Drop function

在层与层之间加入噪音，丢弃一些神经元，防止过拟合。

深度网络的泛化性质令人费解，而这种泛化性质的数学基础仍然是悬而未决的研究问题。

所以本章不做理论解释，只讲解代码！

Pytorch Drop Code

```
import torch
import torch.nn as nn
from d2l import torch as d2l
# Too easy! ！！！！！！！！！！
num_epochs, lr, batch_size = 10, 0.5, 256
dropout1, dropout2 = 0.2, 0.5
loss = nn.CrossEntropyLoss()
train_iter, test_iter = d2l.load_data_fashion_mnist(batch_size)

net = nn.Sequential(nn.Flatten(),
                    nn.Linear(784,256),
                    nn.ReLU(),
                    nn.Dropout(dropout1),
                    nn.Linear(256,256),
                    nn.ReLU(),
                    nn.Dropout(dropout2),
                    nn.Linear(256,10)
                   )

def init_weights(m):
    if type(m) == nn.Linear:
        nn.init.normal_(m.weight,std=0.01)

net.apply(init_weights)

trainer = torch.optim.SGD(net.parameters(),lr=lr)
d2l.train_ch3(net,train_iter, test_iter, loss, num_epochs,trainer)
d2l.plt.show()
```