

## < IRIS Flower data 분류기 구현 >

컴퓨터공학과 201812663 송나은

### [1] 실험 환경(50어절 이내)

- 1) 사용 언어: Python
- 2) 하드웨어 성능: 프로세서 - Intel(R) Core(TM) i3-7020U CPU @ 2.30GHZ  
설치된 메모리(RAM) - 8.00GB  
시스템 종류 - 64비트 운영 체제, x64 기반 프로세서

### [2] 실험 방법과 실험 결과

#### 1) 실험 방법

tensorflow의 keras 모듈을 사용했다.

Neural Network에서는 random 함수를 import해 seed 값을 무작위로 설정했다. Layer은 5개이고, Hidden Node수는 첫 번째 Layer에는 64개와 input 4개, 두 번째~네 번째 Layer에는 64개, 다섯 번째 Layer에는 3개와 output 3개이다. 보통 Hidden Node를 설정할 때는 input과 output의 평균 값으로 선정하는데, 이번에는 layer는 5개, 노드는 64개로 설정해서 진행했다.

너무 많은 횟수를 학습하면 실행 시간이 오래 걸리기 때문에 학습 반복 수(epoch)는 50 회만 진행했다.

활성화 함수(Activation function)는 sigmoid - tanh - relu - elu - softmax 순으로 사용했다. sigmoid 함수는 0~1 사이의 값을 가지며, 0을 기준으로 양수를 가지면 0.5보다 커지고, 음수를 가지면 0.5보다 작아진다. 이러한 특성때문에 이번 과제에서 사용했다. 다만, input쪽으로 weight를 업데이트 해 나갈 때, 활성화 함수의 미분값이 계속해서 곱해지는데 곱할 수록 weight의 변화량이 매우 작아져 잘 업데이트 되지 않는 현상(gradient vanishing)이 발생한다.

tanh함수는 sigmoid함수의 중간이 0이 아니라서 학습이 느리다는 단점을 극복하기 때문

에 사용했지만, 여전히 gradient vanishing이 발생한다.

ReLU함수는  $x$ 가 음의 값을 가지면 0을 출력하고, 양의 값을 가지면  $x$ 를 그대로 출력한다. 함수 형태가  $\max(0, x)$ 로 계산이 간단해 학습의 속도가 매우 빠르다. 그리고 기울기가 1이라 앞서 발생했던 gradient vanishing도 해결할 수 있어 사용했다. 그러나 음의 값을 가지면 전부 0을 출력하기 때문에, 몇몇 weight들이 업데이트 되지 않는 문제가 발생했다.

그래서 ReLU의 음의 값에 대하여 exponential 값이 추가된 함수로 Dying ReLU 현상을 해결하기 위해 ELU함수를 사용했다.

마지막으로 사용한 함수는 softmax함수인데, 이것은 다중 분류 로지스틱 회귀 모델에 사용한다. softmax함수는 입력값을 확률 척도로 보일 수 있어 라벨과 결과를 비교하기에 용이해서 사용했다.

KNN Algorithm을 구현해 Neural Network와 비교해보았다. KNN Algorithm에서는 총 3번의 test를 진행했다. train\_test\_split을 이용해서 학습셋과 데이터셋을 분류해 사용했는데, Test1에서는 학습셋과 데이터셋을 9:1로 분류해 사용했고, Test2에서는 8:2로, Test3에서는 7:3으로 분류해 사용하였다. 이후 KNN Algorithm의 객체를 생성하고 train 집단으로 KNN Algorithm을 학습시켜서 학습 점수와 테스트 점수를 얻어내 모델의 성과를 측정했다. 그리고 predict와 answer을 비교하는 채점표를 얻어냈다.

## 2) 실험 결과

Neural Network의 Test1 정확도는 1.0, Test2의 정확도는 0.96667, Test3의 정확도는 0.9778로 결과가 나왔다.

KNN Algorithm의 Test1 학습 점수는 0.96296296..., 테스트 점수는 1.0이 나왔다. Test2의 학습 점수는 0.99166666..., 테스트 점수는 0.86666666...이 나왔다. Test3의 학습 점수는 0.98095238..., 테스트 점수는 0.95555555...가 나왔다.

