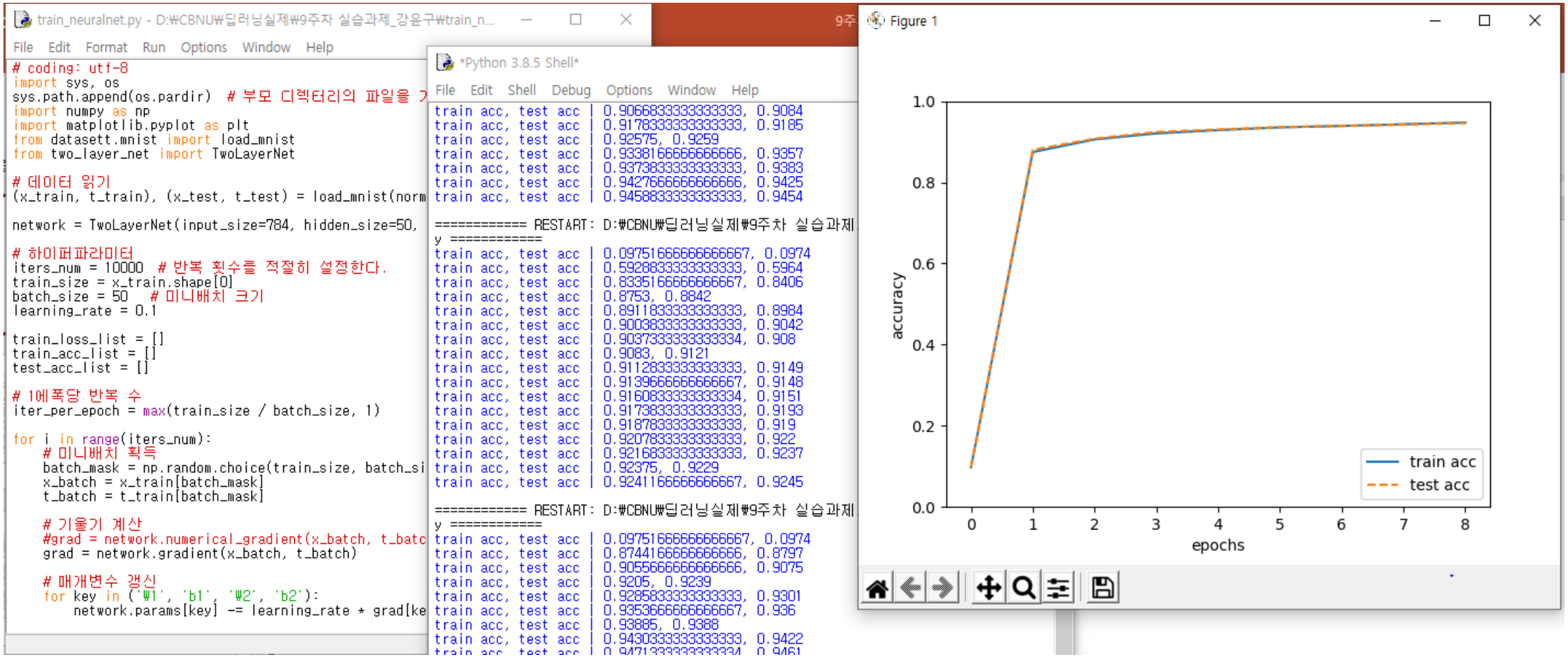
[1] Batch Size 갯수를 변화시키며, 훈련데이터와 테스트데이터의 정확도 추이 확인



batch size = 50
train acc = 0.947
test acc = 0.946

[1] Batch Size 갯수를 변화시키며, 훈련데이터와 테스트데이터의 정확도 추이 확인

train_neuralnet.py - D:₩CBNU₩딥러닝실제₩9주차 실습과제_강윤구₩train_n...

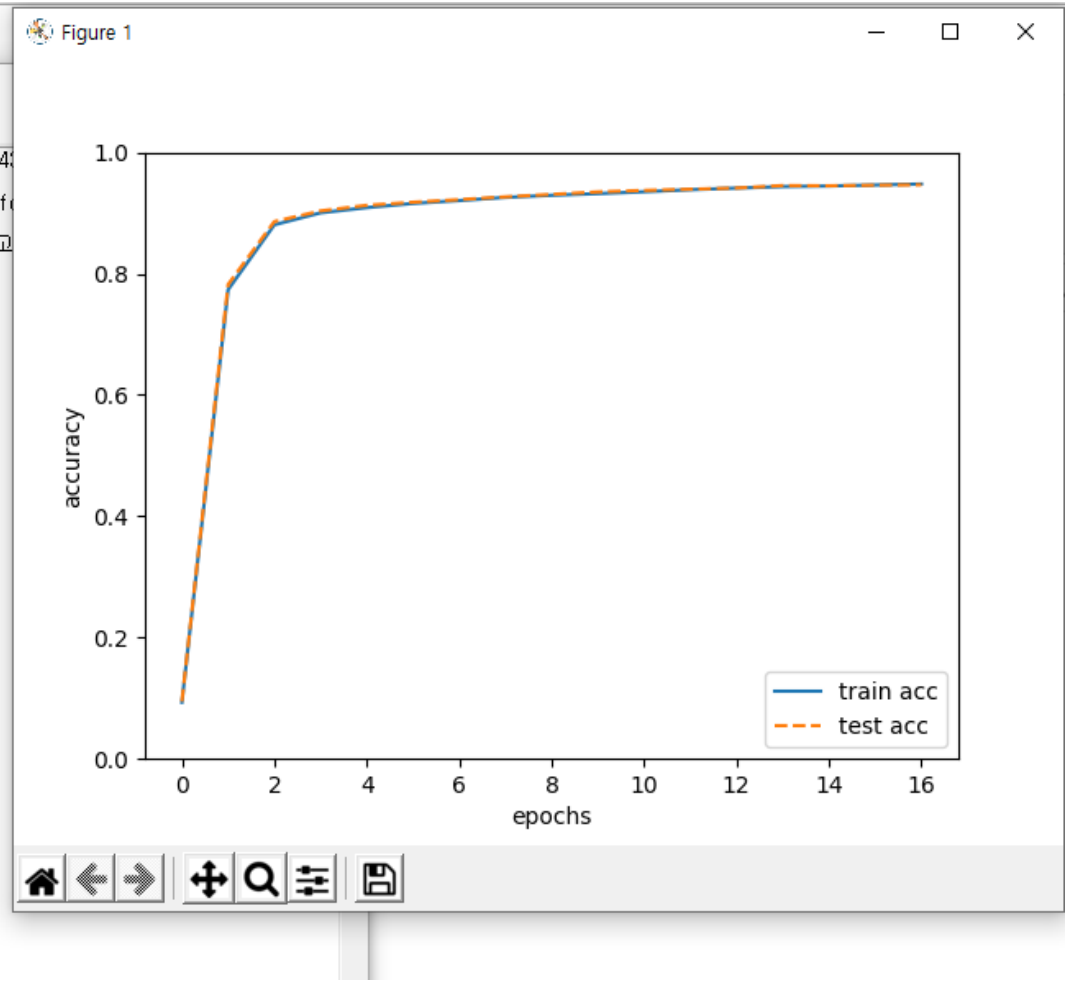
File Edit Format Run Options Window Help

```
# coding: utf-8
import sys, os
sys.path.append(os.pardir)  # 부모 디렉터리
import numpy as np
import matplotlib.pyplot as plt
from datasett.mnist import load_mnist
from two_layer_net import TwoLayerNet

# 데이터 읽기
(x_train, t_train), (x_test, t_test) = load_

network = TwoLayerNet(input_size=784, hidden

# 하이퍼파라미터
iters_num = 10000  # 반복 횟수를 적절히 설정
train_size = x_train.shape[0]
batch_size = 100   # 미니배치 크기
learning_rate = 0.1

train_loss_list = []
train_acc_list = []
test_acc_list = []

# 1에폭당 반복 수
iter_per_epoch = max(train_size / batch_size

for i in range(iters_num):
    # 미니배치 획득
    batch_mask = np.random.choice(train_size
    x_batch = x_train[batch_mask]
    t_batch = t_train[batch_mask]

    # 기울기 계산
    #grad = network.numerical_gradient(x_bat
    grad = network.gradient(x_batch, t_batch

    # 매개변수 갱신
    for key in ('W1', 'b1', 'W2', 'b2'):
        network.params[key] -= learning_rate
```

