Mnist : model

```python
self.main = nn.Sequential(
    nn.Conv2d(1, 16, 3, 1, 1, bias=False),
    nn.BatchNorm2d(16),
    self.choise_activate(),
    nn.MaxPool2d(2, 2),
    nn.Dropout(0.3),
    #printL,

    nn.Conv2d(16, 64, 3, 1, 1, bias=False),
    nn.BatchNorm2d(64),
    self.choise_activate(),
    nn.MaxPool2d(2, 2),
    nn.Dropout(0.3),


    nn.Flatten(),
    nn.Linear(64 * 7 * 7, 128),
    nn.Linear(128, 10),
)
```
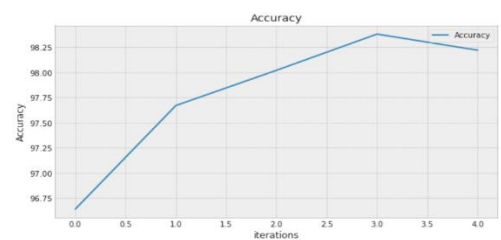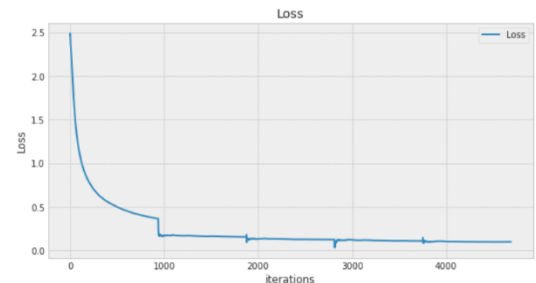
Relu

```
Epoch:5
    Batch:938 of 938, loss:0.016, MeanLoss:0.098
    Accuracy of the network on the 10000 test images: 98.210 %
    Total_time : 0.7701400597890218 Mins
```
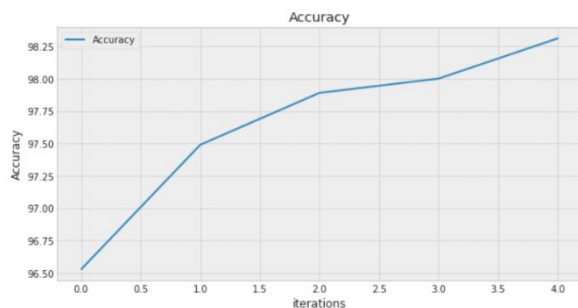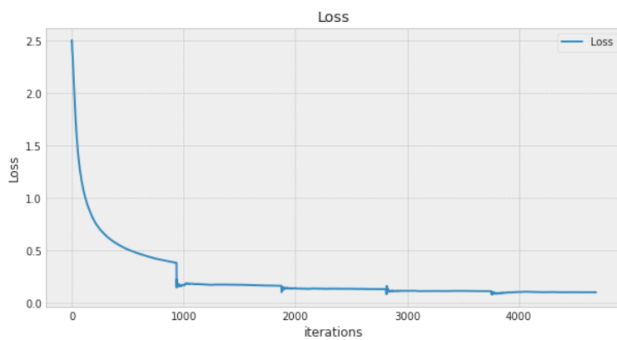

Activate function : relu

Leakyrelu

```
Epoch:5
    Batch:938 of 938, loss:0.215, MeanLoss:0.100
    Accuracy of the network on the 10000 test images: 98.310 %
    Total_time : 0.78238605260849 Mins
```
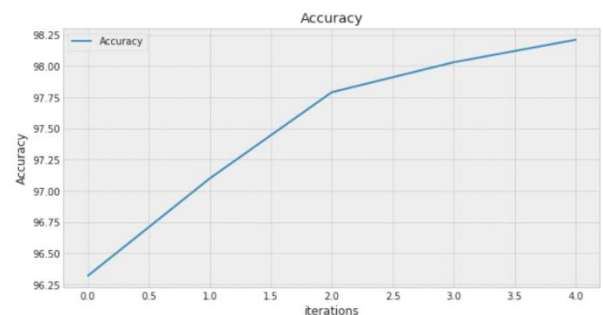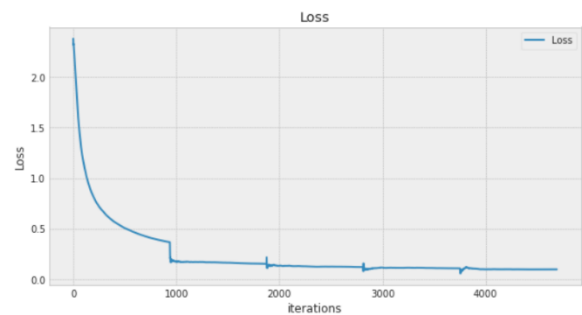

Activate function : leaky_r

ule

```
Epoch:5
    Batch:938 of 938, loss:0.016, MeanLoss:0.098
    Accuracy of the network on the 10000 test images: 98.210 %
    Total_time : 0.7701400597890218 Mins
```


