A7：特征学习

任务1. PCA 或 kernel PCA (10 points)

使用PCA或kernel PCA对手写数字数据集MINST进行降维。观察前两个特征向量所对应的图像，即将数据嵌入到R2空间。绘制降维后的数据，并分析二维特征是否能够足以完成对输入的分类，对结果进行分析和评价。

任务2. Autoencoder (10 points)

使用自动编码器学习输入的特征表示。尝试设计一个全链接前馈神经网络或卷积神经网络。尝试使用不同的损失函数和正则化方法。

附加题 (BONUS: 10 points)

模型训练中，你可以尝试任何可以提升模型性能的合理的方法。例如其它的网络结构、设计多个隐藏层、引入降噪自动编码器等任何你能想到的方法。计算模型在训练集和测试集上的损失，并对结果进行讨论。

**目录**

[1、 PCA 2](#_Toc3645)

[（1）原理 2](#_Toc28037)

[（2）设计 2](#_Toc21398)

[（3）训练 2](#_Toc10394)

[（4）结果 3](#_Toc6935)

[2、kernel PCA 5](#_Toc6520)

[（1） 原理 5](#_Toc23930)

[（2） 设计 5](#_Toc20977)

[（3） 训练 5](#_Toc31048)

[（4） 结果 5](#_Toc25215)

[3、Autoencoder 6](#_Toc219)

[（1） 原理 6](#_Toc28851)

[（2） 设计 6](#_Toc30104)

[①Linear 6](#_Toc2577)

[②Conv 6](#_Toc16862)

[③降噪自编码输出作为特征表示自编码器输入 7](#_Toc24254)

[（3） 训练 7](#_Toc3959)

[①Linear 7](#_Toc10926)

[②Conv 13](#_Toc18067)

[③ 14](#_Toc2357)

[（4） 结果 15](#_Toc8593)

[4、总结与反思 15](#_Toc3619)

1. PCA

（1）原理

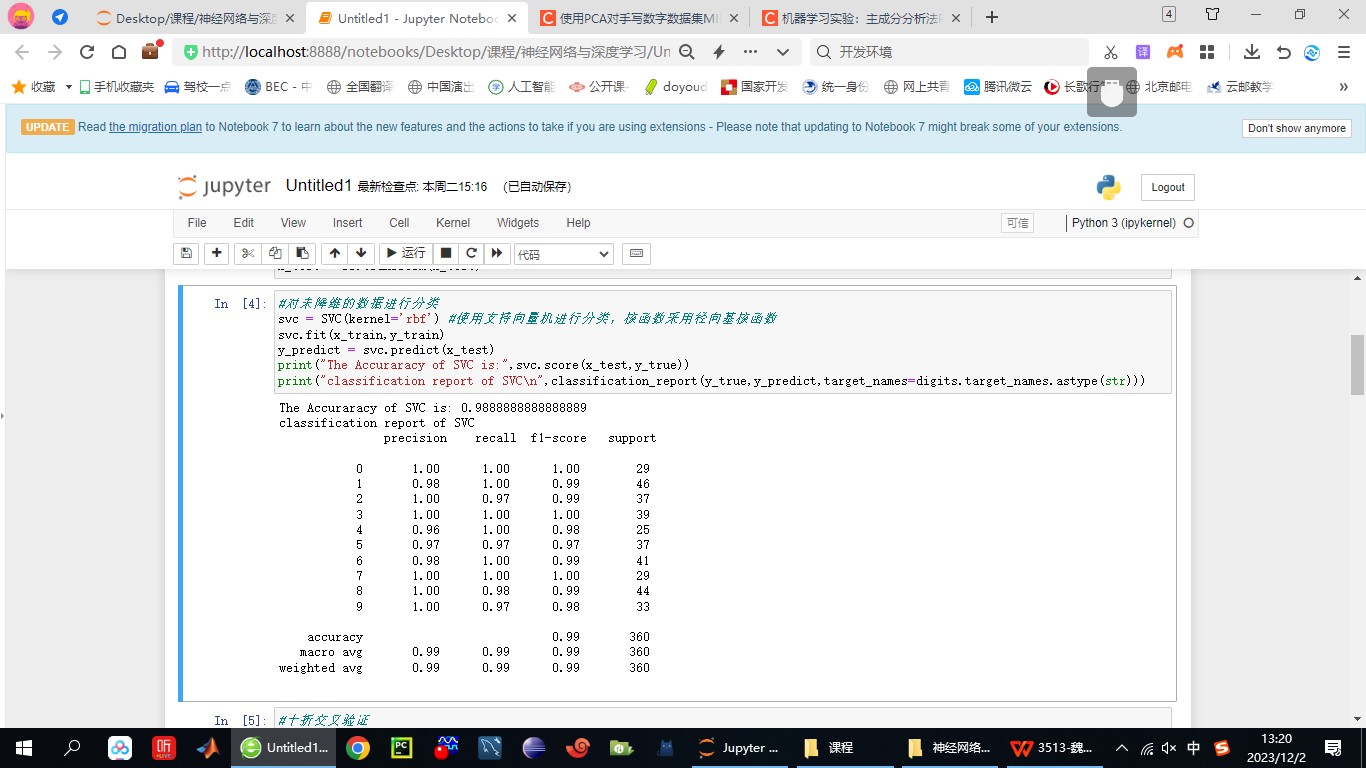
PCA是一种线性降维方法，通过某个投影矩阵将高维空间中的原始样本点线性投影到多个线性无关的低维空间，达到降维的目的。

（2）设计

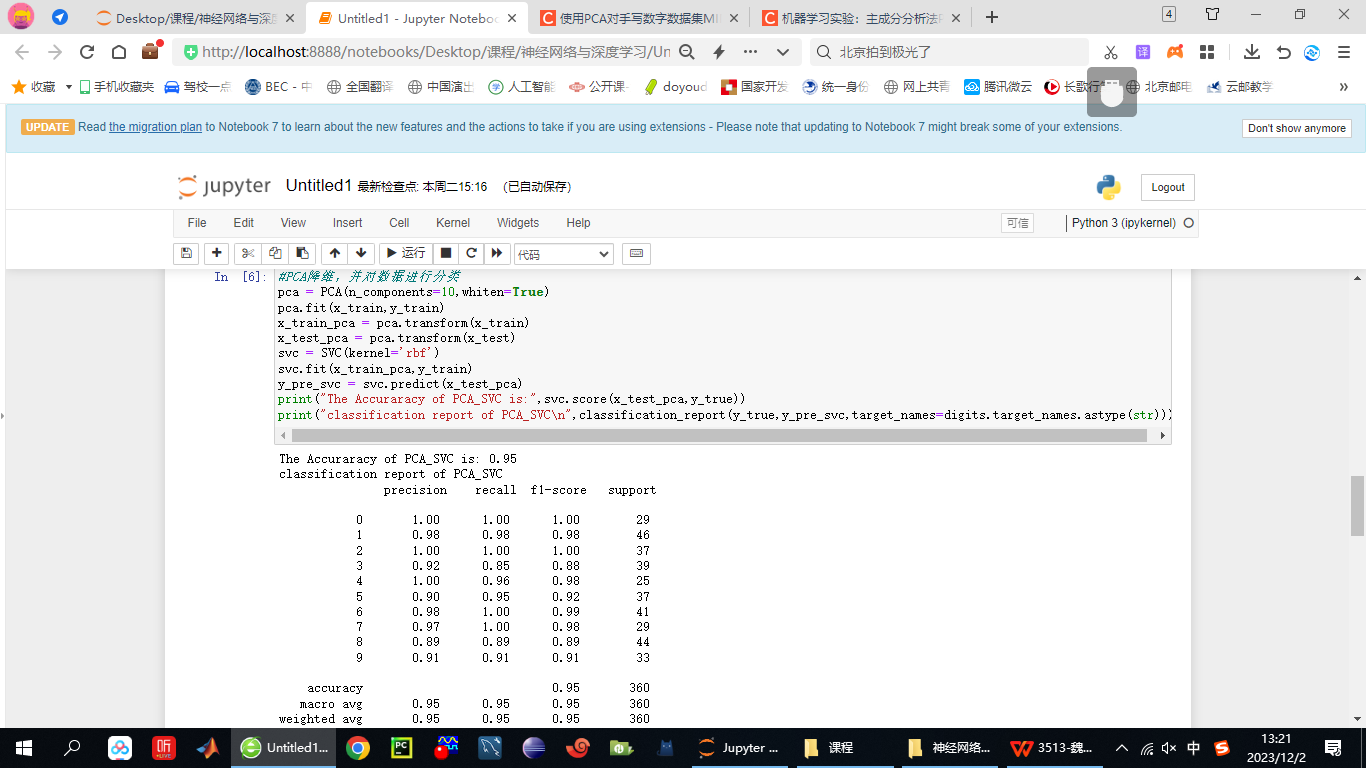
从sklearn.decomposition导入PCA进行实验，分别输出pca处理前分类和处理后分类的相关指标。