Python 3.8.5 Shell

File Edit Shell Debug Options Window Help

```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43
tel)] on win32
Type "help", "copyright", "credits" or "license()" fo
>>>
========== RESTART: D:₩CBNU₩딥러닝실제₩9주차 실습과
y ==========
train acc, test acc | 0.09311666666666667, 0.0944
train acc, test acc | 0.7735166666666666, 0.782
train acc, test acc | 0.8804833333333333, 0.8858
train acc, test acc | 0.9002, 0.9034
train acc, test acc | 0.9087833333333334, 0.9126
train acc, test acc | 0.9156333333333333, 0.9177
train acc, test acc | 0.9204, 0.922
train acc, test acc | 0.9256333333333333, 0.9266
train acc, test acc | 0.9291333333333334, 0.9306
train acc, test acc | 0.9318333333333333, 0.9347
train acc, test acc | 0.9349166666666666, 0.9372
train acc, test acc | 0.9383666666666667, 0.9391
train acc, test acc | 0.9409666666666666, 0.9411
train acc, test acc | 0.94345, 0.9452
train acc, test acc | 0.9444833333333333, 0.9448
train acc, test acc | 0.9461, 0.9458
train acc, test acc | 0.94775, 0.9464
```



batch size = 100
train acc = 0.947
test acc = 0.946

[1] Batch Size 갯수를 변화시키며, 훈련데이터와 테스트데이터의 정확도 추이 확인

train_neuralnet.py - D:₩CBNU₩딥러닝실제₩9주차 실습과제_강윤구₩train_n...

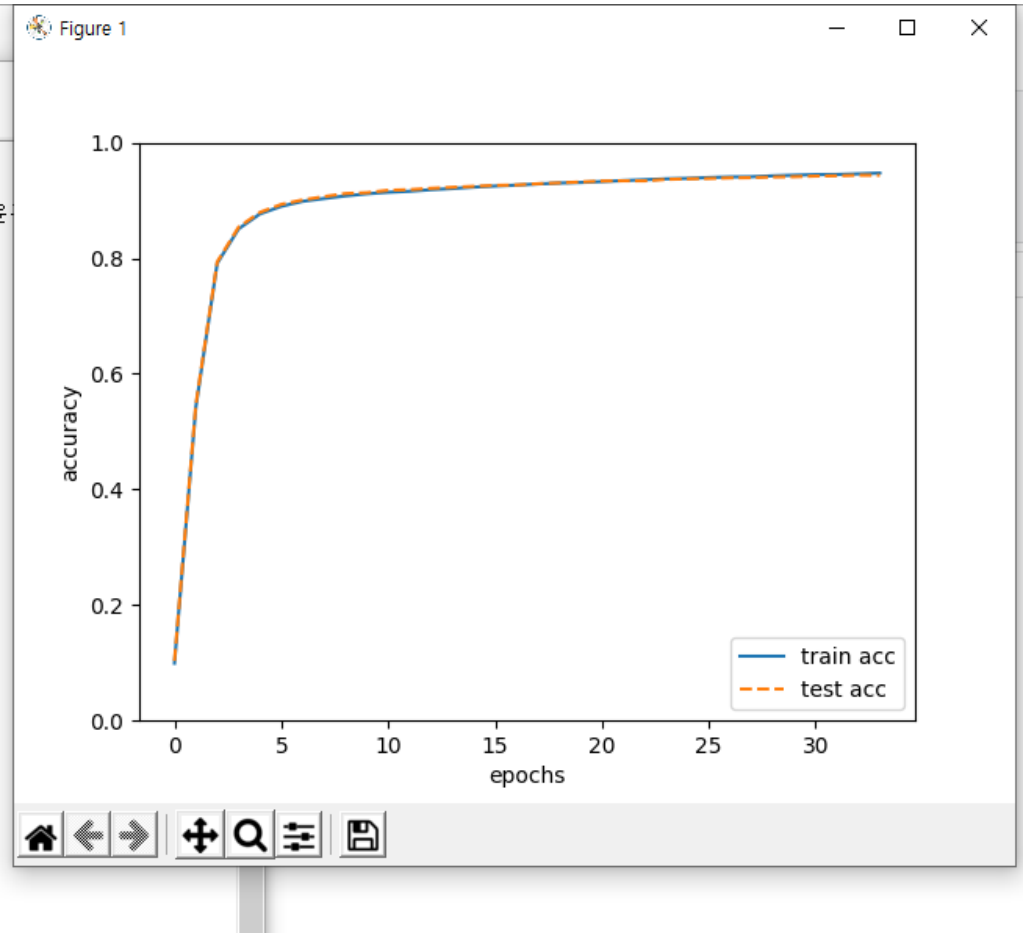
File Edit Format Run Options Window Help

```
# coding: utf-8
import sys, os
sys.path.append(os.pardir)  # 부모 디렉터리
import numpy as np
import matplotlib.pyplot as plt
from datasett.mnist import load_mnist
from two_layer_net import TwoLayerNet

# 데이터 읽기
(x_train, t_train), (x_test, t_test) = load_

network = TwoLayerNet(input_size=784, hidden

# 하이퍼파라미터
iters_num = 10000  # 반복 횟수를 적절히 설정
train_size = x_train.shape[0]
batch_size = 200   # 미니배치 크기
learning_rate = 0.1

train_loss_list = []
train_acc_list = []
test_acc_list = []

# 1에폭당 반복 수
iter_per_epoch = max(train_size / batch_size

for i in range(iters_num):
    # 미니배치 획득
    batch_mask = np.random.choice(train_size
    x_batch = x_train[batch_mask]
    t_batch = t_train[batch_mask]

    # 기울기 계산
    #grad = network.numerical_gradient(x_bat
    grad = network.gradient(x_batch, t_batch

    # 매개변수 갱신
    for key in ('W1', 'b1', 'W2', 'b2'):
        network.params[key] -= learning_rate
```