Mnist 用 leakyrelu 效果最好。

KMnist

```python
self.main = nn.Sequential(
    nn.Conv2d(1, 16, 3, 1, 1, bias=False),
    nn.BatchNorm2d(16),
    self.choise_activate(),
    nn.MaxPool2d(2, 2),
    nn.Dropout(0.3),
    #printL,

    nn.Conv2d(16, 64, 3, 1, 1, bias=False),
    nn.BatchNorm2d(64),
    self.choise_activate(),
    nn.MaxPool2d(2, 2),
    nn.Dropout(0.3),


    nn.Flatten(),
    nn.Linear(64 * 7 * 7, 128),
    nn.Linear(128, 10),
)
```
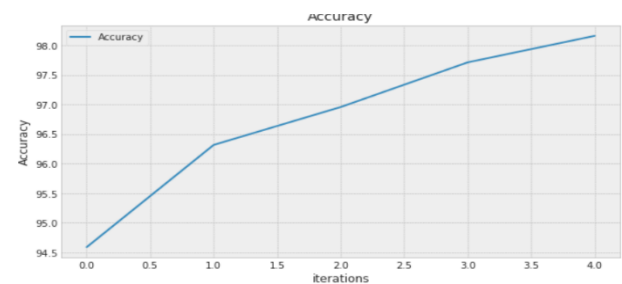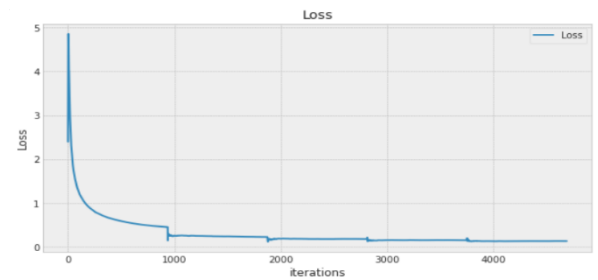
relu

```
Epoch:5
    Batch:938 of 938, loss:0.031, MeanLoss:0.131
    Accuracy of the network on the 10000 test images: 98.160 %
    Total_time : 1.1375240802764892 Mins
```
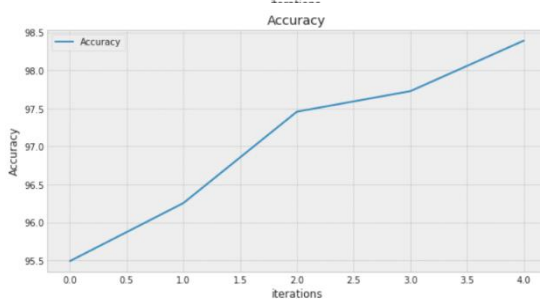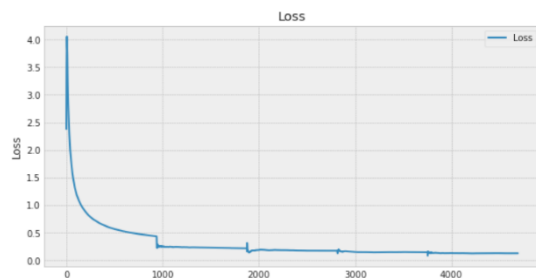

LeakyRelu

```
Epoch:5
    Batch:938 of 938, loss:0.031, MeanLoss:0.125
    Accuracy of the network on the 10000 test images: 98.392 %
    Total_time : 1.1582466284434 Mins
```
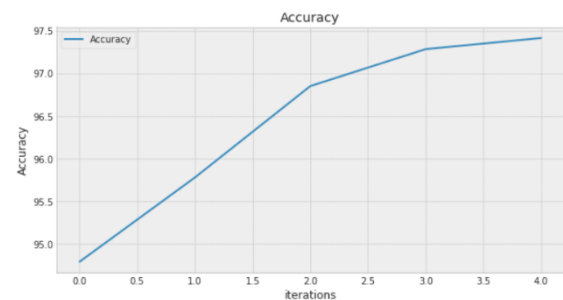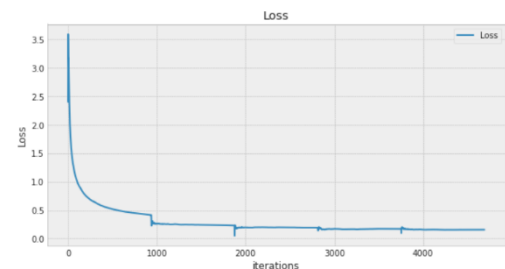


```
Epoch:5
    Batch:938 of 938, loss:0.197, MeanLoss:0.157
    Accuracy of the network on the 10000 test images: 97.413 %
    Total_time : 1.1391323804855347 Mins
```