（3）训练

处理前：



降至十维：



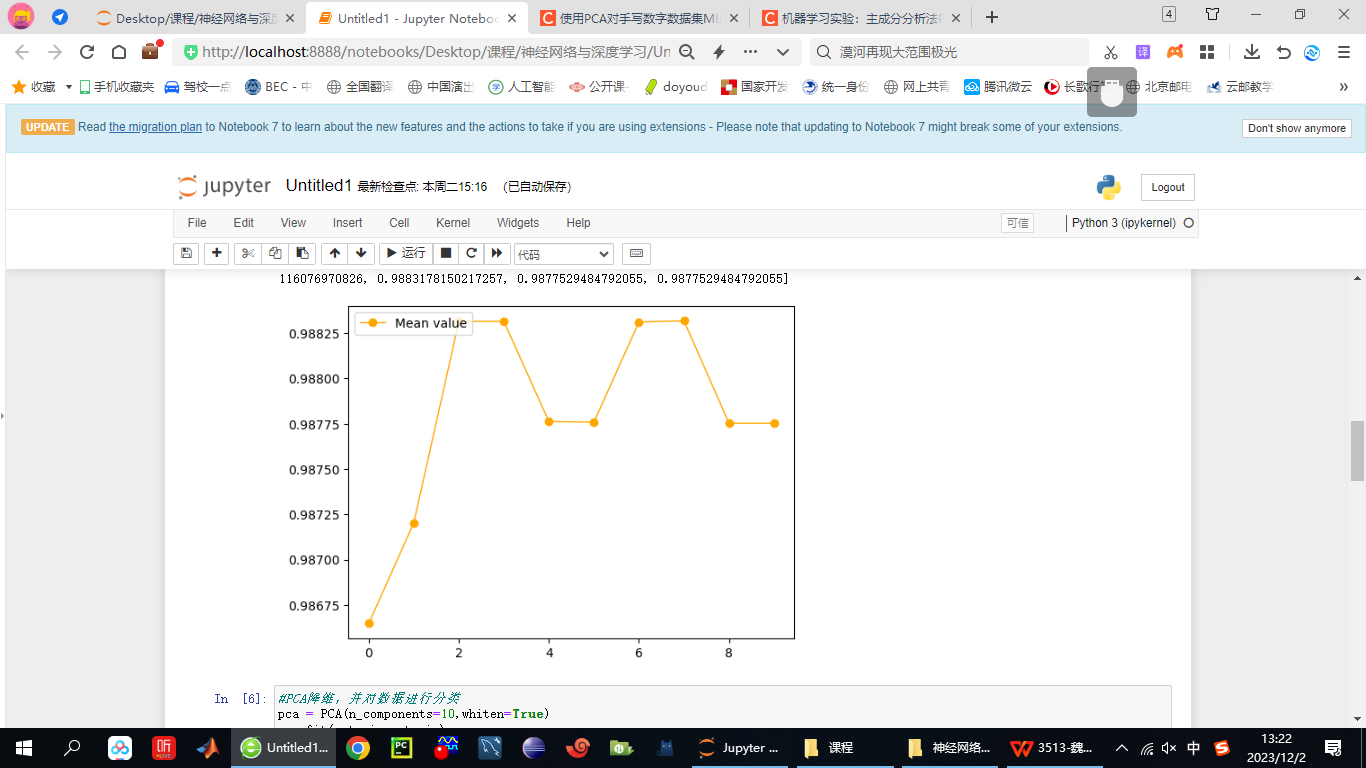
降至二维：



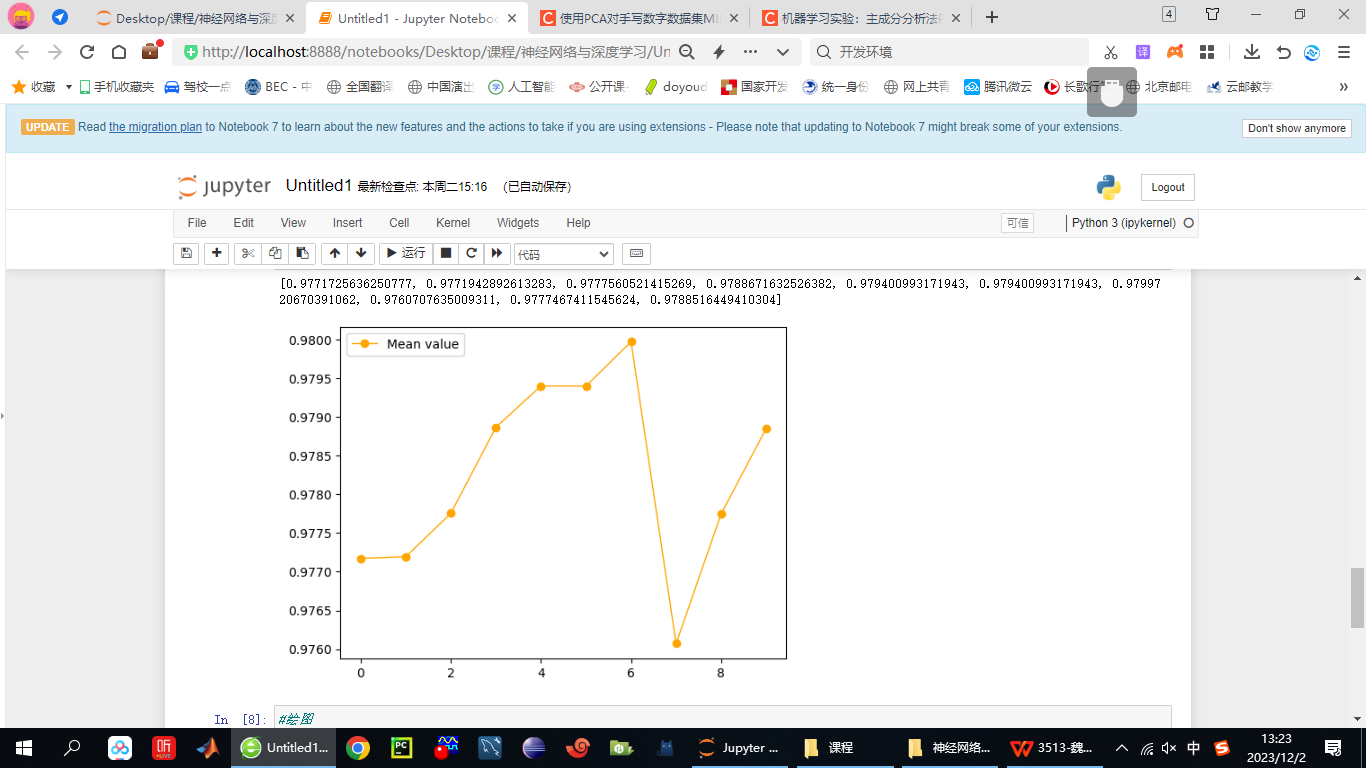
（4）结果

10折交叉验证：

未处理：



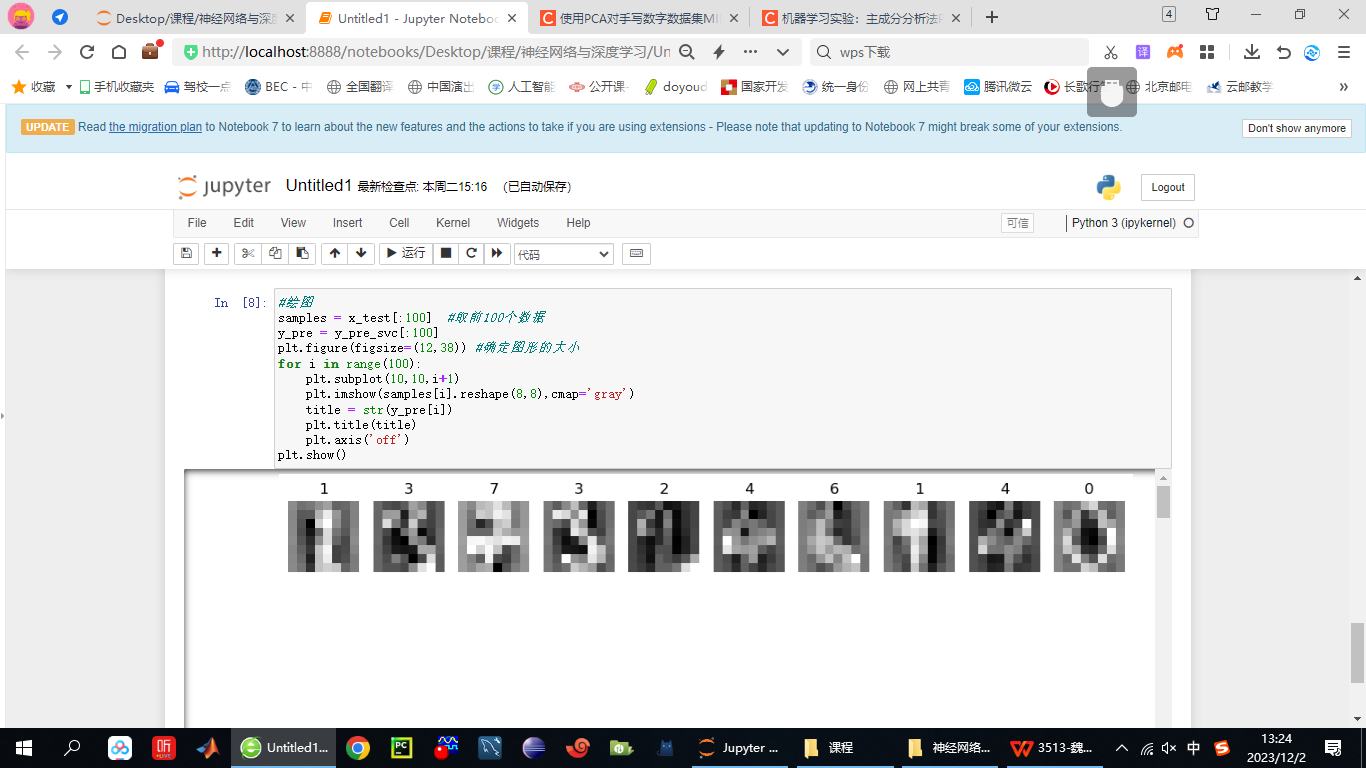
十维：



二维：



绘制降维后的数据（test集前十个）：



结论：二维特征不足以完成分类。

2、kernel PCA

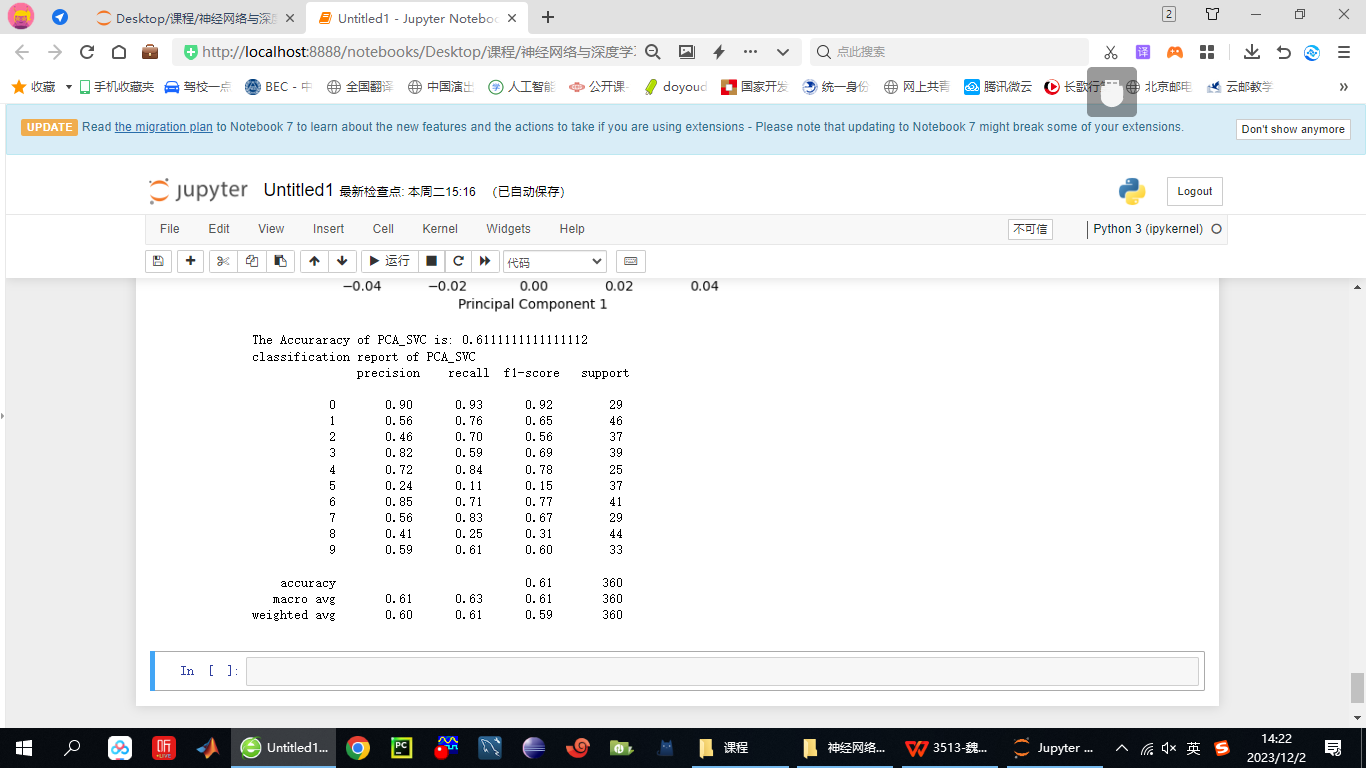
1. 原理

相较PCA，kernel PCA是一个先升维后降维的过程。利用核函数将线性不可分的输入空间映射到线性可分的高维特征空间中，再利用PCA降到想要的维度。

1. 设计

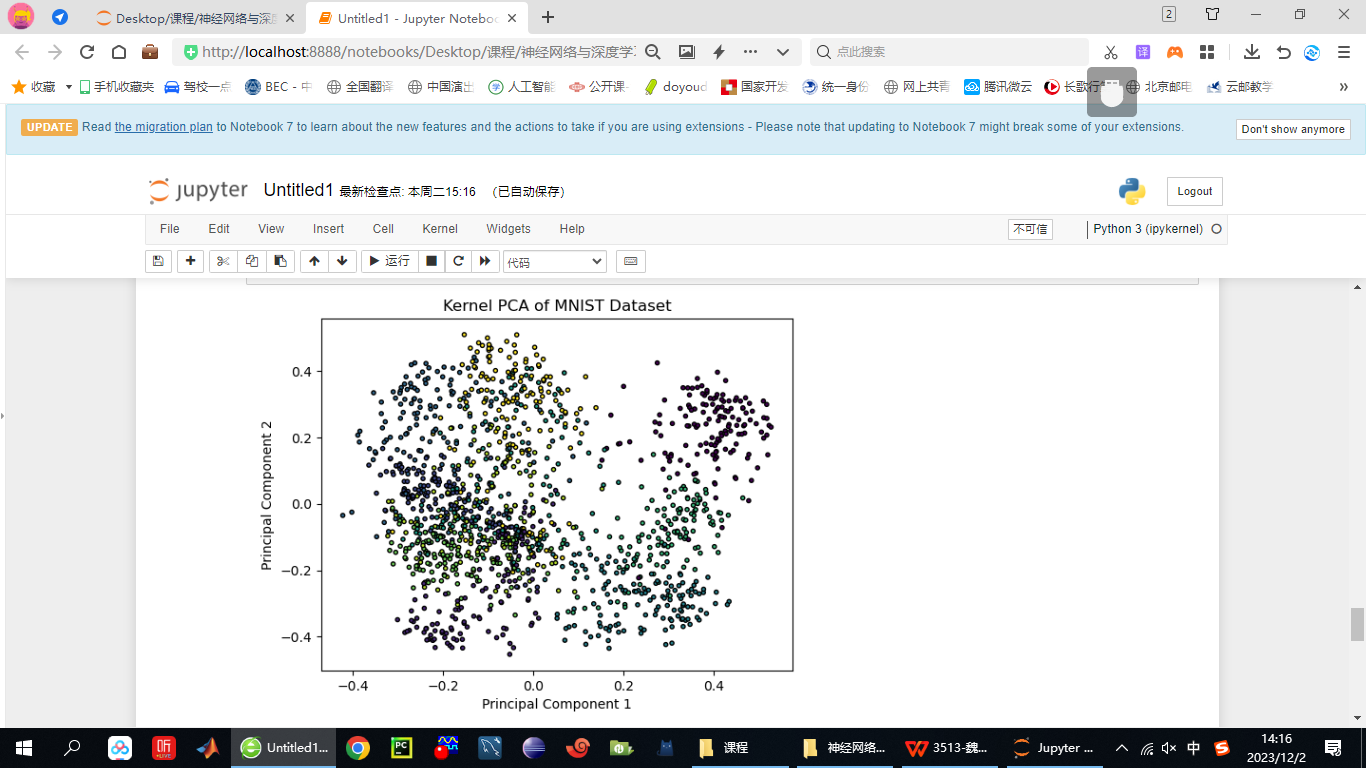
from sklearn.decomposition import KernelPCA，利用高斯核升维，再降至二维。

1. 训练



同样，通过使用降维后的数据进行分类，在二维空间中难以实现良好的分类边界，SVC分类准确率只有60%左右。

1. 结果



数据点在二维空间中呈现出一定的分离趋势，但并没有清晰的类别划分。这也表明二维特征可能不能完全足以完成对输入的分类任务。

结论：同上。

3、Autoencoder

1. 原理

第一个网络是一个编码器，负责接收输入x，并将输入通过函数h变换为信号y；第二个网络将编码的信号y作为其输入，通过函数f得到重构的信号r；定义误差e为原始输入x与重构信号r的差，e=x–r，网络训练的目标是减少均方误差，误差被反向传播回隐藏层。

1. 设计

①Linear

分别使用三层全连接实现编码器解码器，relu作为激活函数，sigmoid做解码器最后输出。

②Conv