Python 3.8.5 Shell

File Edit Shell Debug Options Window Help

```
train acc, test acc | 0.9461, 0.9458
train acc, test acc | 0.94775, 0.9464

============ RESTART: D:₩CBNU₩딥러닝실제₩9주차 실습과제_강윤
y ============
train acc, test acc | 0.0993, 0.1032
train acc, test acc | 0.5424333333333333, 0.5463
train acc, test acc | 0.7910666666666667, 0.7937
train acc, test acc | 0.8507, 0.8537
train acc, test acc | 0.8765833333333334, 0.8793
train acc, test acc | 0.8896666666666667, 0.8931
train acc, test acc | 0.8983833333333333, 0.9008
train acc, test acc | 0.9031666666666667, 0.9065
train acc, test acc | 0.9076666666666666, 0.9118
train acc, test acc | 0.9113333333333333, 0.9133
train acc, test acc | 0.914, 0.9175
train acc, test acc | 0.9157333333333333, 0.9185
train acc, test acc | 0.9184333333333333, 0.9209
train acc, test acc | 0.9204166666666667, 0.9226
train acc, test acc | 0.9229, 0.9243
train acc, test acc | 0.92475, 0.9259
train acc, test acc | 0.92635, 0.9266
train acc, test acc | 0.9284166666666667, 0.9288
train acc, test acc | 0.92995, 0.9302
train acc, test acc | 0.9309333333333333, 0.9318
train acc, test acc | 0.9324833333333333, 0.9335
train acc, test acc | 0.9342666666666667, 0.9339
train acc, test acc | 0.9357166666666666, 0.9338
train acc, test acc | 0.93725, 0.9363
train acc, test acc | 0.93805, 0.9372
train acc, test acc | 0.9395166666666667, 0.9378
train acc, test acc | 0.9405666666666667, 0.9389
train acc, test acc | 0.9408666666666666, 0.9393
train acc, test acc | 0.9425, 0.9404
train acc, test acc | 0.9436, 0.9409
train acc, test acc | 0.9446, 0.942
train acc, test acc | 0.9448, 0.9425
train acc, test acc | 0.9459, 0.9435
train acc, test acc | 0.9472333333333334, 0.9433
```



batch size = 200
train acc = 0.947
test acc = 0.943

[1] Batch Size 갯수를 변화시키며, 훈련데이터와 테스트데이터의 정확도 추이 확인

train_neuralnet.py - D:₩CBNU₩딥러닝실제₩9주차 실습과제_강윤구₩train_n...

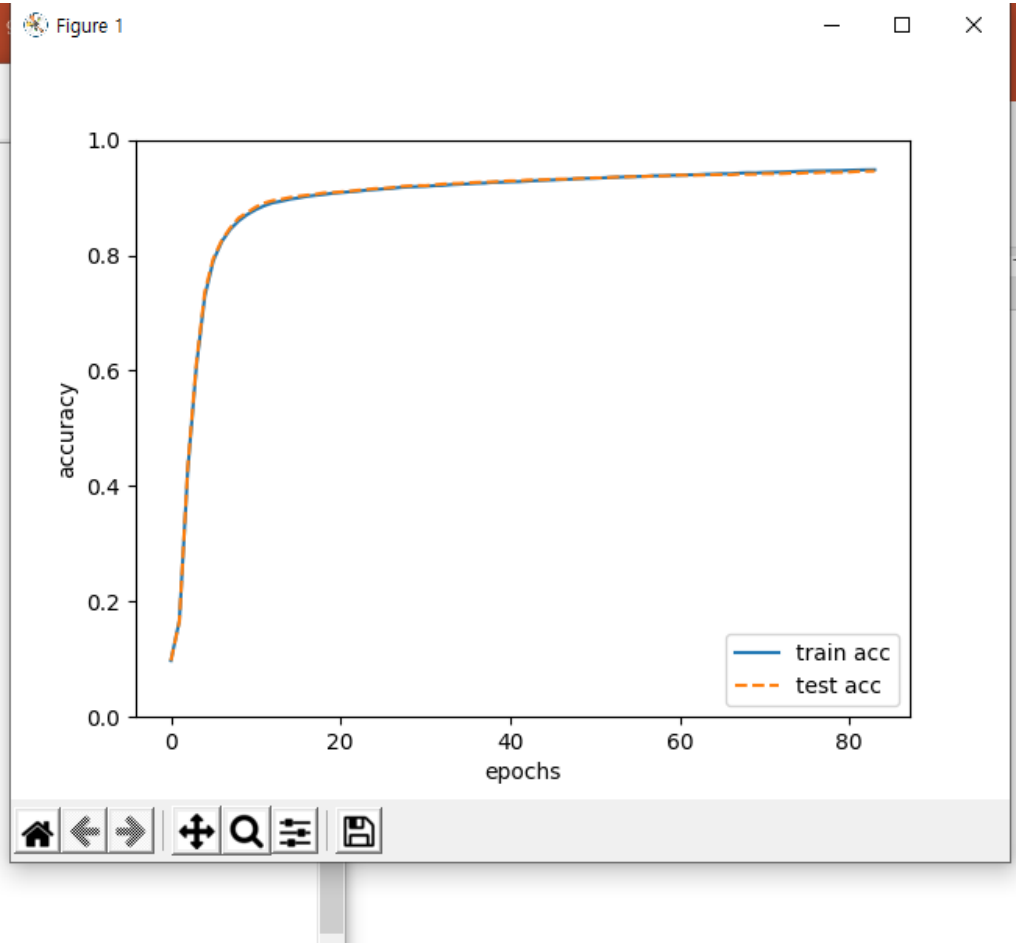
File Edit Format Run Options Window Help

```
# coding: utf-8
import sys, os
sys.path.append(os.pardir)  # 부모 디렉터리의 파일을 ㄱ
import numpy as np
import matplotlib.pyplot as plt
from datasett.mnist import load_mnist
from two_layer_net import TwoLayerNet

# 데이터 읽기
(x_train, t_train), (x_test, t_test) = load_mnist(norm

network = TwoLayerNet(input_size=784, hidden_size=50,

# 하이퍼파라미터
iters_num = 10000  # 반복 횟수를 적절히 설정한다.
train_size = x_train.shape[0]
batch_size = 500   # 미니배치 크기
learning_rate = 0.1

train_loss_list = []
train_acc_list = []
test_acc_list = []

# 1에폭당 반복 수
iter_per_epoch = max(train_size / batch_size, 1)

for i in range(iters_num):
    # 미니배치 획득
    batch_mask = np.random.choice(train_size, batch_si
    x_batch = x_train[batch_mask]
    t_batch = t_train[batch_mask]

    # 기울기 계산
    #grad = network.numerical_gradient(x_batch, t_batc
    grad = network.gradient(x_batch, t_batch)

    # 매개변수 갱신
    for key in ('W1', 'b1', 'W2', 'b2'):
        network.params[key] -= learning_rate * grad[ke
```