## <Neural Network 실험 결과>

```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
<Test>
Model: "sequential"

Layer (type) Output Shape Param #
-----
dense (Dense) (None, 64) 320
dense_1 (Dense) (None, 64) 4160
dense_2 (Dense) (None, 64) 4160
dense_3 (Dense) (None, 64) 4160
dense_4 (Dense) (None, 3) 195

Total params: 12,995
Trainable params: 12,995
Non-trainable params: 0

WARNING:tensorflow: Falling back from v2 loop because of error: Failed to find data adapter that can handle input: <class 'pandas.core.frame.DataFrame'>, <class 'NoneType'>
Train on 135 samples, validate on 15 samples
Epoch 1/50
 32/135 [=====>] - ETA: 0s - loss: 1.0096 - accuracy: 0.3750 64/135 [=====>] - ETA: 0s - loss: 1.0096 - accuracy: 0.3750
(0.46 - accuracy: 0.4148 - val_loss: 1.0879 - val_accuracy: 0.5333)
Epoch 2/50
 32/135 [=====>] - ETA: 0s - loss: 1.0068 - accuracy: 0.7500 64/135 [=====>] - ETA: 0s - loss: 1.0068 - accuracy: 0.7500
(0.9955 - accuracy: 0.6889 - val_loss: 0.9432 - val_accuracy: 0.7333)
Epoch 3/50
 32/135 [=====>] - ETA: 0s - loss: 1.0418 - accuracy: 0.5000 64/135 [=====>] - ETA: 0s - loss: 1.0418 - accuracy: 0.5000
(0.6400 - accuracy: 0.6300 - val_loss: 0.9443 - accuracy: 0.6593 - val_loss: 0.8815 - val_accuracy: 0.7333)
Epoch 4/50
 32/135 [=====>] - ETA: 0s - loss: 0.8870 - accuracy: 0.6250 64/135 [=====>] - ETA: 0s - loss: 0.8870 - accuracy: 0.6250
(0.6874 - accuracy: 0.6874 - val_loss: 0.8815 - val_loss: 0.8043 - val_accuracy: 0.6867)
Epoch 5/50
 32/135 [=====>] - ETA: 0s - loss: 0.7929 - accuracy: 0.8438 64/135 [=====>] - ETA: 0s - loss: 0.7929 - accuracy: 0.8438
(0.8281 - accuracy: 0.8281 - val_loss: 0.8074 - val_loss: 0.7333 - val_accuracy: 1.0000)
Epoch 6/50
 32/135 [=====>] - ETA: 0s - loss: 0.6960 - accuracy: 0.9688 64/135 [=====>] - ETA: 0s - loss: 0.6960 - accuracy: 0.9688
(0.8900 - accuracy: 0.8900 - val_loss: 0.6960 - val_loss: 0.6960 - val_accuracy: 0.8900)
Epoch 7/50
 32/135 [=====>] - ETA: 0s - loss: 0.5589 - accuracy: 0.9125 64/135 [=====>] - ETA: 0s - loss: 0.5589 - accuracy: 0.9125
(0.7815 - accuracy: 0.7815 - val_loss: 0.5589 - val_loss: 0.5589 - val_accuracy: 0.8125)
Epoch 8/50
 32/135 [=====>] - ETA: 0s - loss: 0.5693 - accuracy: 0.8750 64/135 [=====>] - ETA: 0s - loss: 0.5693 - accuracy: 0.8750
(0.8438 - accuracy: 0.8438 - val_loss: 0.5693 - val_loss: 0.5693 - val_accuracy: 0.8438)
Epoch 9/50
 32/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750 64/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750
(0.5171 - accuracy: 0.5171 - val_loss: 0.5080 - val_accuracy: 0.9333)
Epoch 10/50
 32/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750 64/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750
(0.5080 - accuracy: 0.5080 - val_loss: 0.5080 - val_accuracy: 0.9333)
Epoch 11/50
 32/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750 64/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750
(0.5080 - accuracy: 0.5080 - val_loss: 0.5080 - val_accuracy: 0.9333)
Epoch 12/50
 32/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750 64/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750
(0.5080 - accuracy: 0.5080 - val_loss: 0.5080 - val_accuracy: 0.9333)
Epoch 13/50
 32/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750 64/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750
(0.5080 - accuracy: 0.5080 - val_loss: 0.5080 - val_accuracy: 0.9333)
Epoch 14/50
 32/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750 64/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750
(0.5080 - accuracy: 0.5080 - val_loss: 0.5080 - val_accuracy: 0.9333)
Epoch 15/50
 32/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750 64/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750
(0.5080 - accuracy: 0.5080 - val_loss: 0.5080 - val_accuracy: 0.9333)
Epoch 16/50
 32/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750 64/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750
(0.5080 - accuracy: 0.5080 - val_loss: 0.5080 - val_accuracy: 0.9333)
Epoch 17/50
 32/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750 64/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750
(0.5080 - accuracy: 0.5080 - val_loss: 0.5080 - val_accuracy: 0.9333)
Epoch 18/50
 32/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750 64/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750
(0.5080 - accuracy: 0.5080 - val_loss: 0.5080 - val_accuracy: 0.9333)
Epoch 19/50
 32/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750 64/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750
(0.5080 - accuracy: 0.5080 - val_loss: 0.5080 - val_accuracy: 0.9333)
Epoch 20/50
 32/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750 64/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750
(0.5080 - accuracy: 0.5080 - val_loss: 0.5080 - val_accuracy: 0.9333)
Epoch 21/50
 32/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750 64/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750
(0.5080 - accuracy: 0.5080 - val_loss: 0.5080 - val_accuracy: 0.9333)
Epoch 22/50
 32/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750 64/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750
(0.5080 - accuracy: 0.5080 - val_loss: 0.5080 - val_accuracy: 0.9333)
Epoch 23/50
 32/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750 64/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750
(0.5080 - accuracy: 0.5080 - val_loss: 0.5080 - val_accuracy: 0.9333)
Epoch 24/50
 32/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750 64/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750
(0.5080 - accuracy: 0.5080 - val_loss: 0.5080 - val_accuracy: 0.9333)
Epoch 25/50
 32/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750 64/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750
(0.5080 - accuracy: 0.5080 - val_loss: 0.5080 - val_accuracy: 0.9333)
Epoch 26/50
 32/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750 64/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750
(0.5080 - accuracy: 0.5080 - val_loss: 0.5080 - val_accuracy: 0.9333)
Epoch 27/50
 32/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750 64/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750
(0.5080 - accuracy: 0.5080 - val_loss: 0.5080 - val_accuracy: 0.9333)
Epoch 28/50
 32/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750 64/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750
(0.5080 - accuracy: 0.5080 - val_loss: 0.5080 - val_accuracy: 0.9333)
Epoch 29/50
 32/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750 64/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750
(0.5080 - accuracy: 0.5080 - val_loss: 0.5080 - val_accuracy: 0.9333)
Epoch 30/50
 32/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.8750 64/135 [=====>] - ETA: 0s - loss: 0.5590 - accuracy: 0.875
```

