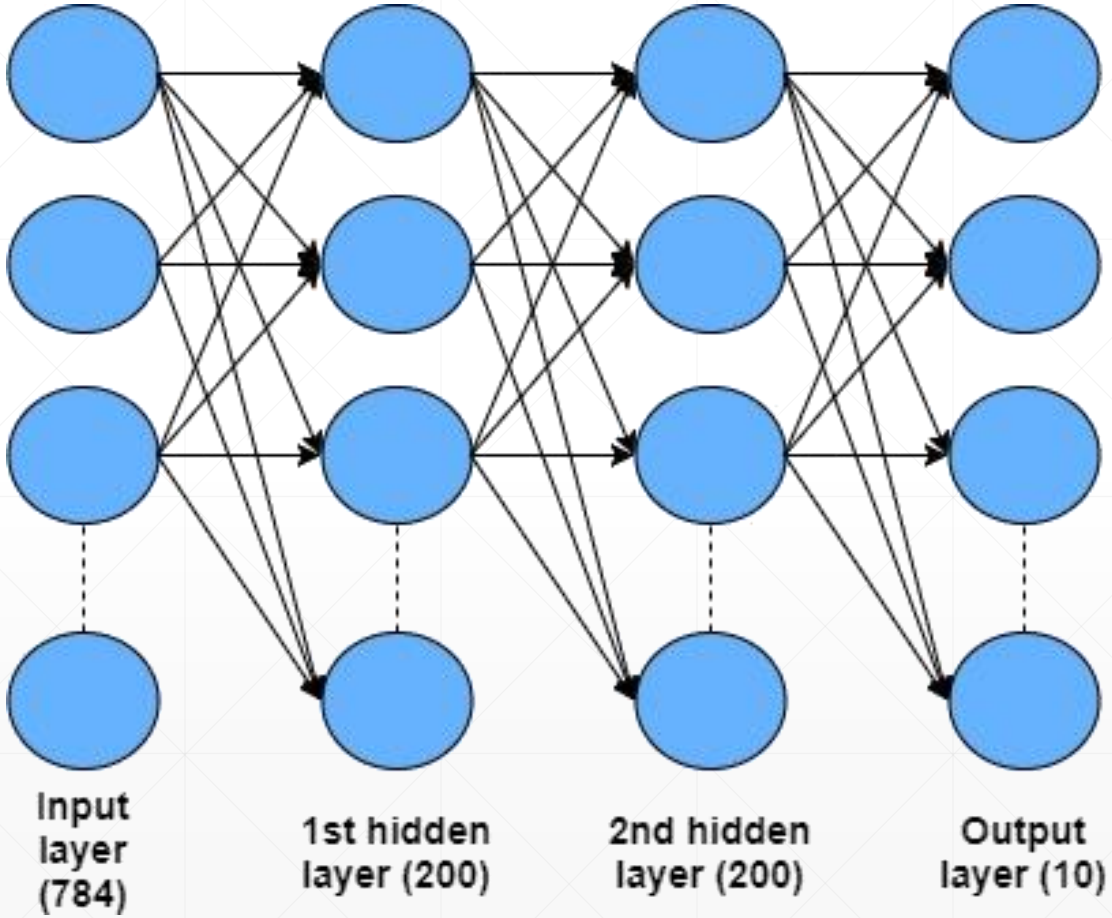
PyTorch

多分类问题

主讲人：龙良曲


Input layer (784)
1st hidden layer (200)
2nd hidden layer (200)
Output layer (10)

Network Architecture

参数初始化

神经元个数

```
w1, b1 = torch.randn(200, 784, requires_grad=True),\
         torch.zeros(200, requires_grad=True)
w2, b2 = torch.randn(200, 200, requires_grad=True),\
         torch.zeros(200, requires_grad=True)
w3, b3 = torch.randn(10, 200, requires_grad=True),\
         torch.zeros(10, requires_grad=True)
```

设计模型

前向传播函数

```
def forward(x):
    x = x@w1.t() + b1
    x = F.relu(x)
    x = x@w2.t() + b2
    x = F.relu(x)
    x = x@w3.t() + b3
    x = F.relu(x)
    return x
```

Train

优化

```
optimizer = optim.SGD([w1, b1, w2, b2, w3, b3], lr=learning_rate)
criteon = nn.CrossEntropyLoss()

for epoch in range(epochs):

    for batch_idx, (data, target) in enumerate(train_loader):
        data = data.view(-1, 28*28)

        logits = forward(data)
        loss = criteon(logits, target)

        optimizer.zero_grad()
        loss.backward()
        # print(w1.grad.norm(), w2.grad.norm())
        optimizer.step()
```

数据打平

损失

三步技术的

有时调用print大法

```
C:\ProgramData\conda\python.exe F:/PytorchTutorial/lesson26-LR多分类实战/main.py
Downloading http://yann.lecun.com/exdb/mnist/train-images-idx3-ubyte.gz
Downloading http://yann.lecun.com/exdb/mnist/train-labels-idx1-ubyte.gz
Downloading http://yann.lecun.com/exdb/mnist/t10k-images-idx3-ubyte.gz
Downloading http://yann.lecun.com/exdb/mnist/t10k-labels-idx1-ubyte.gz
Processing...
Done!
```

em....

```
w1, b1 = torch.randn(200, 784, requires_grad=True),\
         torch.zeros(200, requires_grad=True)
w2, b2 = torch.randn(200, 200, requires_grad=True),\
         torch.zeros(200, requires_grad=True)
w3, b3 = torch.randn(10, 200, requires_grad=True),\
         torch.zeros(10, requires_grad=True)

torch.nn.init.kaiming_normal_(w1)
torch.nn.init.kaiming_normal_(w2)
torch.nn.init.kaiming_normal_(w3)
```

下一课时

PyTorch全连接层

Thank You.