Python 3.8.5 Shell

File Edit Shell Debug Options Window Help

```
train acc, test acc | 0.9303, 0.9316
train acc, test acc | 0.931, 0.9319
train acc, test acc | 0.9316, 0.9318
train acc, test acc | 0.9324833333333333, 0.933
train acc, test acc | 0.9331333333333334, 0.9333
train acc, test acc | 0.93335, 0.9341
train acc, test acc | 0.93415, 0.934
train acc, test acc | 0.9347833333333333, 0.935
train acc, test acc | 0.9352833333333334, 0.9357
train acc, test acc | 0.9358666666666666, 0.9358
train acc, test acc | 0.9362166666666667, 0.9364
train acc, test acc | 0.9366666666666666, 0.9364
train acc, test acc | 0.9376333333333333, 0.9375
train acc, test acc | 0.9376833333333333, 0.9374
train acc, test acc | 0.9383166666666667, 0.938
train acc, test acc | 0.9385166666666667, 0.9389
train acc, test acc | 0.9392666666666667, 0.9383
train acc, test acc | 0.9394333333333333, 0.9387
train acc, test acc | 0.9402, 0.9389
train acc, test acc | 0.9404833333333333, 0.9396
train acc, test acc | 0.94115, 0.9397
train acc, test acc | 0.9410666666666667, 0.9394
train acc, test acc | 0.9422333333333334, 0.9406
train acc, test acc | 0.9428, 0.9402
train acc, test acc | 0.9426333333333333, 0.9404
train acc, test acc | 0.9433333333333334, 0.9413
train acc, test acc | 0.9437166666666666, 0.9412
train acc, test acc | 0.9441, 0.9413
train acc, test acc | 0.9446166666666667, 0.942
train acc, test acc | 0.94515, 0.9418
train acc, test acc | 0.9456166666666667, 0.9428
train acc, test acc | 0.946, 0.9428
train acc, test acc | 0.9461833333333334, 0.9438
train acc, test acc | 0.9464, 0.9432
train acc, test acc | 0.9465833333333333, 0.9437
train acc, test acc | 0.94715, 0.9441
train acc, test acc | 0.9476166666666667, 0.9452
train acc, test acc | 0.9479333333333333, 0.9455
train acc, test acc | 0.9480833333333333, 0.9457
```



batch size = 500
train acc = 0.947
test acc = 0.944

[1] Batch Size 갯수를 변화시키며, 훈련데이터와 테스트데이터의 정확도 추이 확인

train_neuralnet.py - D:₩CBNU₩딥러닝실제₩9주차 실습과제_강윤구₩train_n...

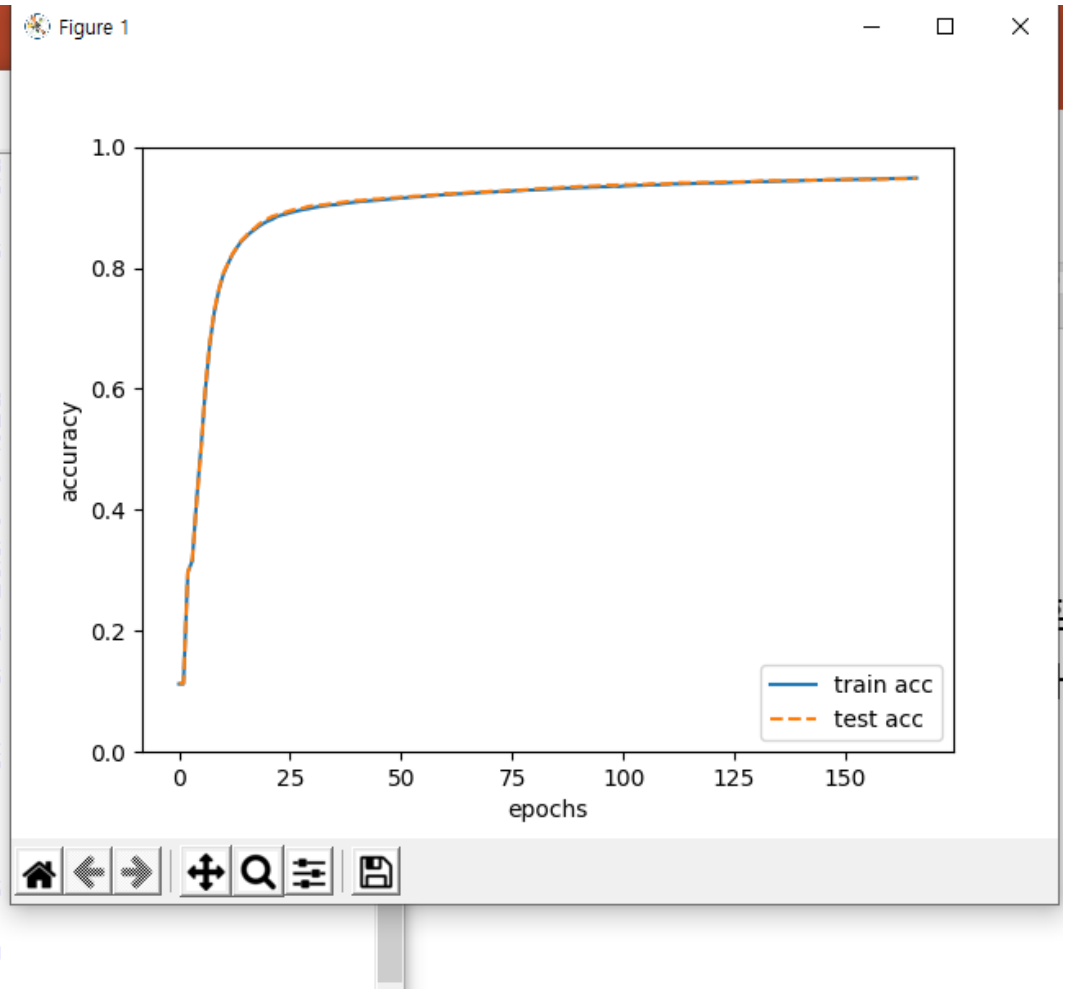
File Edit Format Run Options Window Help

```
# coding: utf-8
import sys, os
sys.path.append(os.pardir)  # 부모 디렉터리의 파일을 ?
import numpy as np
import matplotlib.pyplot as plt
from datasett.mnist import load_mnist
from two_layer_net import TwoLayerNet

# 데이터 읽기
(x_train, t_train), (x_test, t_test) = load_mnist(norm

network = TwoLayerNet(input_size=784, hidden_size=50,

# 하이퍼파라미터
iters_num = 10000  # 반복 횟수를 적절히 설정한다.
train_size = x_train.shape[0]
batch_size = 1000   # 미니배치 크기
learning_rate = 0.1

train_loss_list = []
train_acc_list = []
test_acc_list = []

# 1에폭당 반복 수
iter_per_epoch = max(train_size / batch_size, 1)

for i in range(iters_num):
    # 미니배치 획득
    batch_mask = np.random.choice(train_size, batch_si
    x_batch = x_train[batch_mask]
    t_batch = t_train[batch_mask]

    # 기울기 계산
    #grad = network.numerical_gradient(x_batch, t_batc
    grad = network.gradient(x_batch, t_batch)

    # 매개변수 갱신
    for key in ('W1', 'b1', 'W2', 'b2'):
        network.params[key] -= learning_rate * grad[ke
```

