

# 【 HW1 】 Multilayer Perceptrons

108321033 吳明騰

108321037 林峻葆

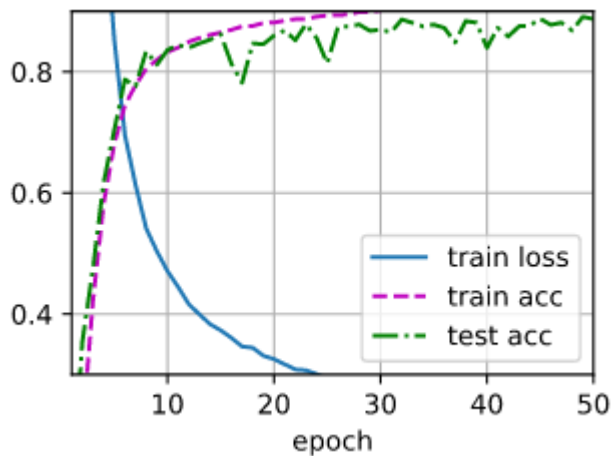
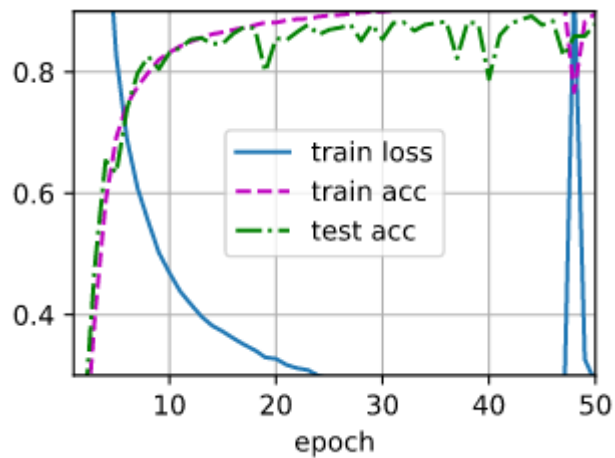
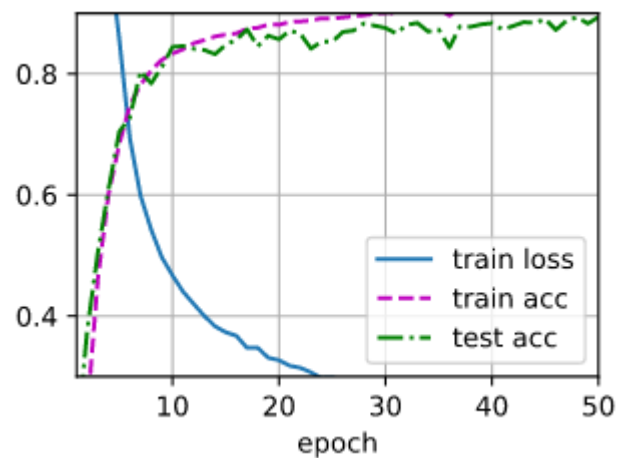
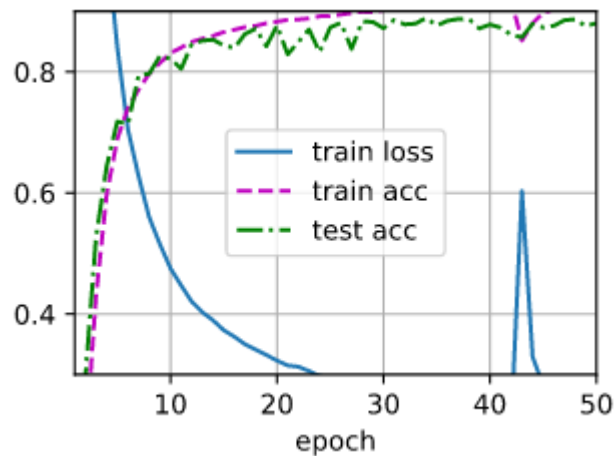
# No dropout

- 3 hidden layers
-

# Code

```
• from mxnet import gluon, init, npx
• from mxnet.gluon import nn
• from d2l import mxnet as d2l
•
• npx.set_np()
•
• dropout = 0.5
• batch_size = 256
• train_iter, test_iter = d2l.load_data_fashion_mnist(batch_size)
•
• net = nn.Sequential()
• for add_layer in range(3):
•     net.add(nn.Dense(256, activation='relu'))
• net.add(nn.Dense(10))
• net.initialize(init.Normal(sigma=0.01))
•
• loss = gluon.loss.SoftmaxCrossEntropyLoss()
•
• trainer = gluon.Trainer(net.collect_params(), 'sgd', {'learning_rate': 0.1})
•
•
• num_epochs = 50
• d2l.train_ch3(net, train_iter, test_iter, loss, num_epochs, trainer)
```

# Result



# Has dropout

- 3 hidden layers
-

# Code

```
• from mxnet import gluon, init, npx
• from mxnet.gluon import nn
• from d2l import mxnet as d2l
•
• npx.set_np()
•
• dropout = 0.5
• batch_size = 256
• train_iter, test_iter = d2l.load_data_fashion_mnist(batch_size)
•
• net = nn.Sequential()
• for add_layer in range(3):
•     net.add(nn.Dense(256, activation='relu'), nn.Dropout(dropout))
• net.add(nn.Dense(10))
• net.initialize(init.Normal(sigma=0.01))
•
• loss = gluon.loss.SoftmaxCrossEntropyLoss()
•
• trainer = gluon.Trainer(net.collect_params(), 'sgd', {'learning_rate': 0.1})
•
•
• num_epochs = 50
• d2l.train_ch3(net, train_iter, test_iter, loss, num_epochs, trainer)
```

# Result

