



Assignment #2

AATG013/ANT5013 Deep Learning and Computer Vision

Assigned: Thursday, 3/21, 2023

Due: Friday, 3/28, 2023 (by midnight)

- Colab code 1 set을 사이버캠퍼스에 제출
 - assignment2.ipynb

code는 colab에서 수행되어 결과를 보여야 합니다.



Problem [1] PyTorch code: the 2-10-1 neural net for binary classification

Problem [1] [50 points]

교재에서는 PyTorch로 작성된 `sequential method`의 설명을 위해 2-inputs, 8-node hidden layer가 있는 2-8-1 neural net의 예제를 보입니다. 이 code는 아래의 course github chapter 2의 `Sequential_method_to_build_a_neural_network.ipynb`에 있습니다.

<https://github.com/PacktPublishing/Modern-Computer-Vision-with-PyTorch>

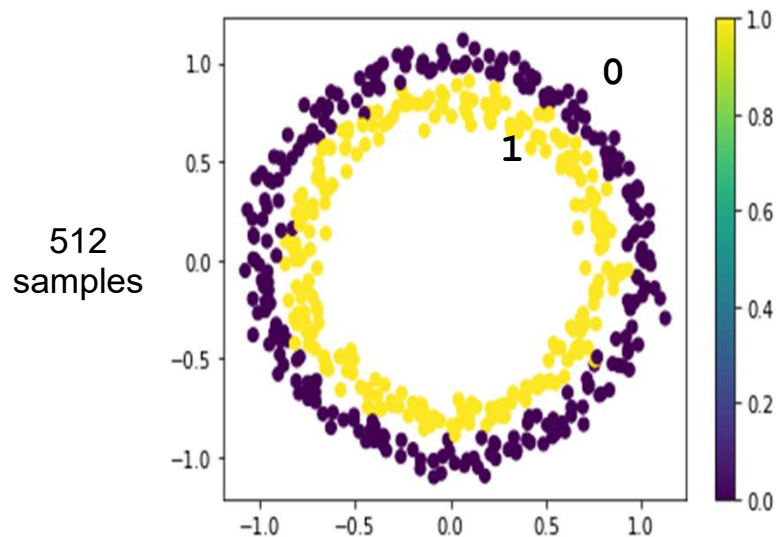
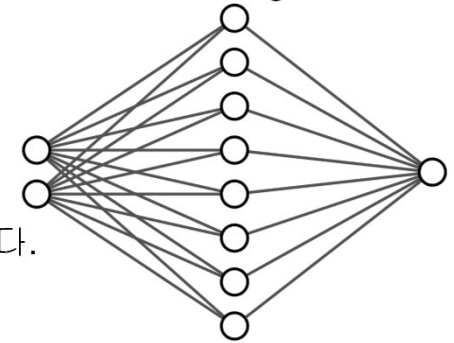
2-8-1 neural net은 아래의 input-output 짝을 받아 training 시키는데, 살펴보면 입력 두 값을 더해 출력을 내도록 training 시키는 내용입니다.

```
x = [[1, 2], [3, 4], [5, 6], [7, 8]]
```

```
y = [[3], [7], [11], [15]]
```

이 net은 continuous output을 출력하므로 MSE loss function인 `nn.MSELoss()`를 사용한 바 있습니다.

본 문제에서는 위의 입력 대신 아래의 그림에서 보인 것과 같은 input-output training set를 사용하여 주어진 입력의 출력을 1과 0으로 classify하는 code를 작성합니다.



- 그림에서 yellow points는 그 값이 1이고 purple points는 0입니다.
- 이 training set으로 훈련된 net은 임의의 test 입력을 1 혹은 0으로 classify하게 됩니다.

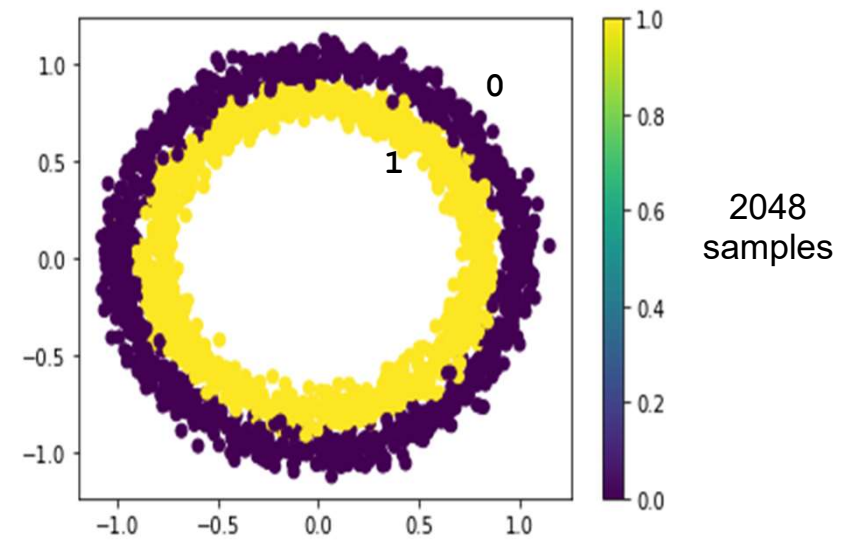


Problem [1] PyTorch code:

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn.datasets import make_circles

# Create a dataset with 2048 samples.
x, y = make_circles(n_samples=2048,
                    factor=0.8,
                    noise= 0.05)

plt.scatter(x[:,0], x[:,1], c=y)
plt.colorbar()
```



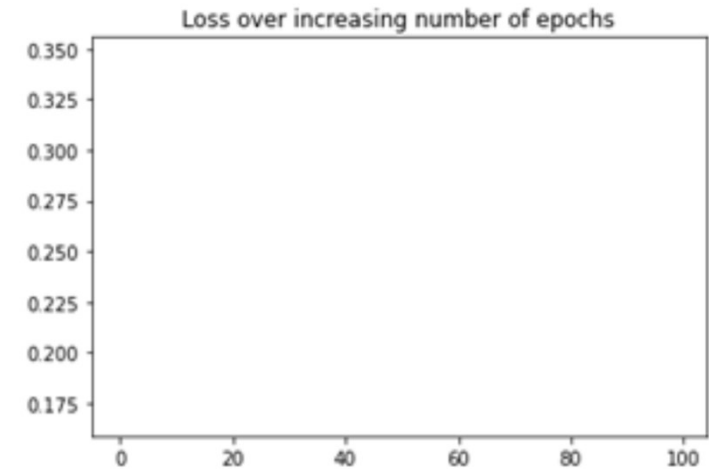
input-output training set은 `sklearn.datasets`의 `make_circles`를 사용하여 생성합니다. 자세한 내용은 online 설명서를 참조하고, 위의 code를 그대로 사용하고, `Sequential_method_to_build_a_neural_network.ipynb`를 개정합니다.

- Batch size는 64.
- Activation function은 `Sigmoid()`
- torch summary를 install하고 summary를 출력
- 이를 위해 loss function은 BCELoss (binary cross-entropy loss)를 사용합니다.
 - MSE loss function을 사용하면 감점.
 - PyTorch,를 조사하여 필요하면 `loss_func()`의 arguments matching이 잘 되도록 조정
- `loss_history`의 plot을 추가
- `W = model.state_dict()` 출력



Problem [1] PyTorch code:

```
import matplotlib.pyplot as plt
%matplotlib inline
plt.plot(loss_history)
plt.show
plt.title('Loss History')
plt.xlabel('epochs')
plt.ylabel('loss')
```



- `loss_history`의 plot을 추가
 - `loss_value`의 값이 `pyplot`이 요구하는 `format`이 아닐 경우 그 값을 적절히 변환하여 `loss_history`에 추가할 필요가 있을 수도 있음.
 - `loss_value` 값이 `cuda` 일 경우 `cpu`로 전환할 필요가 있을 수도 있음