[illegible]

```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Accuracy = 1.00000

<Test2>
Model: "sequential"
Layer (type) Output Shape Param #
dense (Dense) (None, 64) 320
dense_1 (Dense) (None, 64) 4160
dense_2 (Dense) (None, 64) 4160
dense_3 (Dense) (None, 64) 4160
dense_4 (Dense) (None, 3) 195
Total params: 12,995
Trainable params: 12,995
Non-trainable params: 0

WARNING:tensorflow: Falling back from v2 loop because of error: Failed to find data adapter that can handle input: <class 'pandas.core.frame.DataFrame'>, <class 'NoneType'>
Train on 120 samples, validate on 30 samples
Epoch 1/50
32/120 [=====] - ETA: 0s - loss: 0.0446 - accuracy: 1.0000 96/120 [=====] - ETA: 0s - loss: 0.0815 - accuracy: 0.9794 64/120 [=====] - ETA: 0s - loss: 0.0989
accuracy: 0.9844 120/120 [=====] - 1s 12ms/sample - loss: 0.0540 - accuracy: 0.9833 - val_loss: 0.0393 - val_accuracy: 0.9667
Epoch 2/50
32/120 [=====] - ETA: 0s - loss: 0.0318 - accuracy: 1.0000 96/120 [=====] - ETA: 0s - loss: 0.0815 - accuracy: 0.9794 64/120 [=====] - ETA: 0s - loss: 0.0593
accuracy: 0.9844 120/120 [=====] - 1s 12ms/sample - loss: 0.0670 - accuracy: 0.9667 - val_loss: 0.0491 - val_accuracy: 0.9667
Epoch 3/50
32/120 [=====] - ETA: 0s - loss: 0.0304 - accuracy: 1.0000 96/120 [=====] - ETA: 0s - loss: 0.0815 - accuracy: 0.9794 64/120 [=====] - ETA: 0s - loss: 0.0921
accuracy: 0.9531 120/120 [=====] - 1s 12ms/sample - loss: 0.0652 - accuracy: 0.9500 - val_loss: 0.0718 - val_accuracy: 0.9667
Epoch 4/50
32/120 [=====] - ETA: 0s - loss: 0.0414 - accuracy: 1.0000 96/120 [=====] - ETA: 0s - loss: 0.0815 - accuracy: 0.9794 64/120 [=====] - ETA: 0s - loss: 0.0390
accuracy: 1.0000 120/120 [=====] - 1s 12ms/sample - loss: 0.1065 - accuracy: 0.9750 - val_loss: 0.0374 - val_accuracy: 0.9667
Epoch 5/50
32/120 [=====] - ETA: 0s - loss: 0.0556 - accuracy: 0.9667 96/120 [=====] - ETA: 0s - loss: 0.0815 - accuracy: 0.9794 64/120 [=====] - ETA: 0s - loss: 0.0984
accuracy: 0.9531 120/120 [=====] - 1s 12ms/sample - loss: 0.1480 - accuracy: 0.9333 - val_loss: 0.0324 - val_accuracy: 1.0000
Epoch 6/50
32/120 [=====] - ETA: 0s - loss: 0.0708 - accuracy: 0.9333 96/120 [=====] - ETA: 0s - loss: 0.0815 - accuracy: 0.9794 64/120 [=====] - ETA: 0s - loss: 0.0476
accuracy: 0.9531 120/120 [=====] - 1s 12ms/sample - loss: 0.0699 - accuracy: 0.9633 - val_loss: 0.0181 - val_accuracy: 0.9333
Epoch 7/50
32/120 [=====] - ETA: 0s - loss: 0.0593 - accuracy: 0.9375 96/120 [=====] - ETA: 0s - loss: 0.0815 - accuracy: 0.9794 64/120 [=====] - ETA: 0s - loss: 0.0913
accuracy: 0.9531 120/120 [=====] - 1s 12ms/sample - loss: 0.1130 - accuracy: 0.9500 - val_loss: 0.0391 - val_accuracy: 0.9667
Epoch 8/50
32/120 [=====] - ETA: 0s - loss: 0.1021 - accuracy: 1.0000 96/120 [=====] - ETA: 0s - loss: 0.0815 - accuracy: 0.9794 64/120 [=====] - ETA: 0s - loss: 0.1051