Python 3.8.5 Shell

File Edit Shell Debug Options Window Help

```
train acc, test acc | 0.9420333333333333, 0.9428
train acc, test acc | 0.9420333333333333, 0.9425
train acc, test acc | 0.94215, 0.943
train acc, test acc | 0.94265, 0.9436
train acc, test acc | 0.9424833333333333, 0.9436
train acc, test acc | 0.9428, 0.9437
train acc, test acc | 0.94295, 0.9441
train acc, test acc | 0.9432, 0.9435
train acc, test acc | 0.94335, 0.9447
train acc, test acc | 0.94365, 0.9442
train acc, test acc | 0.9436333333333333, 0.944
train acc, test acc | 0.9438333333333333, 0.9443
train acc, test acc | 0.9440833333333334, 0.9439
train acc, test acc | 0.9441833333333334, 0.9452
train acc, test acc | 0.9444, 0.9447
train acc, test acc | 0.9450166666666666, 0.9447
train acc, test acc | 0.94475, 0.9449
train acc, test acc | 0.9450333333333333, 0.9447
train acc, test acc | 0.9450666666666667, 0.9456
train acc, test acc | 0.9453166666666667, 0.9455
train acc, test acc | 0.9454166666666667, 0.9459
train acc, test acc | 0.9454, 0.9455
train acc, test acc | 0.9454833333333333, 0.9458
train acc, test acc | 0.9457166666666666, 0.9461
train acc, test acc | 0.9459666666666666, 0.9466
train acc, test acc | 0.9462, 0.9461
train acc, test acc | 0.94655, 0.9458
train acc, test acc | 0.9466666666666667, 0.9464
train acc, test acc | 0.9464333333333333, 0.9455
train acc, test acc | 0.94675, 0.9465
train acc, test acc | 0.94705, 0.9464
train acc, test acc | 0.9472, 0.9464
train acc, test acc | 0.9473333333333334, 0.947
train acc, test acc | 0.9474833333333333, 0.9471
train acc, test acc | 0.9478166666666666, 0.9475
train acc, test acc | 0.94785, 0.9475
train acc, test acc | 0.94795, 0.9473
train acc, test acc | 0.9481333333333334, 0.9479
train acc, test acc | 0.94835, 0.9479
```



batch size = 1000
train acc = 0.948
test acc = 0.947

[2] Hidden Size 갯수를 변화시키며, 훈련데이터와 테스트데이터의 정확도 추이 확인

train_neuralnet.py - D:₩CBNU₩딥러닝실제₩9주차 실습과제_강윤구₩train_n...

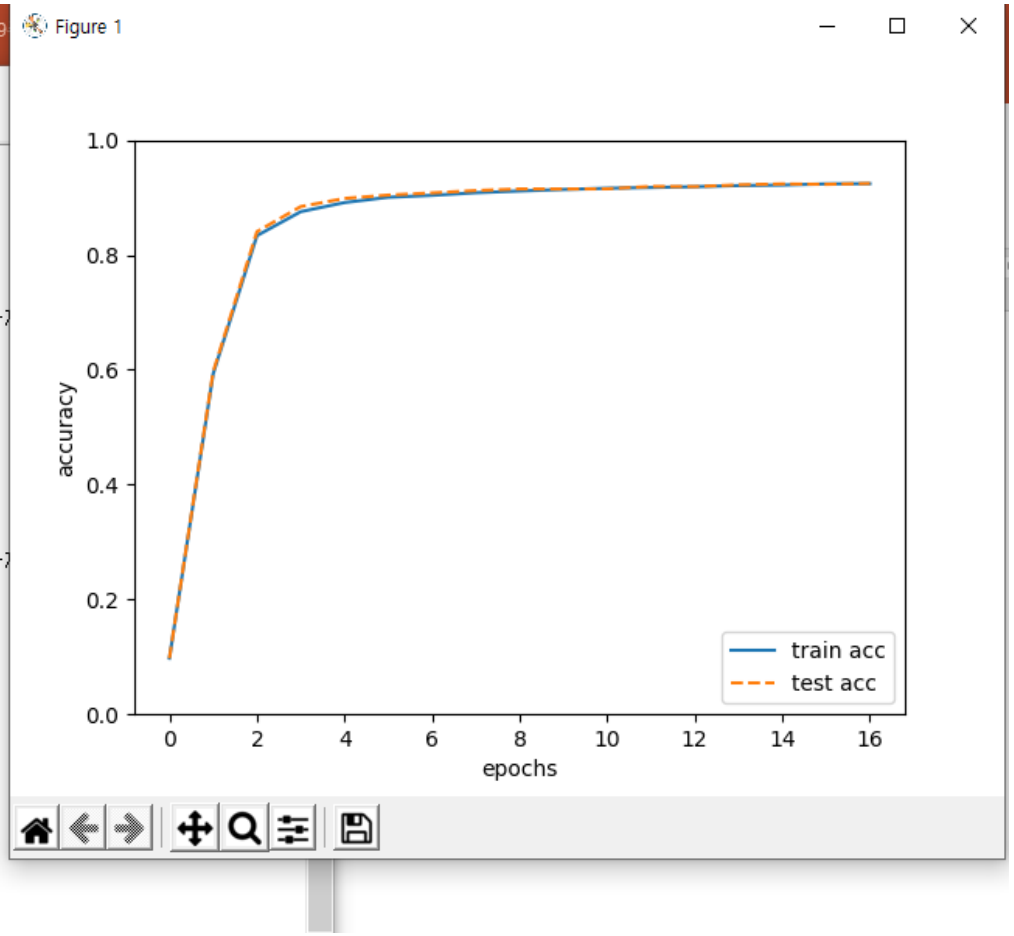
File Edit Format Run Options Window Help

```
# coding: utf-8
import sys, os
sys.path.append(os.pardir)  # 부모 디렉터리의 파일을 ?
import numpy as np
import matplotlib.pyplot as plt
from datasett.mnist import load_mnist
from two_layer_net import TwoLayerNet

# 데이터 읽기
(x_train, t_train), (x_test, t_test) = load_mnist(norm

network = TwoLayerNet(input_size=784, hidden_size=10,

# 하이퍼파라미터
iters_num = 10000  # 반복 횟수를 적절히 설정한다.
train_size = x_train.shape[0]
batch_size = 100   # 미니배치 크기
learning_rate = 0.1

train_loss_list = []
train_acc_list = []
test_acc_list = []

# 1에폭당 반복 수
iter_per_epoch = max(train_size / batch_size, 1)

for i in range(iters_num):
    # 미니배치 획득
    batch_mask = np.random.choice(train_size, batch_si
    x_batch = x_train[batch_mask]
    t_batch = t_train[batch_mask]

    # 기울기 계산
    #grad = network.numerical_gradient(x_batch, t_batc
    grad = network.gradient(x_batch, t_batch)

    # 매개변수 갱신
    for key in ('W1', 'b1', 'W2', 'b2'):
        network.params[key] -= learning_rate * grad[ke
```

Python 3.8.5 Shell

File Edit Shell Debug Options Window Help

```
train acc, test acc | 0.9258333333333333, 0.9261
train acc, test acc | 0.9298166666666666, 0.9306
train acc, test acc | 0.9318833333333333, 0.9321
train acc, test acc | 0.93295, 0.9339
train acc, test acc | 0.9370333333333334, 0.9365
train acc, test acc | 0.9395333333333333, 0.9367
train acc, test acc | 0.9417, 0.939

============ RESTART: D:₩CBNU₩딥러닝실제₩9주차 실습과?
y ============
train acc, test acc | 0.09035, 0.0892
train acc, test acc | 0.8725166666666667, 0.8746
train acc, test acc | 0.9066833333333333, 0.9084
train acc, test acc | 0.9178333333333333, 0.9185
train acc, test acc | 0.92575, 0.9259
train acc, test acc | 0.9338166666666666, 0.9357
train acc, test acc | 0.9373833333333333, 0.9383
train acc, test acc | 0.9427666666666666, 0.9425
train acc, test acc | 0.9458833333333333, 0.9454

============ RESTART: D:₩CBNU₩딥러닝실제₩9주차 실습과?
y ============
train acc, test acc | 0.09751666666666667, 0.0974
train acc, test acc | 0.5928833333333333, 0.5964
train acc, test acc | 0.8335166666666667, 0.8406
train acc, test acc | 0.8753, 0.8842
train acc, test acc | 0.8911833333333333, 0.8984
train acc, test acc | 0.9003833333333333, 0.9042
train acc, test acc | 0.9037333333333334, 0.908
train acc, test acc | 0.9083, 0.9121
train acc, test acc | 0.9112833333333333, 0.9149
train acc, test acc | 0.9139666666666667, 0.9148
train acc, test acc | 0.9160833333333334, 0.9151
train acc, test acc | 0.9173833333333333, 0.9193
train acc, test acc | 0.9187833333333333, 0.919
train acc, test acc | 0.9207833333333333, 0.922
train acc, test acc | 0.9216833333333333, 0.9237
train acc, test acc | 0.92375, 0.9229
train acc, test acc | 0.9241166666666667, 0.9245
```



hidden size = 10
train acc = 0.924
test acc = 0.924

hidden size = 10

[2] Hidden Size 갯수를 변화시키며, 훈련데이터와 테스트데이터의 정확도 추이 확인

train_neuralnet.py - D:₩CBNU₩딥러닝실제₩9주차 실습과제_강윤구₩train_n...

