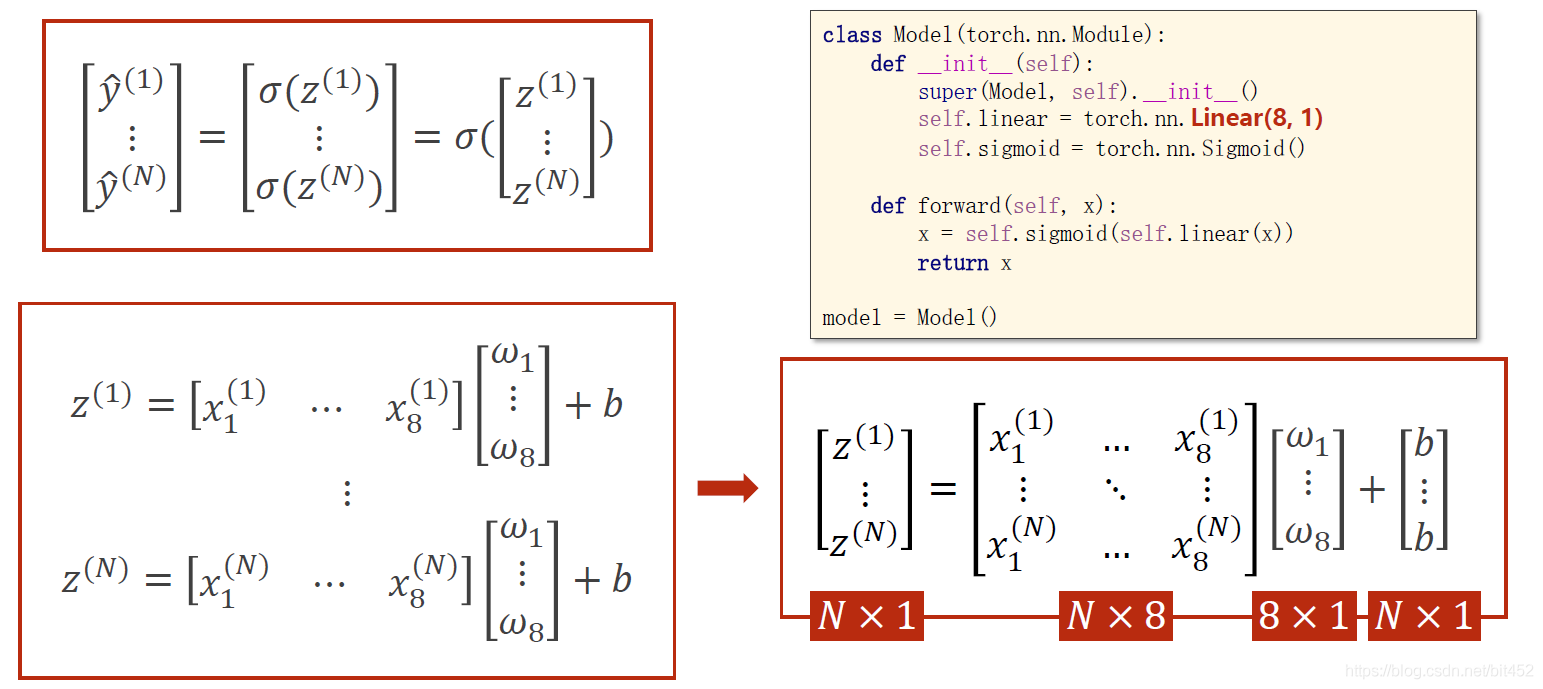
PyTorch 深度学习实践 第7讲

第7讲 处理多维特征的输入 源代码

B站 刘二大人 ，传送门PyTorch深度学习实践——处理多维特征的输入

视频中截图



说明：1、乘的权重(w)都一样，加的偏置(b)也一样。b变成矩阵时使用广播机制。神经网络的参数w和b是网络需要学习的，其他是已知的。

2、学习能力越强，有可能会把输入样本中噪声的规律也学到。我们要学习数据本身真实数据的规律，学习能力要有泛化能力。

3、该神经网络共3层；第一层是8维到6维的非线性空间变换，第二层是6维到4维的非线性空间变换，第三层是4维到1维的非线性空间变换。

4、本算法中torch.nn.Sigmoid() # 将其看作是网络的一层，而不是简单的函数使用

5、torch.sigmoid、torch.nn.Sigmoid和torch.nn.functional.sigmoid的区别

|  |
| --- |
| import numpy as np  import torch  import matplotlib.pyplot as plt    # prepare dataset  xy = np.loadtxt('diabetes.csv', delimiter=',', dtype=np.float32)  x\_data = torch.from\_numpy(xy[:, :-1]) # 第一个‘：’是指读取所有行，第二个‘：’是指从第一列开始，最后一列不要  y\_data = torch.from\_numpy(xy[:, [-1]]) # [-1] 最后得到的是个矩阵    # design model using class      class Model(torch.nn.Module):  def \_\_init\_\_(self):  super(Model, self).\_\_init\_\_()  self.linear1 = torch.nn.Linear(8, 6) # 输入数据x的特征是8维，x有8个特征  self.linear2 = torch.nn.Linear(6, 4)  self.linear3 = torch.nn.Linear(4, 1)  self.sigmoid = torch.nn.Sigmoid() # 将其看作是网络的一层，而不是简单的函数使用    def forward(self, x):  x = self.sigmoid(self.linear1(x))  x = self.sigmoid(self.linear2(x))  x = self.sigmoid(self.linear3(x)) # y hat  return x      model = Model()    # construct loss and optimizer  # criterion = torch.nn.BCELoss(size\_average = True)  criterion = torch.nn.BCELoss(reduction='mean')  optimizer = torch.optim.SGD(model.parameters(), lr=0.1)    epoch\_list = []  loss\_list = []  # training cycle forward, backward, update  for epoch in range(100):  y\_pred = model(x\_data)  loss = criterion(y\_pred, y\_data)  print(epoch, loss.item())  epoch\_list.append(epoch)  loss\_list.append(loss.item())    optimizer.zero\_grad()  loss.backward()    optimizer.step()    plt.plot(epoch\_list, loss\_list)  plt.ylabel('loss')  plt.xlabel('epoch')  plt.show() |

代码说明 ：1、diabetes.csv数据集老师给了下载地址，该数据集需和源代码放在同一个文件夹内。

2、如果想查看某些层的参数，以神经网络的第一层参数为例，可按照以下方法进行。

————————————————

版权声明：本文为CSDN博主「错错莫」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/bit452/article/details/109682078