accuracy: 0.9531 120/120 [=====] - 1s 12ms/sample - loss: 0.1648 - accuracy: 0.9250 - val_loss: 0.0451 - val_accuracy: 0.9667
Epoch 9/50
32/120 [=====] - ETA: 0s - loss: 0.0436 - accuracy: 0.9667 96/120 [=====] - ETA: 0s - loss: 0.0815 - accuracy: 0.9794 64/120 [=====] - ETA: 0s - loss: 0.0971
accuracy: 0.9667 120/120 [=====] - 1s 12ms/sample - loss: 0.1002 - accuracy: 0.9667 - val_loss: 0.0906 - val_accuracy: 0.9667
Epoch 10/50
32/120 [=====] - ETA: 0s - loss: 0.0540 - accuracy: 0.9667 96/120 [=====] - ETA: 0s - loss: 0.0815 - accuracy: 0.9794 64/120 [=====] - ETA: 0s - loss: 0.0989
accuracy: 0.9844 120/120 [=====] - 1s 12ms/sample - loss: 0.0540 - accuracy: 0.9833 - val_loss: 0.0393 - val_accuracy: 0.9667
Epoch 11/50
32/120 [=====] - ETA: 0s - loss: 0.0054 - accuracy: 1.0000 96/120 [=====] - ETA: 0s - loss: 0.0815 - accuracy: 0.9794 64/120 [=====] - ETA: 0s - loss: 0.0530
accuracy: 0.9844 120/120 [=====] - 1s 12ms/sample - loss: 0.0524 - accuracy: 0.9833 - val_loss: 0.0490 - val_accuracy: 0.9667
Epoch 12/50
32/120 [=====] - ETA: 0s - loss: 0.0340 - accuracy: 1.0000 96/120 [=====] - ETA: 0s - loss: 0.0815 - accuracy: 0.9794 64/120 [=====] - ETA: 0s - loss: 0.0818
accuracy: 0.9667 120/120 [=====] - 1s 12ms/sample - loss: 0.0539 - accuracy: 0.9833 - val_loss: 0.0337 - val_accuracy: 0.9667
Epoch 13/50
32/120 [=====] - ETA: 0s - loss: 0.0377 - accuracy: 1.0000 96/120 [=====] - ETA: 0s - loss: 0.0815 - accuracy: 0.9794 64/120 [=====] - ETA: 0s - loss: 0.0544
accuracy: 0.9667 120/120 [=====] - 1s 12ms/sample - loss: 0.0565 - accuracy: 0.9750 - val_loss: 0.0391 - val_accuracy: 0.9667
Epoch 14/50
32/120 [=====] - ETA: 0s - loss: 0.0317 - accuracy: 1.0000 96/120 [=====] - ETA: 0s - loss: 0.0815 - accuracy: 0.9794 64/120 [=====] - ETA: 0s - loss: 0.0276
accuracy: 1.0000 120/120 [=====] - 1s 12ms/sample - loss: 0.0511 - accuracy: 0.9750 - val_loss: 0.0542 - val_accuracy: 0.9667
Epoch 15/50
32/120 [=====] - ETA: 0s - loss: 0.0910 - accuracy: 0.9333 96/120 [=====] - ETA: 0s - loss: 0.0815 - accuracy: 0.9794 64/120 [=====] - ETA: 0s - loss: 0.0704
accuracy: 0.9667 120/120 [=====] - 1s 12ms/sample - loss: 0.0485 - accuracy: 0.9833 - val_loss: 0.0598 - val_accuracy: 0.9667
Epoch 16/50
32/120 [=====] - ETA: 0s - loss: 0.0371 - accuracy: 0.9667 96/120 [=====] - ETA: 0s - loss: 0.0815 - accuracy: 0.9794 64/120 [=====] - ETA: 0s - loss: 0.0462
accuracy: 0.9844 120/120 [=====] - 1s 12ms/sample - loss: 0.0486 - accuracy: 0.9833 - val_loss: 0.0392 - val_accuracy: 0.9667
Epoch 17/50
32/120 [=====] - ETA: 0s - loss: 0.0387 - accuracy: 1.0000 96/120 [=====] - ETA: 0s - loss: 0.0815 - accuracy: 0.9794 64/120 [=====] - ETA: 0s - loss: 0.0340