分别使用三层卷积实现编码器，三层转置卷积作为解码器，relu作为激活函数，sigmoid做解码器最后输出。

为防止通道数不匹配，我们在Dataloader中设置drop\_last=True，丢弃最后一块不满足batchsize的数据。

定义基于卷积神经网络的自编码器如下：

class ConvAutoencoder(nn.Module):

def \_\_init\_\_(self):

super(ConvAutoencoder, self).\_\_init\_\_()

self.encoder = nn.Sequential(

nn.Conv2d(128, 16, kernel\_size=3, stride=1, padding=1),

nn.ReLU(),

nn.Conv2d(16, 32, kernel\_size=3, stride=1, padding=1),

nn.ReLU(),

nn.Conv2d(32, 64, kernel\_size=3, stride=1, padding=1)

)

self.decoder = nn.Sequential(

nn.ConvTranspose2d(64, 32, kernel\_size=3, stride=1, padding=1),

nn.ReLU(),

nn.ConvTranspose2d(32, 16, kernel\_size=3, stride=1, padding=1),

nn.ReLU(),

nn.ConvTranspose2d(16, 128, kernel\_size=3, stride=1, padding=1),

nn.Sigmoid()

)

def forward(self, x):

x = x.unsqueeze(1)

x = self.encoder(x)

x = self.decoder(x)

x = x.squeeze(1)

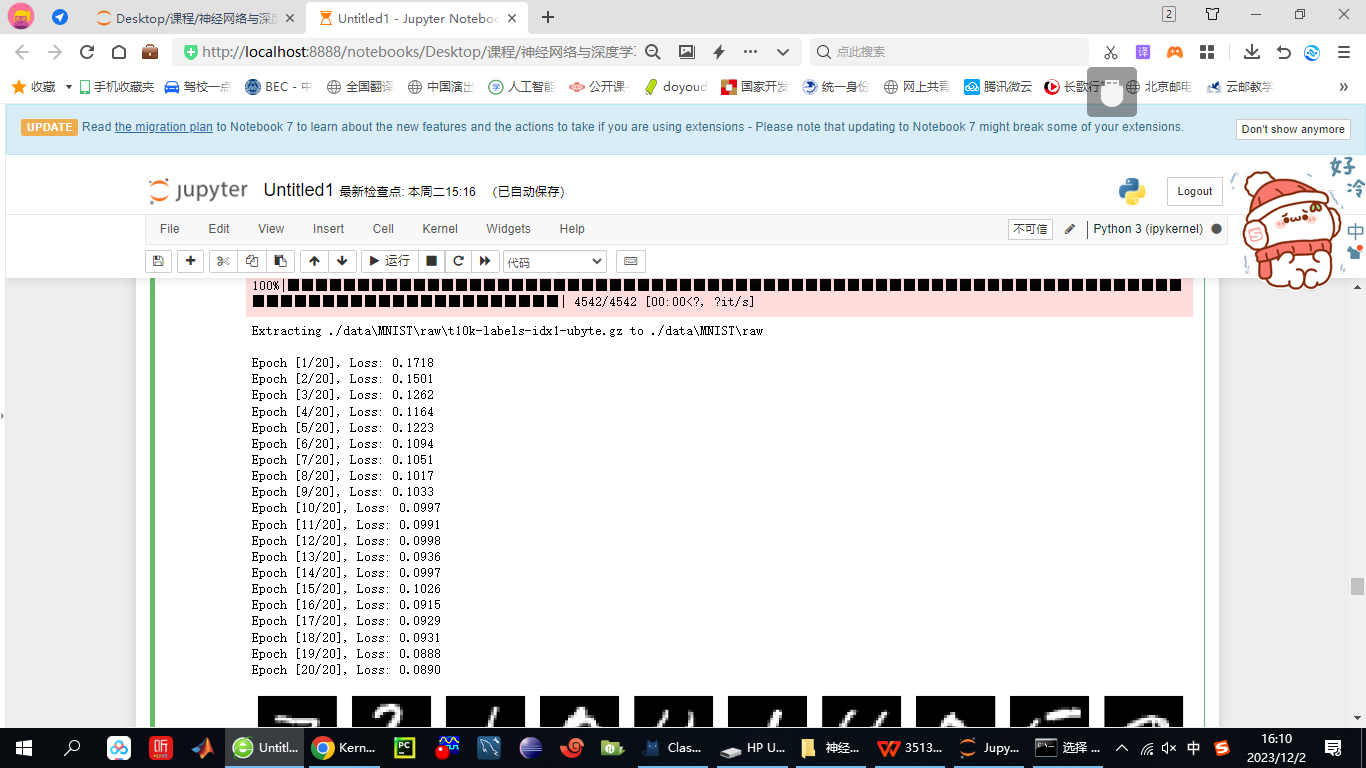
return x

注意（128.784）经过每个层的大小变化，为确保输入和输出维度一致，我们需要在forward方法开始前添加维度，结束后将该维度删除。

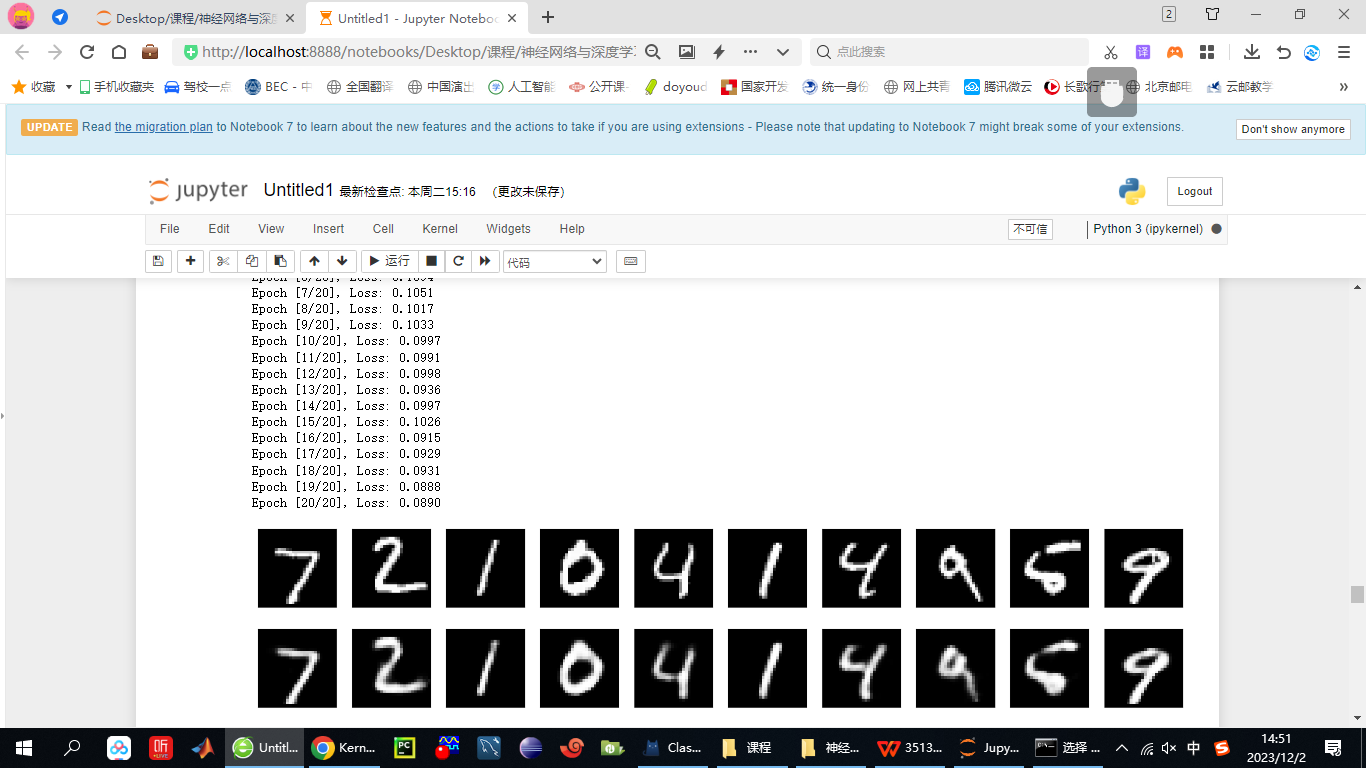
③降噪自编码输出作为特征表示自编码器输入

1. 训练

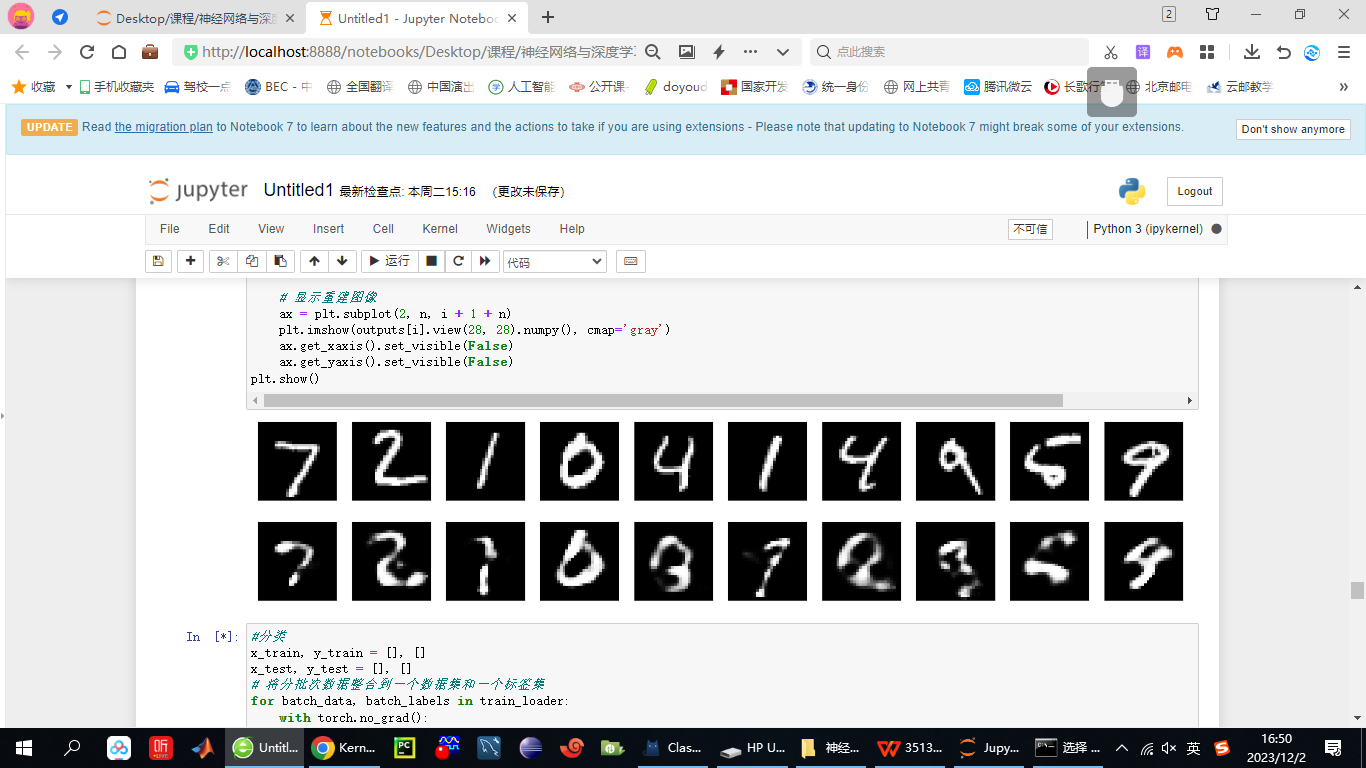
①Linear



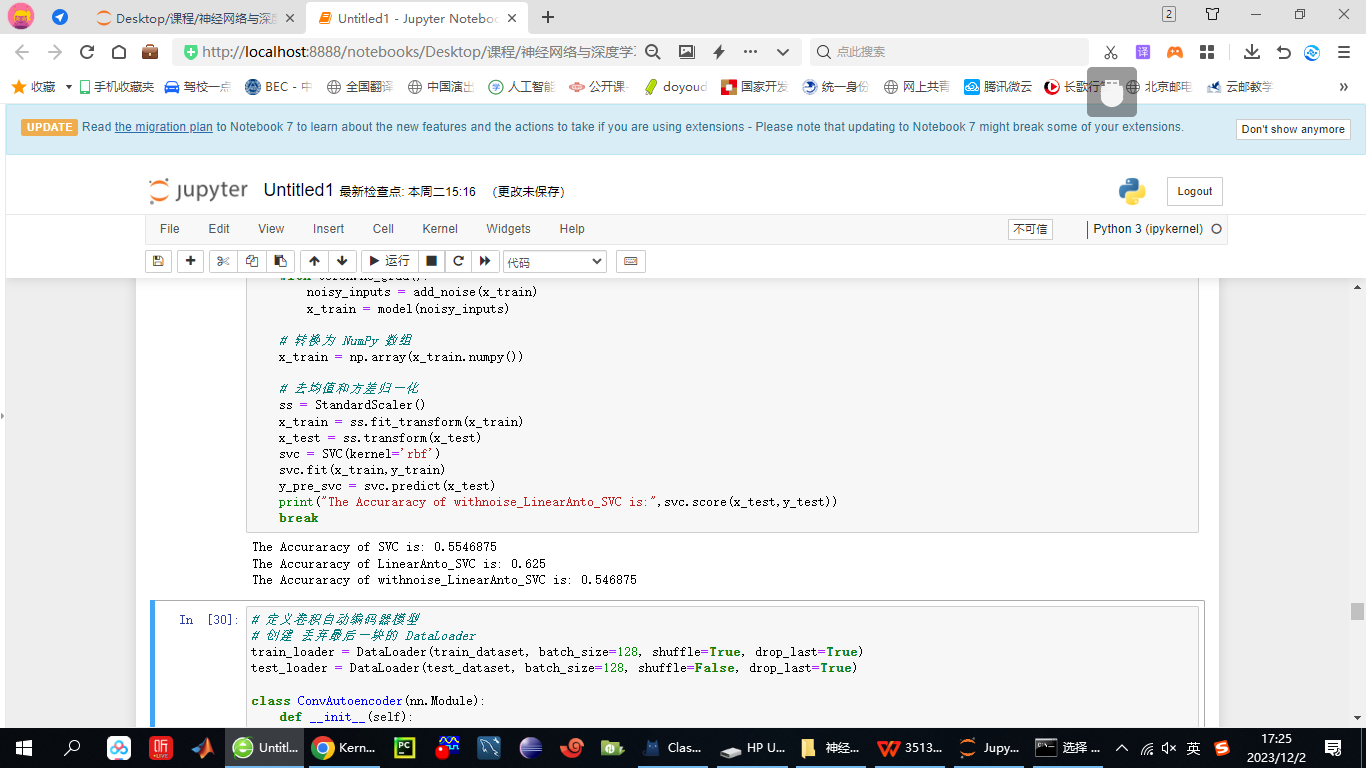