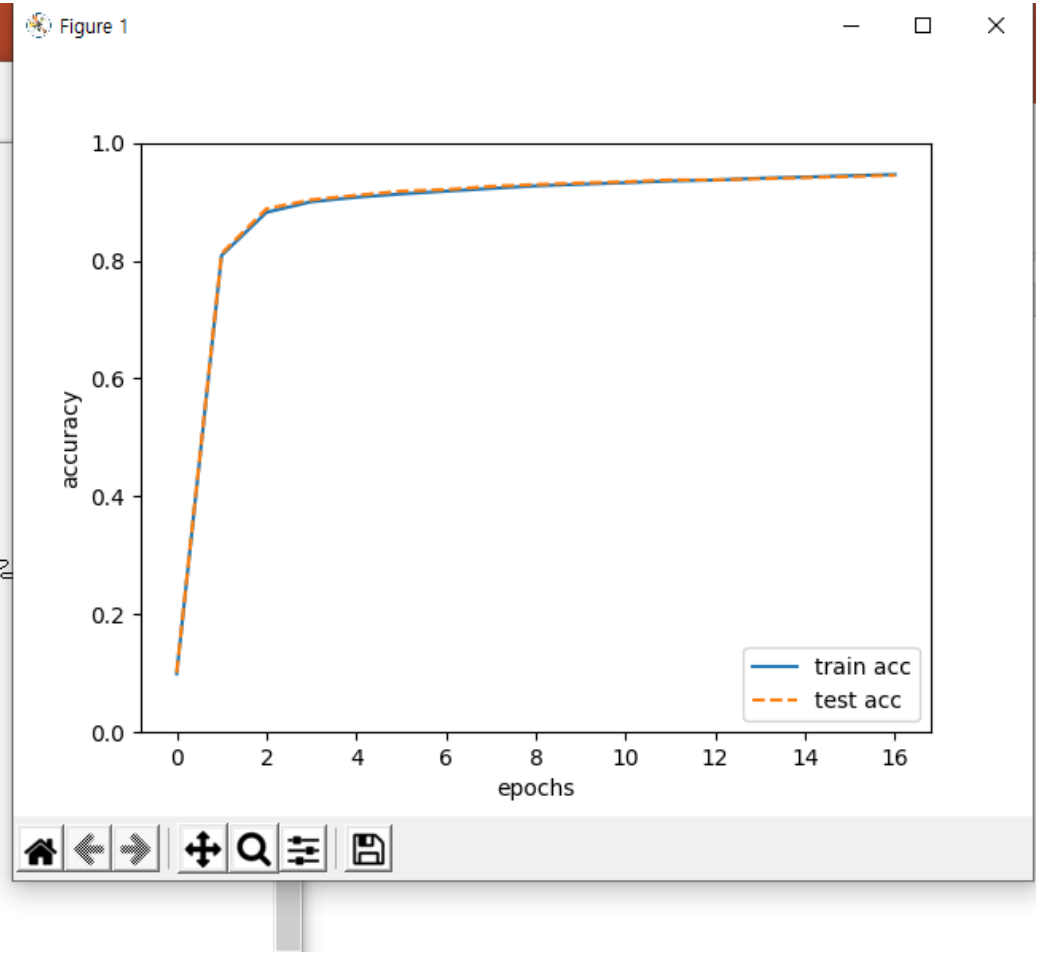
File Edit Format Run Options Window Help

```
# coding: utf-8
import sys, os
sys.path.append(os.pardir)  # 부모 디렉터리
import numpy as np
import matplotlib.pyplot as plt
from datasett.mnist import load_mnist
from two_layer_net import TwoLayerNet

# 데이터 읽기
(x_train, t_train), (x_test, t_test) = load_

network = TwoLayerNet(input_size=784, hidden

# 하이퍼파라미터
iters_num = 10000  # 반복 횟수를 적절히 설정
train_size = x_train.shape[0]
batch_size = 100   # 미니배치 크기
learning_rate = 0.1

train_loss_list = []
train_acc_list = []
test_acc_list = []

# 1에폭당 반복 수
iter_per_epoch = max(train_size / batch_size

for i in range(iters_num):
    # 미니배치 획득
    batch_mask = np.random.choice(train_size
    x_batch = x_train[batch_mask]
    t_batch = t_train[batch_mask]

    # 기울기 계산
    #grad = network.numerical_gradient(x_bat
    grad = network.gradient(x_batch, t_batch

    # 매개변수 갱신
    for key in ('W1', 'b1', 'W2', 'b2'):
        network.params[key] -= learning_rate
```

Python 3.8.5 Shell

File Edit Shell Debug Options Window Help

```
train acc, test acc | 0.9358833333333333, 0.9359
train acc, test acc | 0.93705, 0.9369
train acc, test acc | 0.9381666666666667, 0.9382
train acc, test acc | 0.9390666666666667, 0.9385
train acc, test acc | 0.9400833333333334, 0.9399
train acc, test acc | 0.94095, 0.9408
train acc, test acc | 0.9413666666666667, 0.9409
train acc, test acc | 0.9424833333333333, 0.9416
train acc, test acc | 0.9430333333333333, 0.9421
train acc, test acc | 0.9437, 0.9421
train acc, test acc | 0.9443166666666667, 0.9423
train acc, test acc | 0.9450666666666667, 0.9433
train acc, test acc | 0.9455833333333333, 0.9433
train acc, test acc | 0.9462666666666667, 0.9441
train acc, test acc | 0.9468833333333333, 0.9459
train acc, test acc | 0.9471333333333334, 0.9453
train acc, test acc | 0.9478333333333333, 0.9461
train acc, test acc | 0.9485, 0.9473
train acc, test acc | 0.9488666666666666, 0.9474

========== RESTART: D:₩CBNU₩딥러닝실제₩9주차 실습과제_강
y ==========
train acc, test acc | 0.09915, 0.1009
train acc, test acc | 0.8087166666666666, 0.812
train acc, test acc | 0.8819166666666667, 0.8879
train acc, test acc | 0.8998833333333334, 0.9031
train acc, test acc | 0.9075, 0.9108
train acc, test acc | 0.9131, 0.9178
train acc, test acc | 0.9177666666666666, 0.9205
train acc, test acc | 0.9223666666666667, 0.926
train acc, test acc | 0.9267666666666666, 0.9291
train acc, test acc | 0.9294833333333333, 0.9317
train acc, test acc | 0.9321833333333334, 0.9337
train acc, test acc | 0.93515, 0.937
train acc, test acc | 0.9366333333333333, 0.937
train acc, test acc | 0.93975, 0.9388
train acc, test acc | 0.9417666666666666, 0.9408
train acc, test acc | 0.9440333333333333, 0.9427
train acc, test acc | 0.9461333333333334, 0.9448
```



hidden size = 100
train acc = 0.946
test acc = 0.944

[2] Hidden Size 갯수를 변화시키며, 훈련데이터와 테스트데이터의 정확도 추이 확인

train_neuralnet.py - D:₩CBNU₩딥러닝실제₩9주차 실습과제_강윤구₩train_n...