accuracy: 0.9844 120/120 [=====] - 1s 12ms/sample - loss: 0.0557 - accuracy: 0.9750 - val_loss: 0.0485 - val_accuracy: 0.9667
Epoch 18/50
32/120 [=====] - ETA: 0s - loss: 0.0257 - accuracy: 1.0000 96/120 [=====] - ETA: 0s - loss: 0.0815 - accuracy: 0.9794 64/120 [=====] - ETA: 0s - loss: 0.0468
accuracy: 0.9844 120/120 [=====] - 1s 12ms/sample - loss: 0.0621 - accuracy: 0.9667 - val_loss: 0.0422 - val_accuracy: 0.9667
Epoch 19/50
32/120 [=====] - ETA: 0s - loss: 0.0313 - accuracy: 1.0000 96/120 [=====] - ETA: 0s - loss: 0.0815 - accuracy: 0.9794 64/120 [=====] - ETA: 0s - loss: 0.0395
accuracy: 0.9844 120/120 [=====] - 1s 12ms/sample - loss: 0.0804 - accuracy: 0.9750 - val_loss: 0.0680 - val_accuracy: 0.9667
Epoch 20/50
32/120 [=====] - ETA: 0s - loss: 0.0235 - accuracy: 1.0000 96/120 [=====] - ETA: 0s - loss: 0.0815 - accuracy: 0.9794 64/120 [=====] - ETA: 0s - loss: 0.0333
accuracy: 0.9844 120/120 [=====] - 1s 12ms/sample - loss: 0.0562 - accuracy: 0.9667 - val_loss: 0.0399 - val_accuracy: 0.9667
Epoch 21/50
32/120 [=====] - ETA: 0s - loss: 0.0308 - accuracy: 1.0000 96/120 [=====] - ETA: 0s - loss: 0.0815 - accuracy: 0.9794 64/120 [=====] - ETA: 0s - loss: 0.0574
accuracy: 0.9531 120/120 [=====] - 1s 12ms/sample - loss: 0.0515 - accuracy: 0.9583 - val_loss: 0.0672 - val_accuracy: 0.9667
Epoch 22/50
32/120 [=====] - ETA: 0s - loss: 0.0536 - accuracy: 0.9667 96/120 [=====] - ETA: 0s - loss: 0.0815 - accuracy: 0.9794 64/120 [=====] - ETA: 0s - loss: 0.0515
accuracy: 0.9667 120/120 [=====] - 1s 12ms/sample - loss: 0.0632 - accuracy: 0.9750 - val_loss: 0.0979 - val_accuracy: 0.9667
WARNING:tensorflow: Falling back from v2 loop because of error: Failed to find data adapter that can handle input: <class 'pandas.core.frame.DataFrame'>, <class 'NoneType'>
Accuracy = 0.9667

<Test3>
Model: "sequential"
Layer (type) Output Shape Param #
```

```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Accuracy = 0.9667

<Test3>
Model: "sequential"
Layer (type) Output Shape Param #
```

```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help

WARNING:tensorflow: Falling back from v2 loop because of error: Failed to find data adapter that can handle input: <class 'pandas.core.frame.DataFrame'>, <class 'NoneType'>
2019/12/20 14:20:00 - Os 0s/sample - loss: 0.0379 - accuracy: 0.9667
Accuracy = 0.9667

<Test>
Model: "sequential"
Layer (type) Output Shape Param #
dense (Dense) (None, 64) 320
dense_1 (Dense) (None, 64) 4160
dense_2 (Dense) (None, 64) 4160
dense_3 (Dense) (None, 64) 4160
dense_4 (Dense) (None, 3) 195
Total params: 12,995
Trainable params: 12,995
Nontrainable params: 0

WARNING:tensorflow: Falling back from v2 loop because of error: Failed to find data adapter that can handle input: <class 'pandas.core.frame.DataFrame'>, <class 'NoneType'>
Train on 105 samples, validate on 45 samples