可视化：

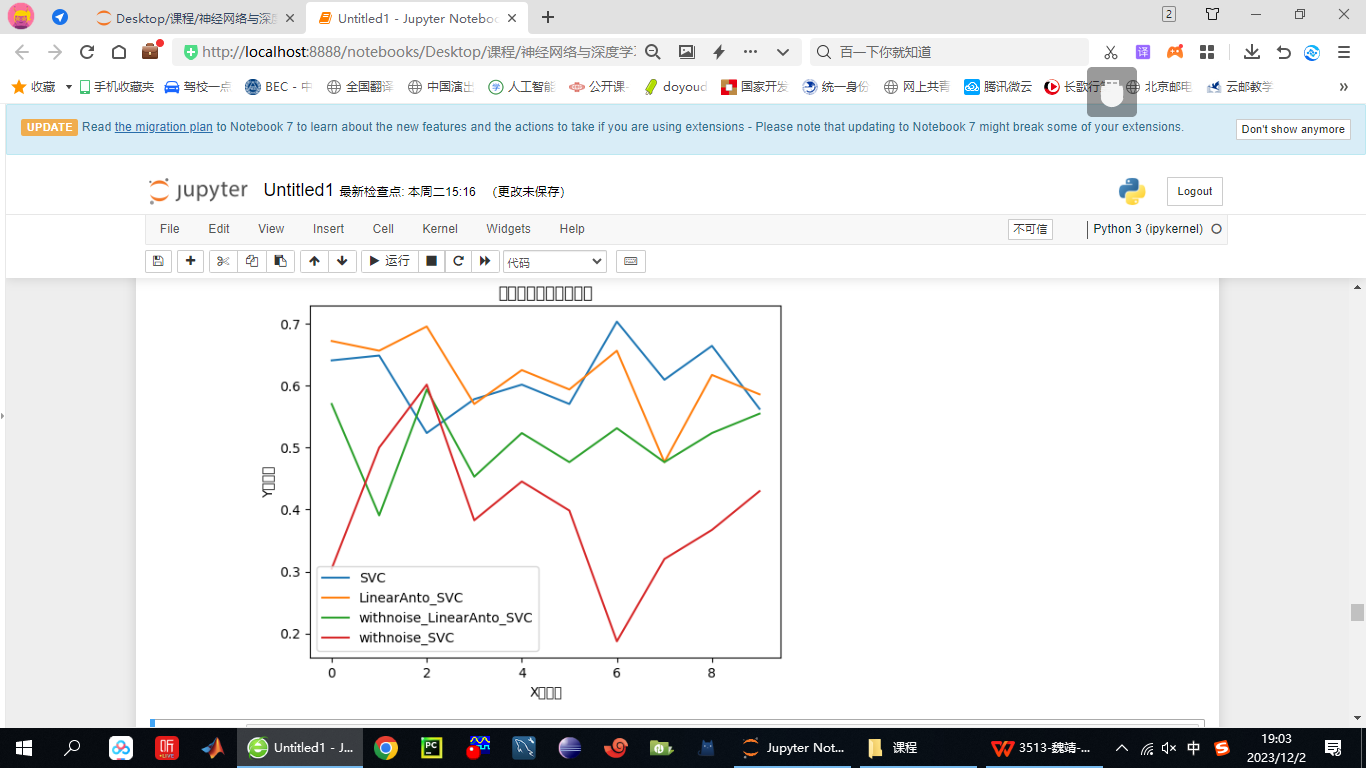


鲁棒性：（这里我们对数据加噪，再看模型特征表示的可视化结果）



聚类效果：

以其中一个batch做测试，多次运行，结果大致如下：  


若对添加噪声的数据做SVC，再观察相对分类准确度，循环十次输出图像：  


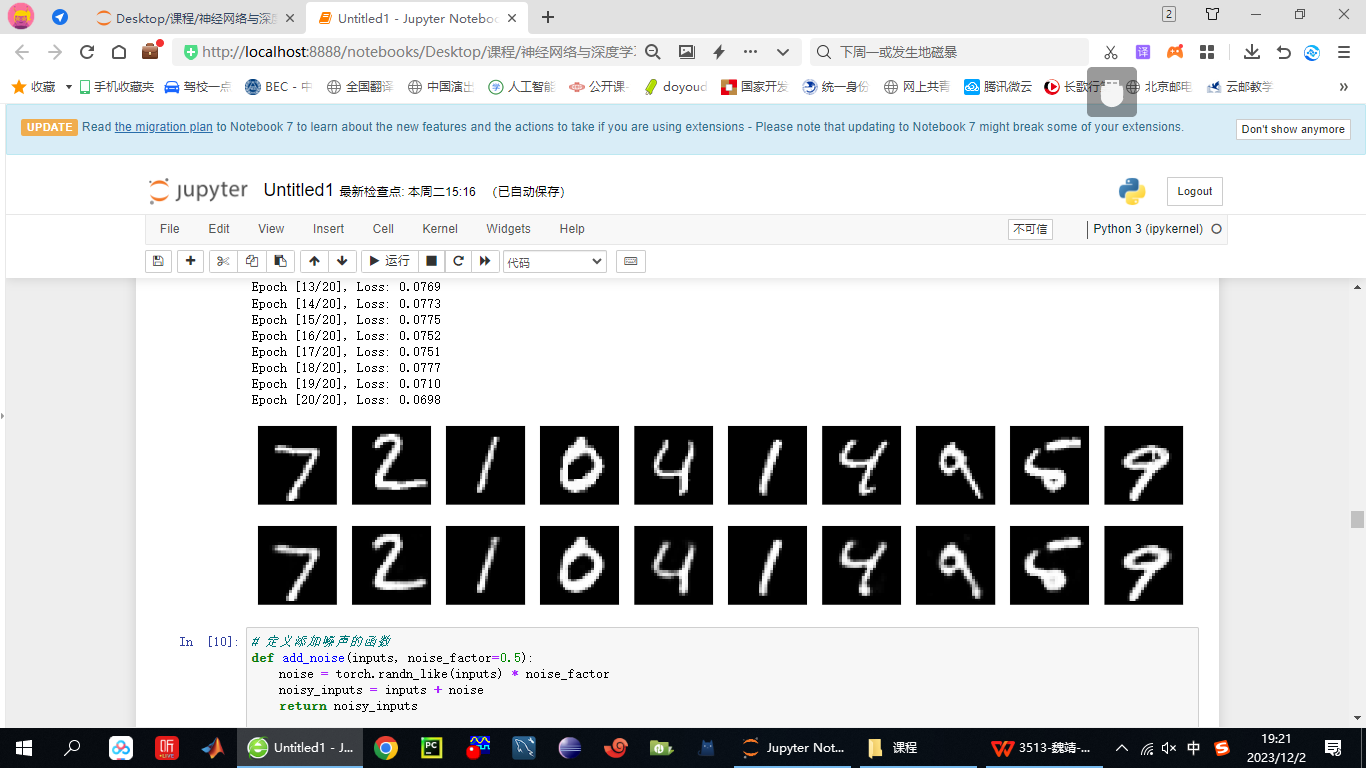
可以发现，相较不做自编码特征提取，使用全连接网络的自编码特征提取，SVC分类准确度相差不大，但在对输入数据引入噪声的情况下，后者准确率明显高于前者，且更加稳定。

