

Multilayer Perceptron in Gluon

```
In [1]: import d2l
        from mxnet import gluon, init
        from mxnet.gluon import loss as gloss, nn
```

The Model

```
In [2]: net = nn.Sequential()  
net.add(nn.Dense(256, activation='relu'))  
net.add(nn.Dense(10))  
net.initialize(init.Normal(sigma=0.01))
```

Training

```
In [3]: batch_size = 256
train_iter, test_iter = d2l.load_data_fashion_mnist(batch_size)

loss = gloss.SoftmaxCrossEntropyLoss()
trainer = gluon.Trainer(net.collect_params(), 'sgd', {'learning_rate': 0.5})
num_epochs = 10
d2l.train_ch3(net, train_iter, test_iter, loss, num_epochs, batch_size,
              None, None, trainer)
```

```
epoch 1, loss 0.8333, train acc 0.688, test acc 0.817
epoch 2, loss 0.5031, train acc 0.815, test acc 0.829
epoch 3, loss 0.4303, train acc 0.842, test acc 0.860
epoch 4, loss 0.3942, train acc 0.855, test acc 0.857
epoch 5, loss 0.3694, train acc 0.864, test acc 0.873
epoch 6, loss 0.3534, train acc 0.869, test acc 0.864
epoch 7, loss 0.3410, train acc 0.873, test acc 0.875
epoch 8, loss 0.3221, train acc 0.880, test acc 0.883
epoch 9, loss 0.3158, train acc 0.884, test acc 0.882
epoch 10, loss 0.3083, train acc 0.885, test acc 0.885
```