Epoch 1/50
32/105 [=====] 1 - ETA: 0s - loss: 0.1290 - accuracy: 0.9688 64/105 [=====] 1 - ETA: 0s - loss: 0.0706 - accuracy: 0.9894
105/105 [=====] 1s 8ms/sample - loss: 0.0510 - accuracy: 0.9905 - val_loss: 0.0639 - val_accuracy: 0.9556
Epoch 2/50
32/105 [=====] 1 - ETA: 0s - loss: 0.0078 - accuracy: 1.0000 64/105 [=====] 1 - ETA: 0s - loss: 0.0339 - accuracy: 0.9564
105/105 [=====] 1s 9ms/sample - loss: 0.0078 - accuracy: 0.9905 - val_loss: 0.0702 - val_accuracy: 0.9778
Epoch 3/50
32/105 [=====] 1 - ETA: 0s - loss: 0.0084 - accuracy: 1.0000 64/105 [=====] 1 - ETA: 0s - loss: 0.0148 - accuracy: 0.9894
105/105 [=====] 1s 10ms/sample - loss: 0.1506 - accuracy: 0.9619 - val_loss: 0.1233 - val_accuracy: 0.9556
Epoch 4/50
32/105 [=====] 1 - ETA: 0s - loss: 0.0246 - accuracy: 1.0000 64/105 [=====] 1 - ETA: 0s - loss: 0.0200 - accuracy: 0.9794
105/105 [=====] 1s 11ms/sample - loss: 0.0795 - accuracy: 0.9714 - val_loss: 0.2768 - val_accuracy: 0.9333
Epoch 5/50
32/105 [=====] 1 - ETA: 0s - loss: 0.2187 - accuracy: 0.9063 64/105 [=====] 1 - ETA: 0s - loss: 0.2161 - accuracy: 0.9214
105/105 [=====] 1s 13ms/sample - loss: 0.1462 - accuracy: 0.9333 - val_loss: 0.1541 - val_accuracy: 0.9556
Epoch 6/50
32/105 [=====] 1 - ETA: 0s - loss: 0.0183 - accuracy: 1.0000 64/105 [=====] 1 - ETA: 0s - loss: 0.1440 - accuracy: 0.9478
105/105 [=====] 1s 14ms/sample - loss: 0.1342 - accuracy: 0.9524 - val_loss: 0.1852 - val_accuracy: 0.9333
Epoch 7/50
32/105 [=====] 1 - ETA: 0s - loss: 0.0292 - accuracy: 1.0000 64/105 [=====] 1 - ETA: 0s - loss: 0.0863 - accuracy: 0.9794
105/105 [=====] 1s 14ms/sample - loss: 0.0688 - accuracy: 0.9810 - val_loss: 0.0799 - val_accuracy: 0.9556
Epoch 8/50
32/105 [=====] 1 - ETA: 0s - loss: 0.0391 - accuracy: 0.9063 64/105 [=====] 1 - ETA: 0s - loss: 0.1041 - accuracy: 0.9378
105/105 [=====] 1s 14ms/sample - loss: 0.0748 - accuracy: 0.9429 - val_loss: 0.0890 - val_accuracy: 0.9556
Epoch 9/50
32/105 [=====] 1 - ETA: 0s - loss: 0.0149 - accuracy: 1.0000 64/105 [=====] 1 - ETA: 0s - loss: 0.0963 - accuracy: 0.9894
105/105 [=====] 1s 14ms/sample - loss: 0.0608 - accuracy: 0.9619 - val_loss: 0.0614 - val_accuracy: 0.9556
Epoch 10/50
32/105 [=====] 1 - ETA: 0s - loss: 0.0753 - accuracy: 0.9688 64/105 [=====] 1 - ETA: 0s - loss: 0.0411 - accuracy: 0.9794
105/105 [=====] 1s 12ms/sample - loss: 0.0508 - accuracy: 0.9619 - val_loss: 0.0614 - val_accuracy: 0.9556
Epoch 11/50
32/105 [=====] 1 - ETA: 0s - loss: 0.0092 - accuracy: 1.0000 64/105 [=====] 1 - ETA: 0s - loss: 0.0591 - accuracy: 0.9794
105/105 [=====] 1s 12ms/sample - loss: 0.0516 - accuracy: 0.9810 - val_loss: 0.0673 - val_accuracy: 0.9778
Epoch 12/50
32/105 [=====] 1 - ETA: 0s - loss: 0.0193 - accuracy: 1.0000 64/105 [=====] 1 - ETA: 0s - loss: 0.0441 - accuracy: 0.9894
