
        [ 0.0084,  0.0105,  0.0039,  ...,  0.0059, -0.0142, -0.001
0]],
       requires_grad=True), Parameter containing:
tensor([[ 0.0178,  0.0086,  0.0049,  ..., -0.0132, -0.0218, -0.017
1]],
       requires_grad=True)]
```





In [97]:
```
for epoch in range(1500): # 0 - 19
    for i, current_data in enumerate(data):
        X, Y = current_data
        X, Y = torch.FloatTensor([X]), torch.FloatTensor([Y])
        optimizer.zero_grad()
        outputs = net(X)
        loss = criterion(outputs, Y)
        loss.backward()
        optimizer.step()    ## This line is equivalent to "W = W - lr*
    print("Epoch {} - loss: {}".format(epoch, loss))

### Test the trained network ###
for i, current_data in enumerate(data):
    X, Y = current_data
    X, Y = torch.FloatTensor([X]), torch.FloatTensor([Y])
    out = net(torch.FloatTensor(X))
    print("when x = {}, y = {}".format(X, out))
```

```
Epoch 1486 - loss: 1.4406759873963892e-07
Epoch 1487 - loss: 1.0762596502900124e-07
Epoch 1488 - loss: 1.0637813829816878e-07
Epoch 1489 - loss: 7.754715625196695e-08
Epoch 1490 - loss: 7.968628779053688e-08
Epoch 1491 - loss: 5.6843418860808015e-08
Epoch 1492 - loss: 5.6843418860808015e-08
Epoch 1493 - loss: 4.165121936239302e-08
Epoch 1494 - loss: 4.165121936239302e-08
Epoch 1495 - loss: 2.8816430130973458e-08
Epoch 1496 - loss: 3.0126102501526475e-08
Epoch 1497 - loss: 2.0463630789890885e-08
Epoch 1498 - loss: 2.386877895332873e-08
Epoch 1499 - loss: 1.1819793144240975e-08
when x = tensor([1.]), y = tensor([3.0000], grad_fn=<SqueezeBackward4
>)
when x = tensor([2.]), y = tensor([6.0000], grad_fn=<SqueezeBackward4
>)
when x = tensor([3.]), y = tensor([10.0000], grad_fn=<SqueezeBackward
4>)
```

In [ ]:
```
for epoch in range(20):
    print(epoch)
```

In [ ]:
```
W = torch.tensor([1.0], requires_grad=True)
W = W*2
label = 1.0
loss = W*5 - label
loss.backward()
W.grad
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: