

`make_circles()`:

`homework_classification_make_circles.ipynb`

Вдалось забрати один рівень нейронів, додавши функцію активації LeakyReLU, також пробував

```
self.sigmoid = nn.Sigmoid()# <- add in Sigmoid activation function
```

```
self.Tanh = nn.Tanh()# <- add in Tanh activation function
```

також давали цікаві результати. Вдалось досягти

```
Loss: 0.01949, Accuracy: 99.94% | Test Loss: 0.02177, Test Accuracy: 100.00%
```

кількість епох можна зменшити бже шкоди для точності

Додав графік ln результатів лос функції тренування, якась дуже нудна =)

`make_moon()`

`homework_classification_make_moon.ipynb`

та сама кількість шарів і кількості нейронів з тієюж функцією активації одразу дали досить хороший результат

```
Epoch: 900 | Loss: 0.09638, Accuracy: 96.75% | Test Loss: 0.10314, Test Accuracy: 96.25%
```

і логарифм результатів лос цункції цікавіший

`make_blobs()`

`homework_classification_make_blobs.ipynb`

як і зайнятті Linear чудово справилась з завданням, лос функція значно цікавіша, оскільки містить різну динаміку на різних відтинках епох