In [2]:

```
import numpy as np
```

In [3]:

```
data = np.random.normal(0, 0.01, (40000, 3))
```

In [4]:

```
np.random.shuffle(data)
```

In [9]:

```
train_data = data[:30000]
test_data = data[30000:]
```

k折交叉

In [10]:

```
k = 4
num_val_samples = len(train_data) // k
num_epochs = 20
all_val_scores = []
```

In [ ]:

```
for i in range(k):
    print('processing fold #', i)
    print('range', i * num_val_samples, (i+1) * num_val_samples)
    # 第k个分区的验证集
    x_val = train_data[i * num_val_samples :  (i+1) * num_val_samples ]
    y_val = train_data[i * num_val_samples :  (i+1) * num_val_samples ]

    # 剩下的作为训练集，
    x_train =  train_data[ : i * num_val_samples] + train_data[(i+1) * num_val_samples : ]
    y_train = np.concatenate([ train_data[ : i * num_val_samples], train_data[(i+1) * num_v

    model = build_model()
    model.fit(x_train, y_train, epochs=num_epochs, batch_size=1)

    val_score = model.evaluate(x_val, y_val, verbose=0)

    all_val_scores.append(val_score)
```

In [ ]:

```
validation_score = np.average(all_val_scores)
```

最终模型的训练

In [ ]:

```
model = build_model()
model.fit(train_data)
model.evaluate(test_data)
```