Homework 4: Backpropagation Neural Network

B07611001 楊禮蔚
B07611002 鄭泊聲
B07611016 黃評堅

Problem 1:

Code is in the attachment or accessed by the link below:
https://colab.research.google.com/drive/1EZf1X9GChxXa4c4wCy7MPmM42Kn48WdY?usp=sharing

模型架構:

```
nn.Linear(input_dim,  32),
nn.Sigmoid(),
nn.Linear(32,  64),
nn.RReLU(),
nn.Linear(64,  32),
nn.LeakyReLU(),
nn.Linear(32,  output_dim)
```

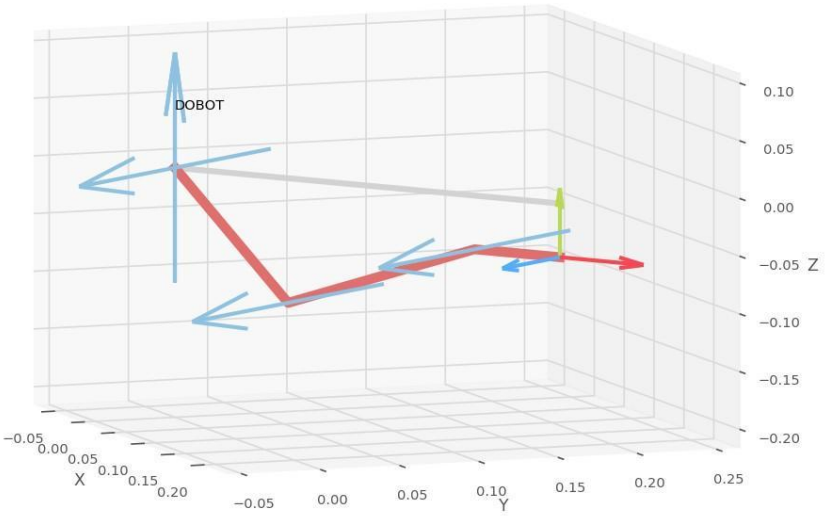
input_dim: 3, output_dim: 3

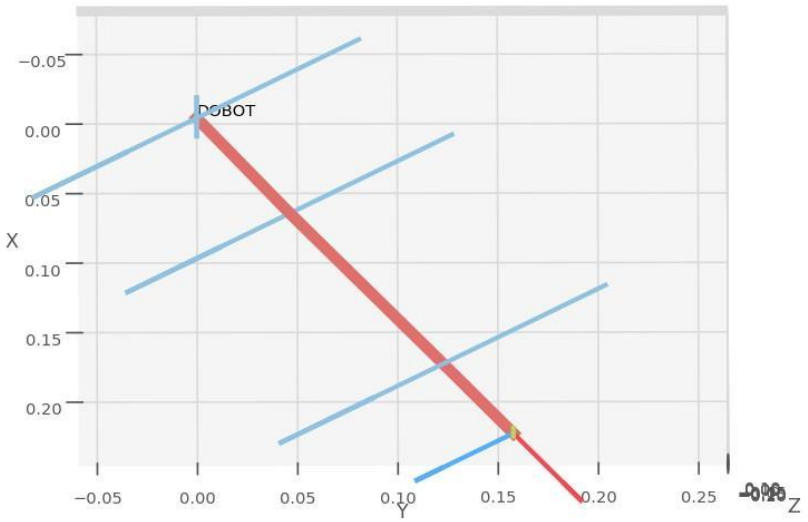
最終MSE結果:
epoch = 9730, train_loss = 0.0032, dev_loss = 0.0037

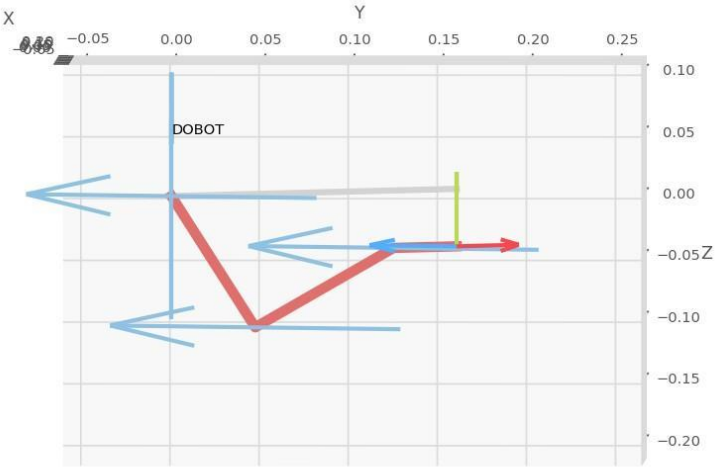
Test Dataset檢驗:

```
pred
[ 0.6087154 -0.9272763  1.3586692]

target
tensor([ 0.6289, -0.8834,  1.2974], device='cuda:0')

joint config
[ 0.60871542 -0.92727631  1.35866916 -0.43139285  0.        ]

   0.8204    0         0.5718    0.226
   0.5718    0        -0.8204    0.1576
   0         1         0        -0.04653
   0         0         0         1

X Y Z
tensor([227.4335, 165.4449,  45.2012], device='cuda:0')


PyPlot3D backend, t = 0.05, scene:
  DOBOT
```

pred：模型預測結果（Input 為下方所列的XYZ）
target：實際數據（Ground Truth）
將pred 轉成joint config（補齊其他joint的角度）
用joint config順向運動學可得到轉移矩陣，所得XYZ值，
近似於Input的XYZ
（轉移矩陣中的XYZ有經過0.001的scaling，方便顯示於下圖）