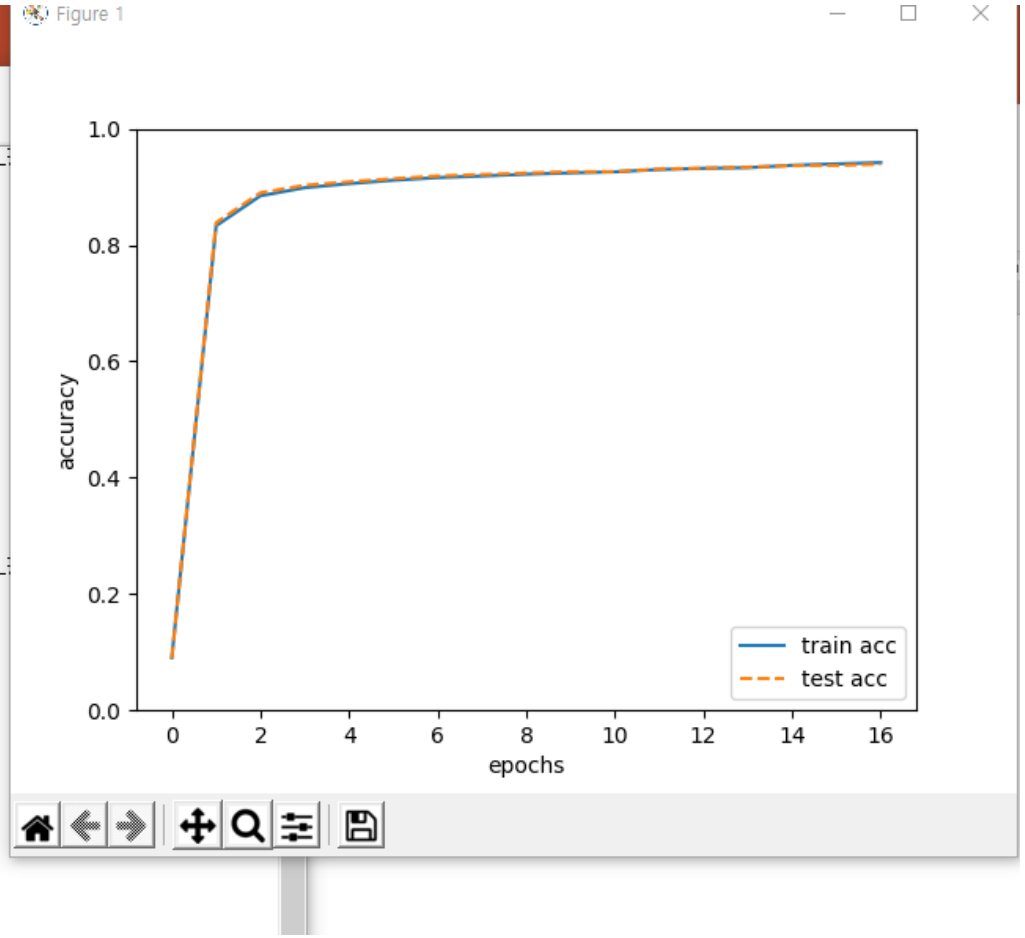
File Edit Format Run Options Window Help

```
# coding: utf-8
import sys, os
sys.path.append(os.pardir)  # 부모 디렉터리의 파일을 ㄱ
import numpy as np
import matplotlib.pyplot as plt
from datasett.mnist import load_mnist
from two_layer_net import TwoLayerNet

# 데이터 읽기
(x_train, t_train), (x_test, t_test) = load_mnist(norm

network = TwoLayerNet(input_size=784, hidden_size=200,

# 하이퍼파라미터
iters_num = 10000  # 반복 횟수를 적절히 설정한다.
train_size = x_train.shape[0]
batch_size = 100   # 미니배치 크기
learning_rate = 0.1

train_loss_list = []
train_acc_list = []
test_acc_list = []

# 1에폭당 반복 수
iter_per_epoch = max(train_size / batch_size, 1)

for i in range(iters_num):
    # 미니배치 획득
    batch_mask = np.random.choice(train_size, batch_si
    x_batch = x_train[batch_mask]
    t_batch = t_train[batch_mask]

    # 기울기 계산
    #grad = network.numerical_gradient(x_batch, t_batc
    grad = network.gradient(x_batch, t_batch)

    # 매개변수 갱신
    for key in ('W1', 'b1', 'W2', 'b2'):
        network.params[key] -= learning_rate * grad[ke
```

Python 3.8.5 Shell

File Edit Shell Debug Options Window Help

```
========== RESTART: D:₩CBNU₩딥러닝실제₩9주차 실습과제_ㅈ
y ============
train acc, test acc | 0.10551666666666666, 0.1039
train acc, test acc | 0.8222166666666667, 0.8261
train acc, test acc | 0.8862833333333333, 0.8906
train acc, test acc | 0.89975, 0.9034
train acc, test acc | 0.90725, 0.9099
train acc, test acc | 0.9125166666666666, 0.9153
train acc, test acc | 0.9169666666666667, 0.9203
train acc, test acc | 0.9204333333333333, 0.922
train acc, test acc | 0.92395, 0.927
train acc, test acc | 0.9264333333333333, 0.9288
train acc, test acc | 0.9305666666666667, 0.9322
train acc, test acc | 0.9331666666666667, 0.9333
train acc, test acc | 0.9367833333333333, 0.9374
train acc, test acc | 0.93775, 0.9379
train acc, test acc | 0.9400666666666667, 0.939
train acc, test acc | 0.94195, 0.9406
train acc, test acc | 0.9448, 0.9425

========== RESTART: D:₩CBNU₩딥러닝실제₩9주차 실습과제_ㅈ
y ============
train acc, test acc | 0.09035, 0.0892
train acc, test acc | 0.833, 0.8384
train acc, test acc | 0.8845833333333334, 0.8898
train acc, test acc | 0.89845, 0.9027
train acc, test acc | 0.90545, 0.9087
train acc, test acc | 0.9113, 0.914
train acc, test acc | 0.9158166666666666, 0.9187
train acc, test acc | 0.9185833333333333, 0.9212
train acc, test acc | 0.9217333333333333, 0.9238
train acc, test acc | 0.9239666666666667, 0.9261
train acc, test acc | 0.9258333333333333, 0.9261
train acc, test acc | 0.9298166666666666, 0.9306
train acc, test acc | 0.9318833333333333, 0.9321
train acc, test acc | 0.93295, 0.9339
train acc, test acc | 0.9370333333333334, 0.9365
train acc, test acc | 0.9395333333333333, 0.9367
```



hidden size = 200
train acc = 0.941
test acc = 0.939

[2] Hidden Size 갯수를 변화시키며, 훈련데이터와 테스트데이터의 정확도 추이 확인

train_neuralnet.py - D:₩CBNU₩딥러닝실제₩9주차 실습과제_강윤구₩train_n...