KMnist : LeakyRelu 效果最好

```
self.main = nn.Sequential(
    nn.Conv2d(3, 128, 3, 1, 1, bias=False),
    nn.BatchNorm2d(128),
    self.choise_activate(),
    nn.MaxPool2d(2, 2),

    nn.Conv2d(128, 256, 3, 1, 1, bias=False),
    nn.BatchNorm2d(256),
    self.choise_activate(),
    nn.MaxPool2d(2, 2),
    nn.Dropout(0.3),

    self.res(256, 256, self.activate_F),
    self.choise_activate(),

    nn.Conv2d(256, 512, 3, 1, 1, bias=False),
    nn.BatchNorm2d(512),
    self.choise_activate(),
    nn.MaxPool2d(2, 2),
    nn.Dropout(0.3),

    self.res(512, 512, self.activate_F),
    self.choise_activate(),

    nn.Conv2d(512, 256, 3, 1, 1, bias=False),
    nn.BatchNorm2d(256),
    self.choise_activate(),
    nn.MaxPool2d(2, 2),
    nn.Dropout(0.3),

    self.res(256, 256, self.activate_F),
    self.choise_activate(),

    nn.Conv2d(256, 64, 3, 1, 1, bias=False),
    nn.BatchNorm2d(64),
    self.choise_activate(),
    nn.MaxPool2d(2, 2),
    #printL,

    nn.Flatten(),
    nn.Linear(64 * 6 * 6, 2048),
    nn.Linear(2048, 512),
    nn.Linear(512, 102),
```
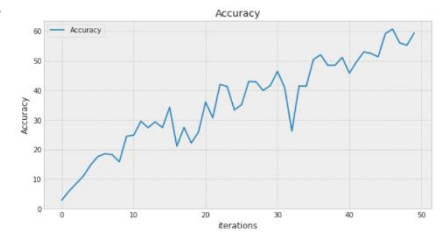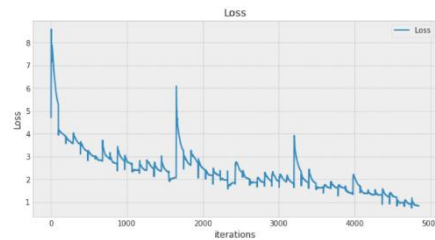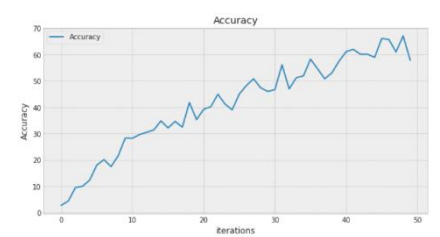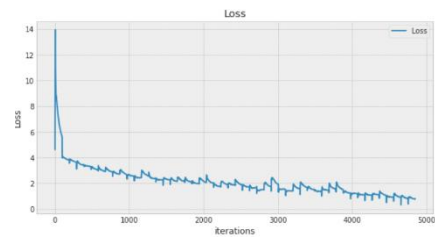
Flower102 的 model 由 CNN ,residualblock(其中包含兩層 channel 數不變的 CNN), FN 所組成，總共 14 層。

Activate function : relu

```
Epoch:50
    Batch:097 of 97, loss:0.896, MeanLoss:0.835
    Accuracy of the network on the 10000 test images: 59.314 %
    Total_time : 50.56152824163437 Mins
```
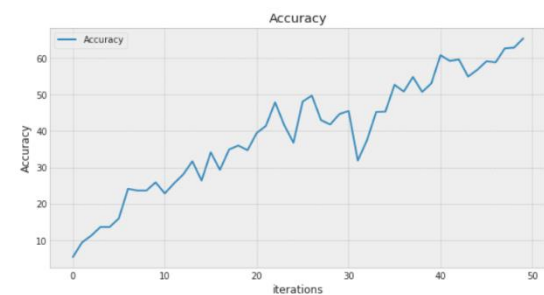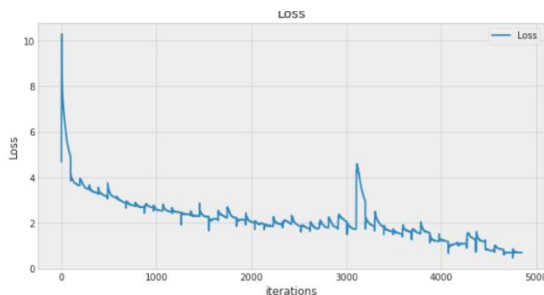

Activate function : leaky_r

```
Epoch:49
    Batch:097 of 97, loss:1.593, MeanLoss:0.812
    Accuracy of the network on the 10000 test images: 67.059 %

Epoch:50
    Batch:097 of 97, loss:3.724, MeanLoss:0.801
    Accuracy of the network on the 10000 test images: 57.843 %
    Total_time : 50.99641267061234 Mins
```


Activate function : elu

```
Epoch:50
    Batch:097 of 97, loss:0.362, MeanLoss:0.696
    Accuracy of the network on the 10000 test images: 65.392 %
    Total_time : 50.73040278752645 Mins
```

```
Epoch:200
    Batch:097 of 97, loss:0.783, MeanLoss:0.083
    Accuracy of the network on the 10000 test images: 73.039 %
    Total_time : 11589.766588926315
```

Flower102:leackyRelu 效果最好，目前最高的 accuracy 是 73.8%