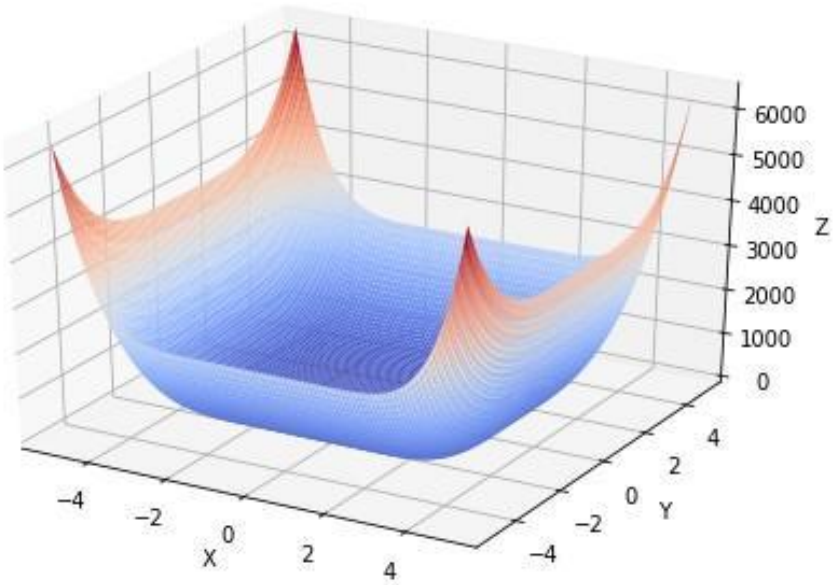
Problem 2:

Code is in the attachment or accessed by the link below:
https://colab.research.google.com/drive/1o48fu9acy7K5418KSgTTwMXOwPmZvZok?usp=sharing

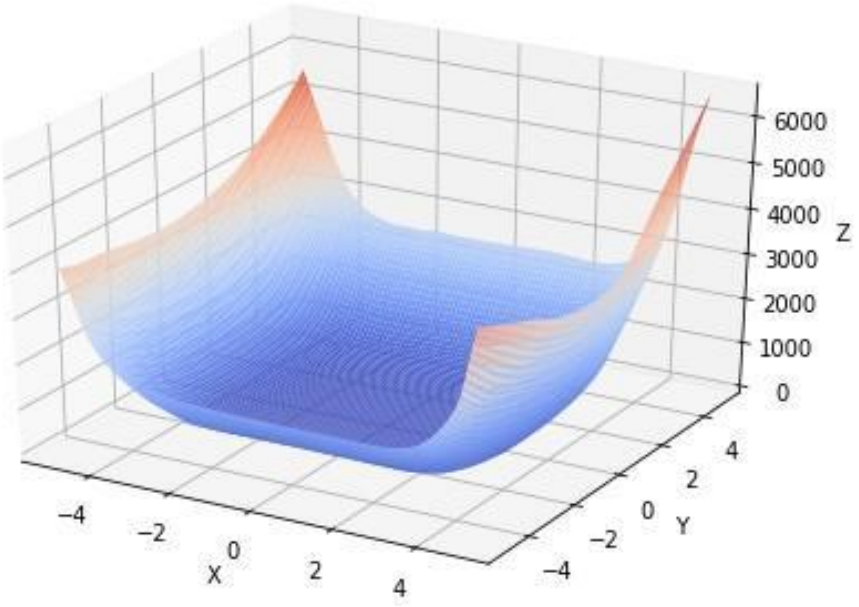
模型架構:

```
nn.Linear(in_features=2,  out_features=256,  bias=True),
nn.ELU(),
nn.Linear(256,  1024),
nn.ELU(),
nn.Linear(1024,  128),
nn.ELU(),
nn.Linear(128,  1),
```

理論數據繪圖分布如下(Ground Truth):



模型計算結果如下:



整體分布結構近似，惟在圖形邊緣極值時產生較大誤差。
最終MSE結果:
Epoch 100
loss: 5765897.000000 [0/10201]
loss: 71.376923 [1000/10201]
loss: 3.974406 [2000/10201]
loss: 94.899734 [3000/10201]
loss: 17.212233 [4000/10201]
loss: 9.615299 [5000/10201]
loss: 0.020685 [6000/10201]
loss: 29.108335 [7000/10201]
loss: 0.819741 [8000/10201]
loss: 22.270058 [9000/10201]
loss: 339674.406250 [10000/10201]