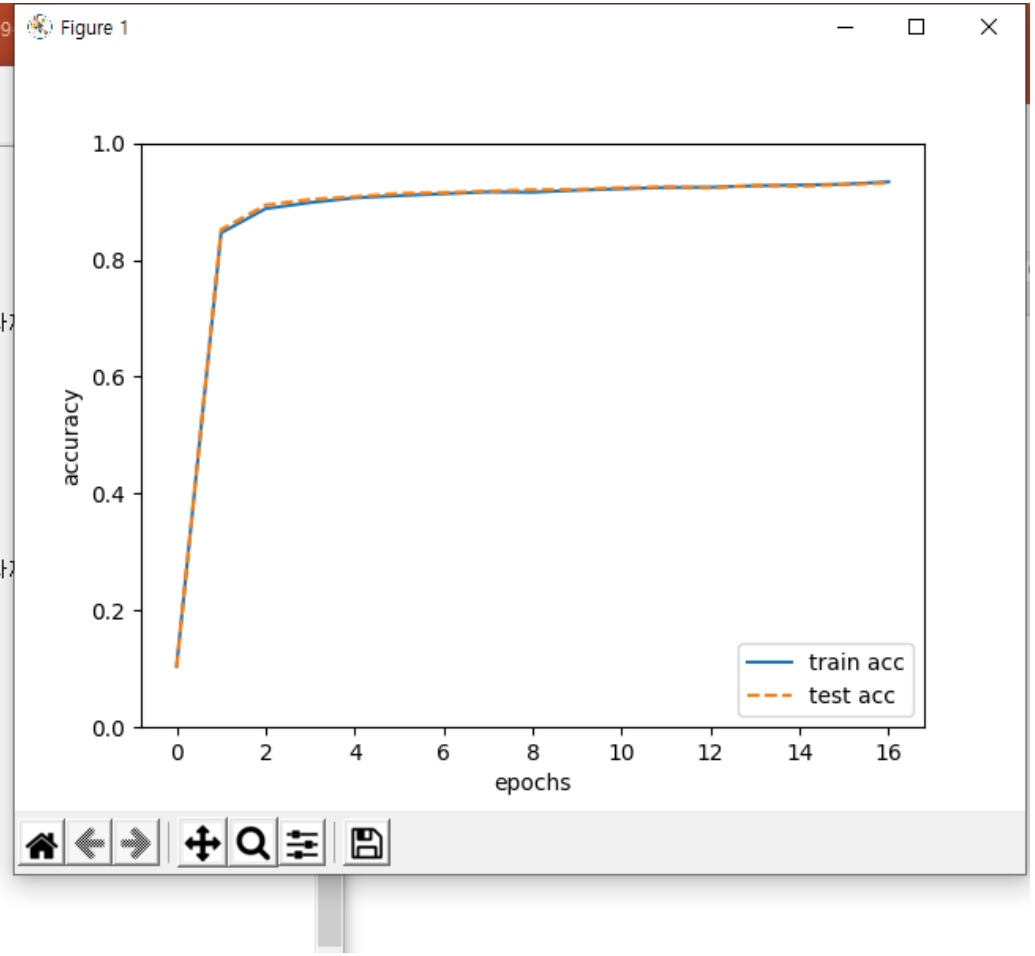
File Edit Format Run Options Window Help

```
# coding: utf-8
import sys, os
sys.path.append(os.pardir)  # 부모 디렉터리의 파일을 ?
import numpy as np
import matplotlib.pyplot as plt
from datasett.mnist import load_mnist
from two_layer_net import TwoLayerNet

# 데이터 읽기
(x_train, t_train), (x_test, t_test) = load_mnist(norm

network = TwoLayerNet(input_size=784, hidden_size=500,

# 하이퍼파라미터
iters_num = 10000  # 반복 횟수를 적절히 설정한다.
train_size = x_train.shape[0]
batch_size = 100   # 미니배치 크기
learning_rate = 0.1

train_loss_list = []
train_acc_list = []
test_acc_list = []

# 1에폭당 반복 수
iter_per_epoch = max(train_size / batch_size, 1)

for i in range(iters_num):
    # 미니배치 획득
    batch_mask = np.random.choice(train_size, batch_si
    x_batch = x_train[batch_mask]
    t_batch = t_train[batch_mask]

    # 기울기 계산
    #grad = network.numerical_gradient(x_batch, t_batc
    grad = network.gradient(x_batch, t_batch)

    # 매개변수 갱신
    for key in ('W1', 'b1', 'W2', 'b2'):
        network.params[key] -= learning_rate * grad[ke
```

Python 3.8.5 Shell

File Edit Shell Debug Options Window Help

```
train acc, test acc | 0.9160833333333334, 0.9151
train acc, test acc | 0.9173833333333333, 0.9193
train acc, test acc | 0.9187833333333333, 0.919
train acc, test acc | 0.9207833333333333, 0.922
train acc, test acc | 0.9216833333333333, 0.9237
train acc, test acc | 0.92375, 0.9229
train acc, test acc | 0.9241166666666667, 0.9245

============ RESTART: D:₩CBNU₩딥러닝실제₩9주차 실습과?
y ============
train acc, test acc | 0.09751666666666667, 0.0974
train acc, test acc | 0.8744166666666666, 0.8797
train acc, test acc | 0.9055666666666666, 0.9075
train acc, test acc | 0.9205, 0.9239
train acc, test acc | 0.9285833333333333, 0.9301
train acc, test acc | 0.9353666666666667, 0.936
train acc, test acc | 0.93885, 0.9388
train acc, test acc | 0.9430333333333333, 0.9422
train acc, test acc | 0.9471333333333334, 0.9461

============ RESTART: D:₩CBNU₩딥러닝실제₩9주차 실습과?
y ============
train acc, test acc | 0.10441666666666667, 0.1028
train acc, test acc | 0.8462166666666666, 0.852
train acc, test acc | 0.88765, 0.8939
train acc, test acc | 0.8981, 0.9035
train acc, test acc | 0.9062, 0.9081
train acc, test acc | 0.9099666666666667, 0.9138
train acc, test acc | 0.9131666666666667, 0.9152
train acc, test acc | 0.9164166666666667, 0.9177
train acc, test acc | 0.9158666666666667, 0.92
train acc, test acc | 0.9193166666666667, 0.9207
train acc, test acc | 0.9217333333333333, 0.9238
train acc, test acc | 0.9240333333333334, 0.9249
train acc, test acc | 0.9243166666666667, 0.9234
train acc, test acc | 0.9268833333333333, 0.9276
train acc, test acc | 0.92805, 0.9258
train acc, test acc | 0.92955, 0.9289
train acc, test acc | 0.9335, 0.9314
```



hidden size = 500
train acc = 0.933
test acc = 0.931

[3] 위의 [1][2]에서 오버피팅 문제

위의 [1]에서 보다시피 batch size를 변화시켜가며 학습이 진행되어도
훈련데이터와 시험 데이터를 평가한 정확도에는 차이가 없었으나
위의 [2]에서 보다시피 hidden size를 변화시켜가며 학습이 진행 되었을때
개수가 어느정도 수준을 넘어서면 오히려 정확도가 떨어지는 모습을 보입니다.
따라서, 오버피팅을 줄이기 위해서는 데이터의 양은 늘리고, 복잡도는 줄여야 한다고 판단됩니다.