基于可视化、鲁棒性和分类效果对比，可以认为该设计实现了不错的特征提取。

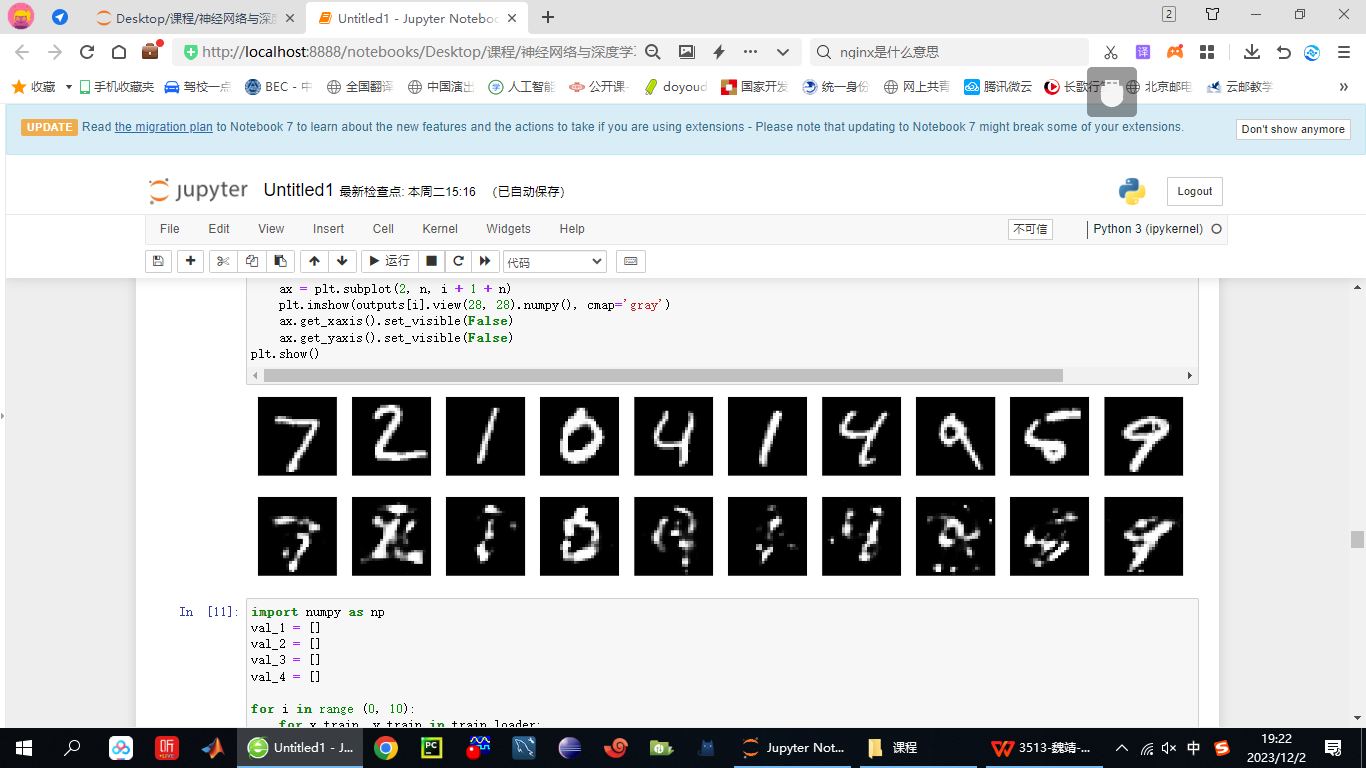
减少一层全连接：



可视化：

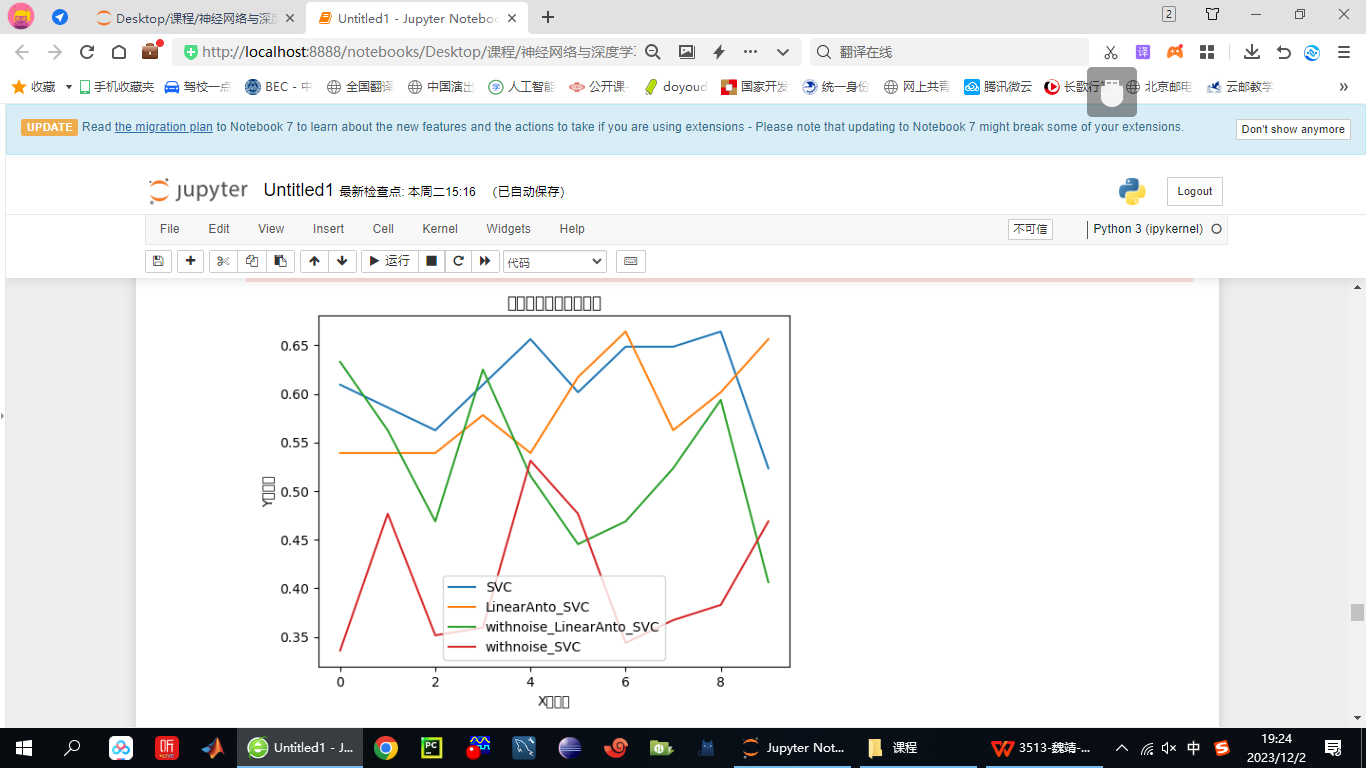


添加噪声：

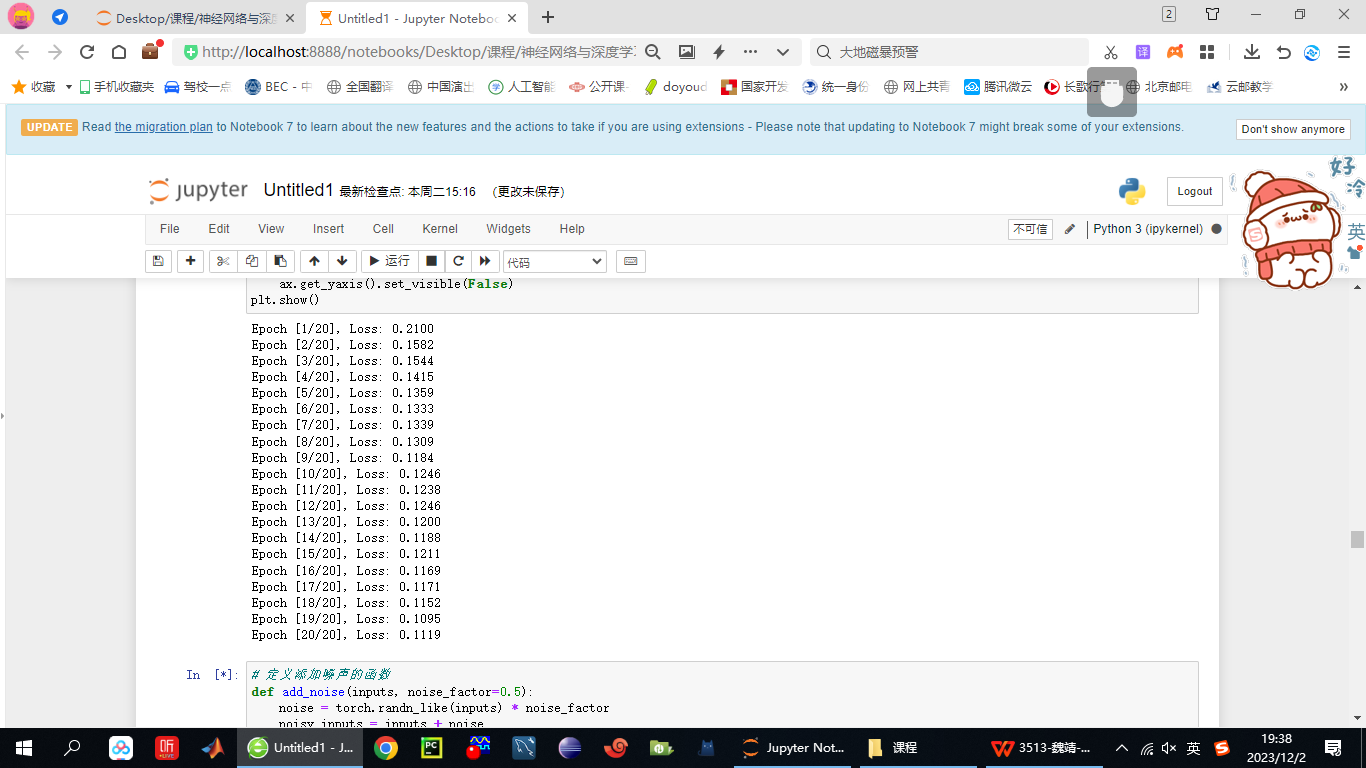


鲁棒性不如初始的三层Linear。

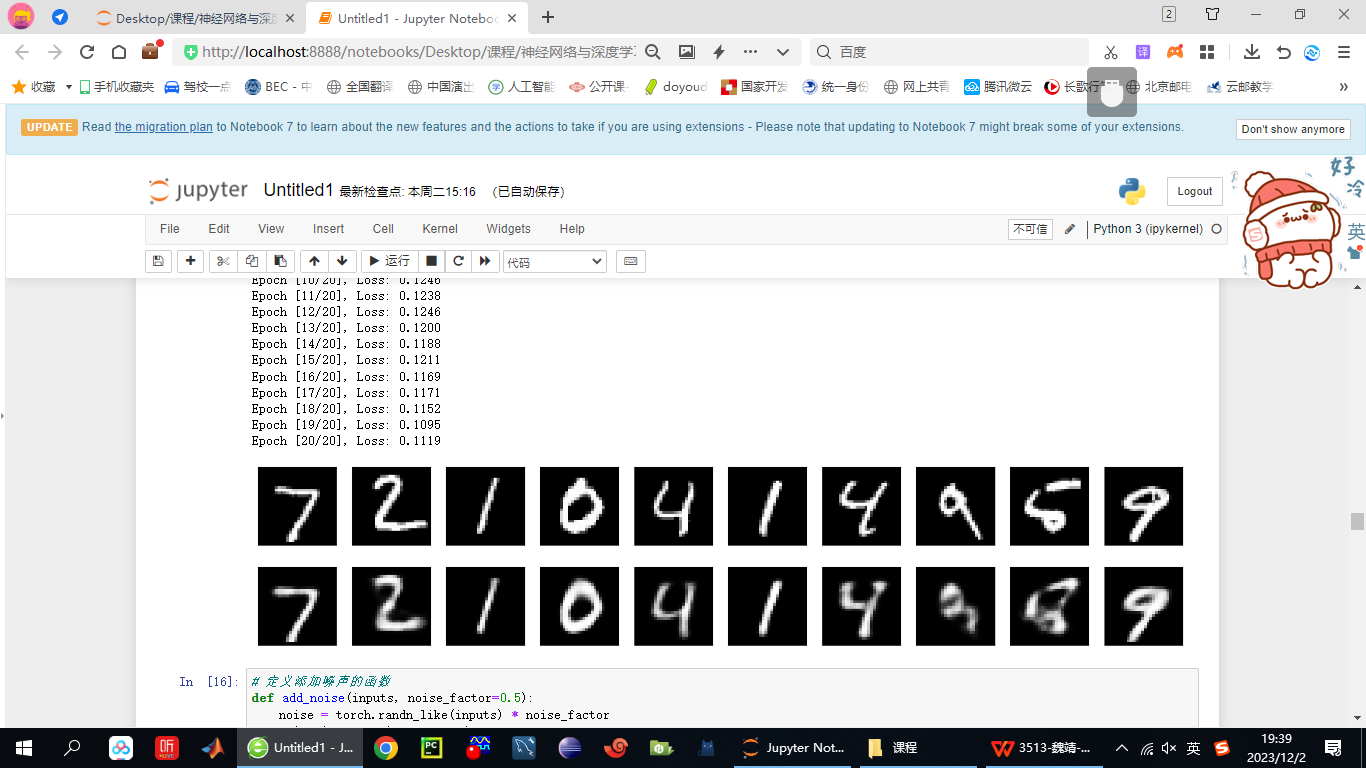
分类结果：



