



# 多层感知机的从零开始实现

Eng Teong Cheah

# 目录


1. 多层感知机的从零开始实现
2. 定义模型参数
3. 定义激活函数
4. 定义模型
5. 定义损失函数
6. 训练模型

# 多层感知机的从零开始实现



```
import d2l  
from mxnet import nd, gluon
```

## 定义模型参数




```
num_inputs, num_outputs, num_hiddens = 784, 10, 256

W1 = nd.random.normal(scale=0.01, shape=(num_inputs, num_hiddens))
b1 = nd.zeros(num_hiddens)
W2 = nd.random.normal(scale=0.01, shape=(num_hiddens,
num_outputs))
b2 = nd.zeros(num_outputs)
params = [W1, b1, W2, b2]

for param in params:
    param.attach_grad()
```

# 定义激活函数



```
def relu(X):  
    return nd.maximum(X, 0)
```

# 定义模型



```
def net(X):  
    X = X.reshape((-1, num_inputs))  
    H = relu(nd.dot(X, W1) + b1)  
    return nd.dot(H, W2) + b2
```

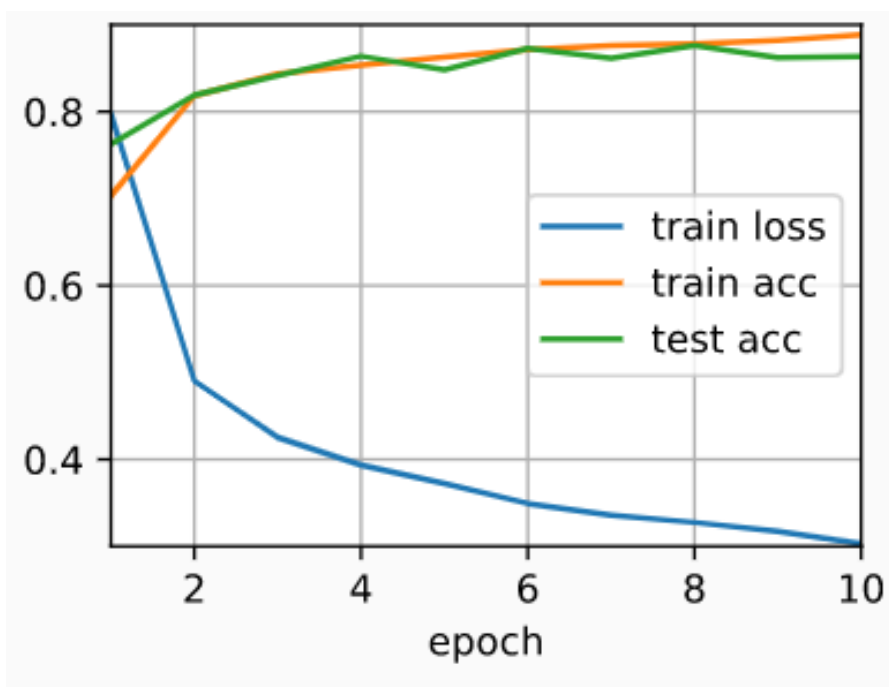
# 定义损失函数



```
loss = gluon.loss.SoftmaxCrossEntropyLoss()
```

# 训练模型

```
num_epochs, lr = 10, 0.5
d2l.train_ch3(net, train_iter, test_iter, loss, num_epochs,
              lambda batch_size: d2l.sgd(params, lr, batch_size))
```





---

# 谢谢！

**Does anyone have any questions?**

**Twitter:** @walkercet

**Blog:** <https://ceteongvanness.wordpress.com>

# 资源

Dive into Deep Learning