105/105 [=====] 1s 12ms/sample - loss: 0.0430 - accuracy: 0.9905 - val_loss: 0.0635 - val_accuracy: 0.9778
Epoch 13/50
32/105 [=====] 1 - ETA: 0s - loss: 0.0639 - accuracy: 1.0000 64/105 [=====] 1 - ETA: 0s - loss: 0.0320 - accuracy: 0.9894
105/105 [=====] 1s 11ms/sample - loss: 0.0431 - accuracy: 0.9905 - val_loss: 0.0605 - val_accuracy: 0.9556
Epoch 14/50
32/105 [=====] 1 - ETA: 0s - loss: 0.0777 - accuracy: 0.9688 64/105 [=====] 1 - ETA: 0s - loss: 0.0428 - accuracy: 0.9794
105/105 [=====] 1s 12ms/sample - loss: 0.0430 - accuracy: 0.9905 - val_loss: 0.0634 - val_accuracy: 0.9556
Epoch 15/50
32/105 [=====] 1 - ETA: 0s - loss: 0.0156 - accuracy: 1.0000 64/105 [=====] 1 - ETA: 0s - loss: 0.0698 - accuracy: 0.9894
105/105 [=====] 1s 11ms/sample - loss: 0.0449 - accuracy: 0.9810 - val_loss: 0.0611 - val_accuracy: 0.9556
Epoch 16/50
32/105 [=====] 1 - ETA: 0s - loss: 0.0082 - accuracy: 1.0000 64/105 [=====] 1 - ETA: 0s - loss: 0.0299 - accuracy: 0.9894
105/105 [=====] 1s 12ms/sample - loss: 0.0396 - accuracy: 0.9905 - val_loss: 0.0594 - val_accuracy: 0.9556
Epoch 17/50
32/105 [=====] 1 - ETA: 0s - loss: 0.0761 - accuracy: 0.9688 64/105 [=====] 1 - ETA: 0s - loss: 0.0454 - accuracy: 0.9894
105/105 [=====] 1s 11ms/sample - loss: 0.0447 - accuracy: 0.9905 - val_loss: 0.0599 - val_accuracy: 0.9556
Epoch 18/50
32/105 [=====] 1 - ETA: 0s - loss: 0.0208 - accuracy: 1.0000 64/105 [=====] 1 - ETA: 0s - loss: 0.0535 - accuracy: 0.9894
105/105 [=====] 1s 12ms/sample - loss: 0.0424 - accuracy: 0.9905 - val_loss: 0.0799 - val_accuracy: 0.9556
Epoch 19/50
32/105 [=====] 1 - ETA: 0s - loss: 0.0134 - accuracy: 1.0000 64/105 [=====] 1 - ETA: 0s - loss: 0.0562 - accuracy: 0.9794
105/105 [=====] 1s 12ms/sample - loss: 0.0510 - accuracy: 0.9810 - val_loss: 0.0722 - val_accuracy: 0.9556
Epoch 20/50
32/105 [=====] 1 - ETA: 0s - loss: 0.0408 - accuracy: 0.9688 64/105 [=====] 1 - ETA: 0s - loss: 0.0238 - accuracy: 0.9794
105/105 [=====] 1s 12ms/sample - loss: 0.0424 - accuracy: 0.9810 - val_loss: 0.0598 - val_accuracy: 0.9778
Epoch 21/50
32/105 [=====] 1 - ETA: 0s - loss: 0.0224 - accuracy: 1.0000 64/105 [=====] 1 - ETA: 0s - loss: 0.0136 - accuracy: 0.9894
105/105 [=====] 1s 11ms/sample - loss: 0.0399 - accuracy: 0.9905 - val_loss: 0.0592 - val_accuracy: 0.9778
Epoch 22/50
32/105 [=====] 1 - ETA: 0s - loss: 0.0511 - accuracy: 0.9688 64/105 [=====] 1 - ETA: 0s - loss: 0.0569 - accuracy: 0.9894
105/105 [=====] 1s 12ms/sample - loss: 0.0429 - accuracy: 0.9905 - val_loss: 0.0620 - val_accuracy: 0.9778
WARNING:tensorflow: Falling back from v2 loop because of error: Failed to find data adapter that can handle input: <class 'pandas.core.frame.DataFrame'>, <class 'NoneType'>
2019/12/20 14:20:00 - Os 5ms/sample - loss: 0.0620
Accuracy = 0.9778
>>>
```