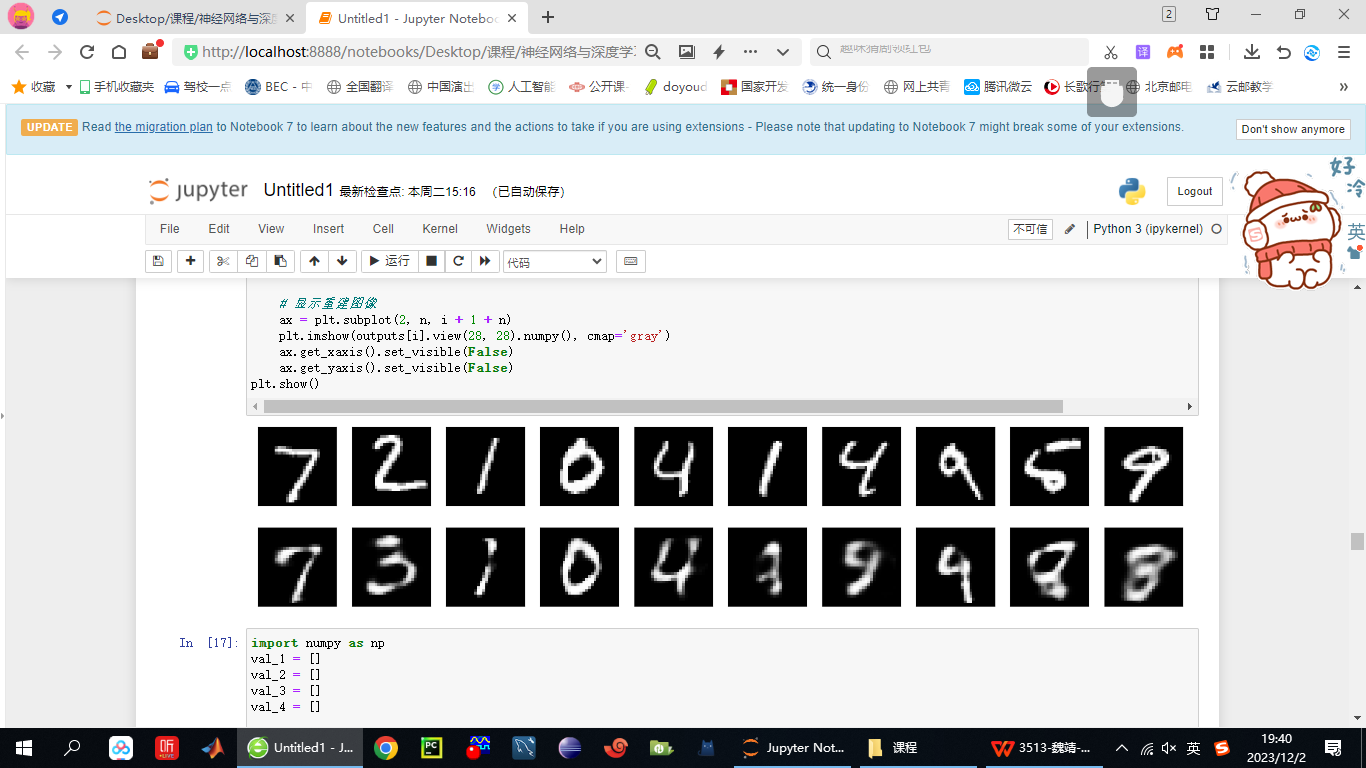
增加一层全连接：



可视化：

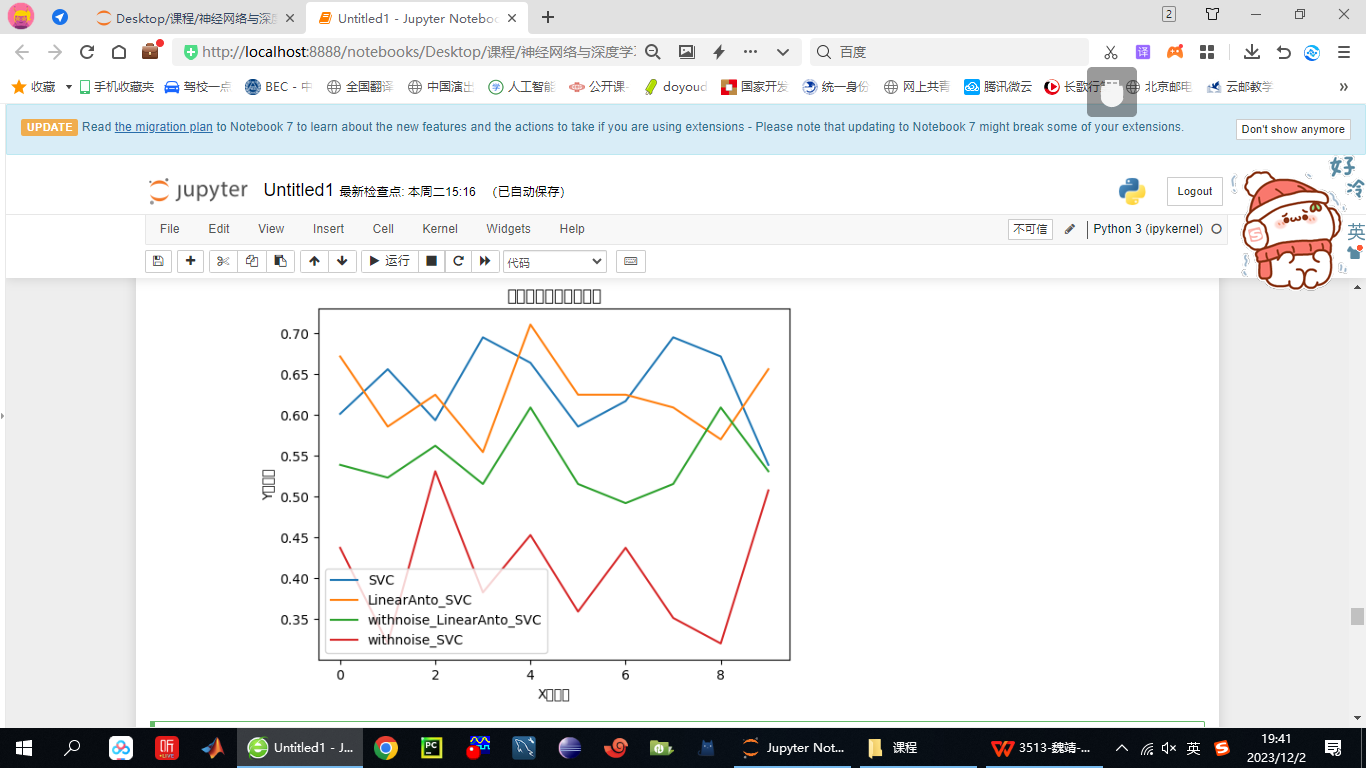


添加噪声：



鲁棒性较原始的三层Linear较差，有明显的错误（比如红圈内的9被提取特征表示为8）。

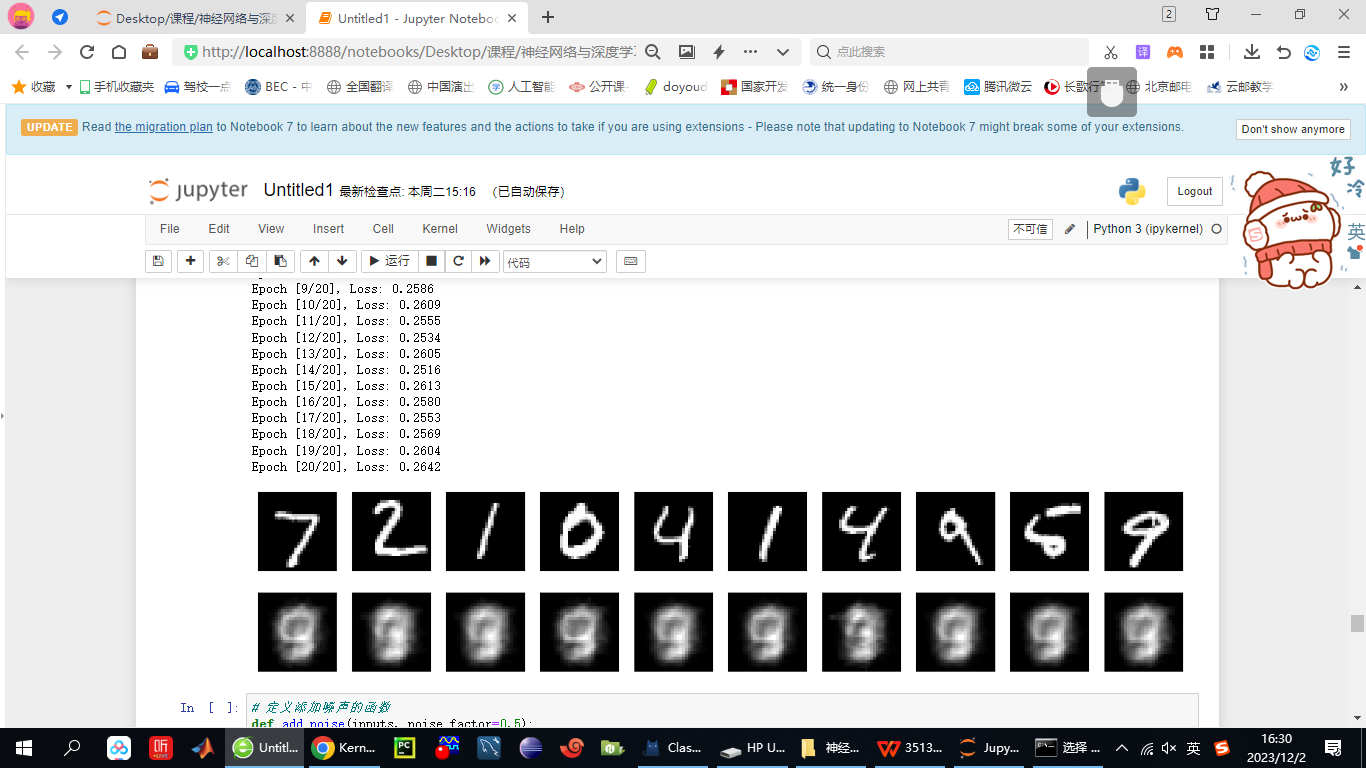
分类结果：



综上，可以认为，在简单改变Linear层数的情况下，对自编码器特征表示的效果差别不大。

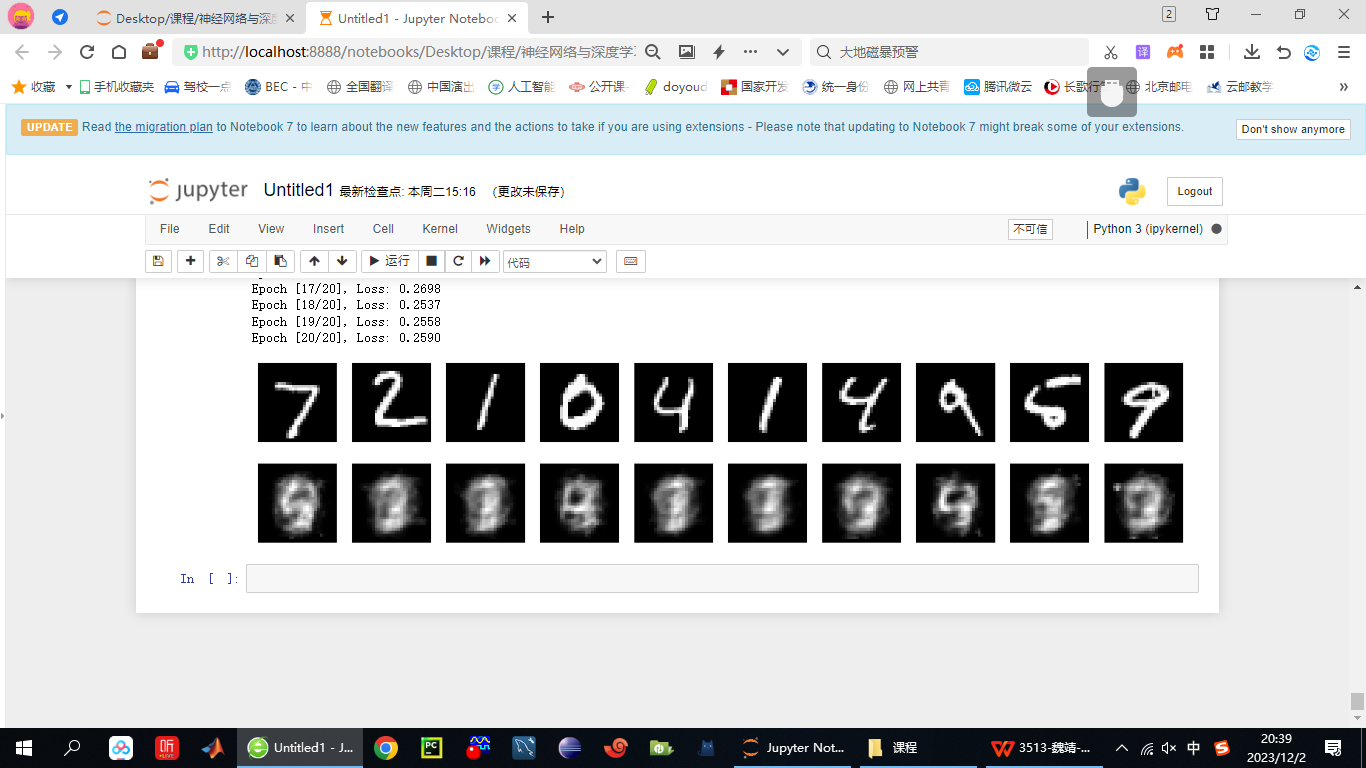