## <KNN실험 결과>

<Test1>

|  
학습 점수: 0.9629629629629629

테스트 점수: 1.0

채점표:

	predict	answer
50	versicolor	versicolor
145	virginica	virginica
86	versicolor	versicolor
118	virginica	virginica
122	virginica	virginica
57	versicolor	versicolor
56	versicolor	versicolor
51	versicolor	versicolor
69	versicolor	versicolor
66	versicolor	versicolor
59	versicolor	versicolor
4	setosa	setosa
110	virginica	virginica
94	versicolor	versicolor
29	setosa	setosa

<Test2>

학습 점수: 0.9916666666666667

테스트 점수: 0.8666666666666667

채점표:

	predict	answer
43	setosa	setosa
38	setosa	setosa
97	versicolor	versicolor
32	setosa	setosa
93	versicolor	versicolor
6	setosa	setosa
2	setosa	setosa
63	virginica	versicolor
51	versicolor	versicolor
73	versicolor	versicolor
74	versicolor	versicolor
139	virginica	virginica
131	virginica	virginica
54	versicolor	versicolor
83	virginica	versicolor
71	versicolor	versicolor
103	virginica	virginica
61	versicolor	versicolor
17	setosa	setosa
68	virginica	versicolor
3	setosa	setosa
72	virginica	versicolor
79	versicolor	versicolor
48	setosa	setosa
9	setosa	setosa
108	virginica	virginica
50	versicolor	versicolor
109	virginica	virginica
86	versicolor	versicolor
149	virginica	virginica

<Test3>

학습 점수: 0.9809523809523809

테스트 점수: 0.9555555555555556

채점표:

	predict	answer			
46	setosa	setosa			
81	versicolor	versicolor			
32	setosa	setosa			
102	virginica	virginica			
104	virginica	virginica			
55	versicolor	versicolor			
13	setosa	setosa			
72	virginica	versicolor			
61	versicolor	versicolor			
77	virginica	versicolor			
33	setosa	setosa			
110	virginica	virginica			
87	versicolor	versicolor			
85	versicolor	versicolor			
18	setosa	setosa			
12	setosa	setosa	143	virginica	virginica
47	setosa	setosa	127	virginica	virginica
125	virginica	virginica	74	versicolor	versicolor
90	versicolor	versicolor	148	virginica	virginica
22	setosa	setosa	91	versicolor	versicolor
138	virginica	virginica	0	setosa	setosa
5	setosa	setosa	16	setosa	setosa
1	setosa	setosa	41	setosa	setosa
49	setosa	setosa	82	versicolor	versicolor
19	setosa	setosa	100	virginica	virginica
21	setosa	setosa	141	virginica	virginica
117	virginica	virginica	144	virginica	virginica
76	versicolor	versicolor	120	virginica	virginica
23	setosa	setosa	29	setosa	setosa
50	versicolor	versicolor	10	setosa	setosa

### [3] 실험에 대한 결론(100어절 이내)

Neural Network, KNN Algorithm 모두 9:1에서 7:3으로 비율을 조정할수록 Neural NetWork와 KNN의 테스트 점수가 떨어지는 경향이 보인다. 이것은 더 많은 비율을 교육시키는 데 쓰고, 테스트에 더 작은 비율을 쓰기 때문이다. 6:4로 비율을 조정한 결과, 더 낮은 정확도를 보임을 확인했다.

이번 실험에서 Layer수는 5개를 썼는데, Layer를 너무 많이 쓰면 각각의 Layer가 분류하는데 서로에서 영향을 많이 줘 적당한 수의 Layer를 써야한다는 것을 알았다. Activation function은 이번 실험에서 발전 순서대로 썼는데, 만약 역순으로 사용할 경우, 누락되는 데이터가 있을 것이라 예상했다.