②Conv

可视化：



几乎没有实现重建，可能是conv结构或者损失函数的问题，原设计每次只有通道数变化，导致模型无效学习。

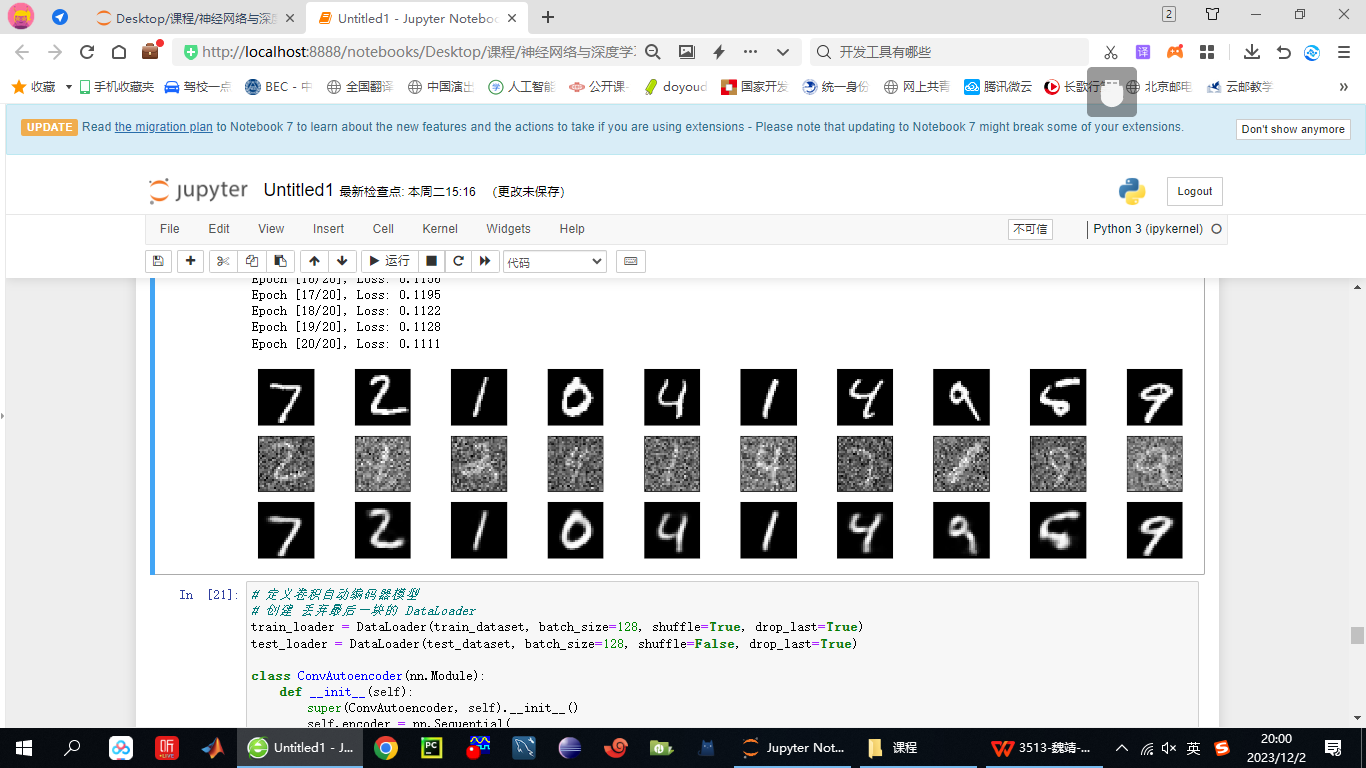
第二层kernel size变为2，重建效果稍好一点，但没有实质性的提升。



③

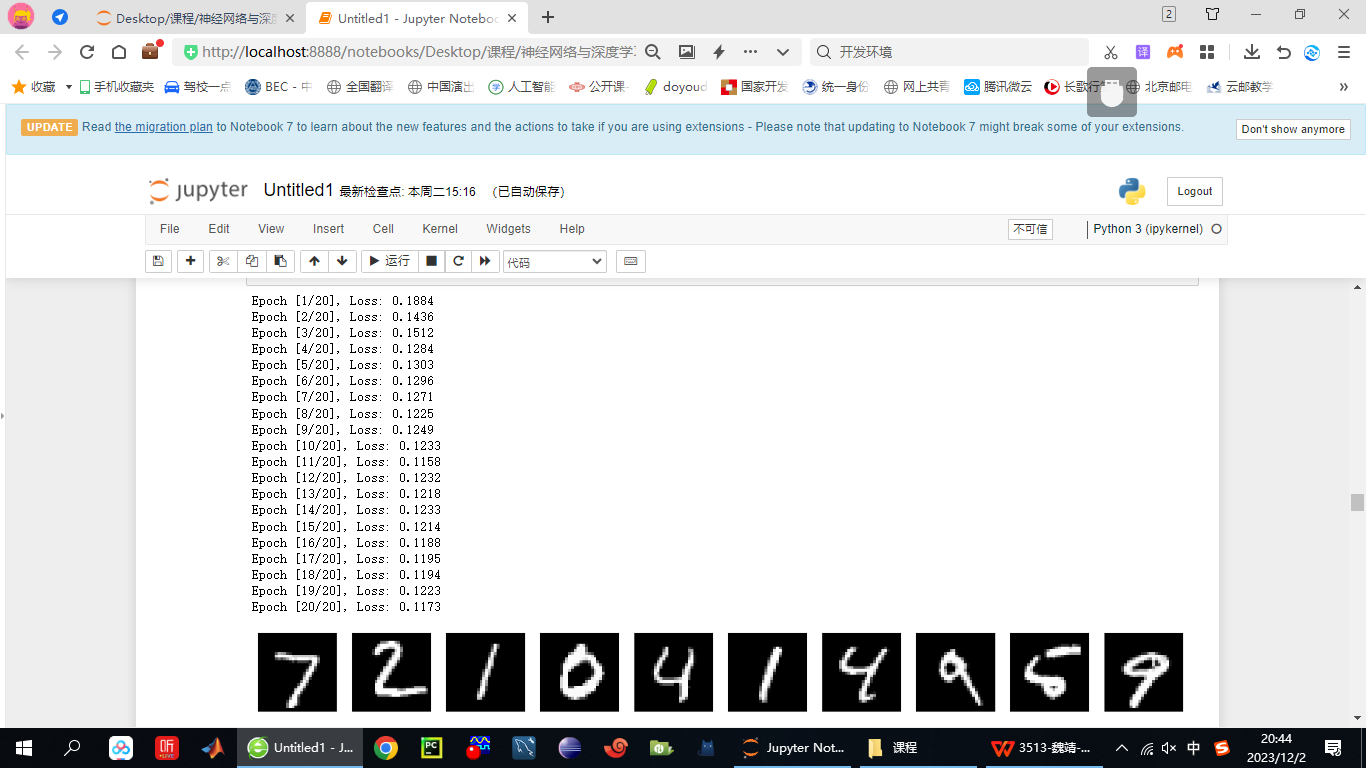
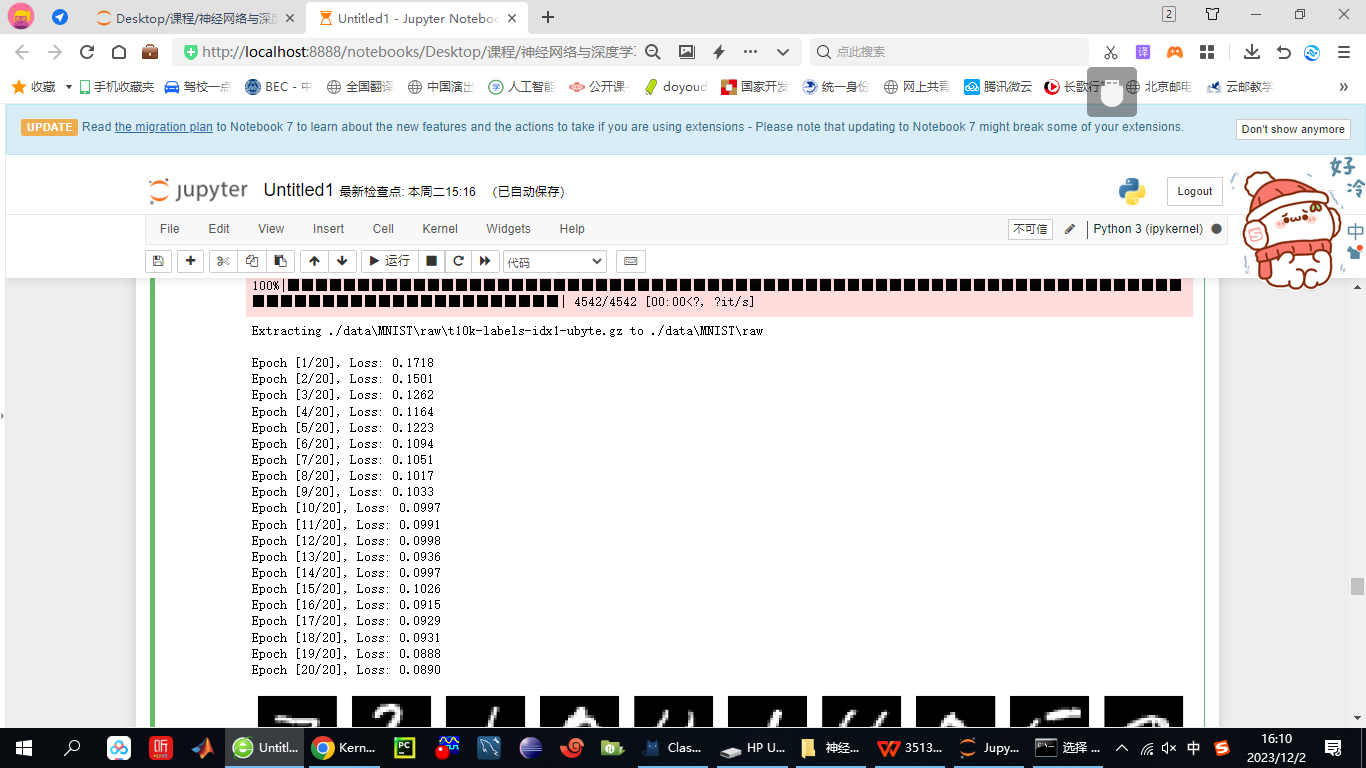
降噪自编码：

输入加噪后的图片去拟合原图片，用原图片作为测试查看去噪自编码的去噪效果：

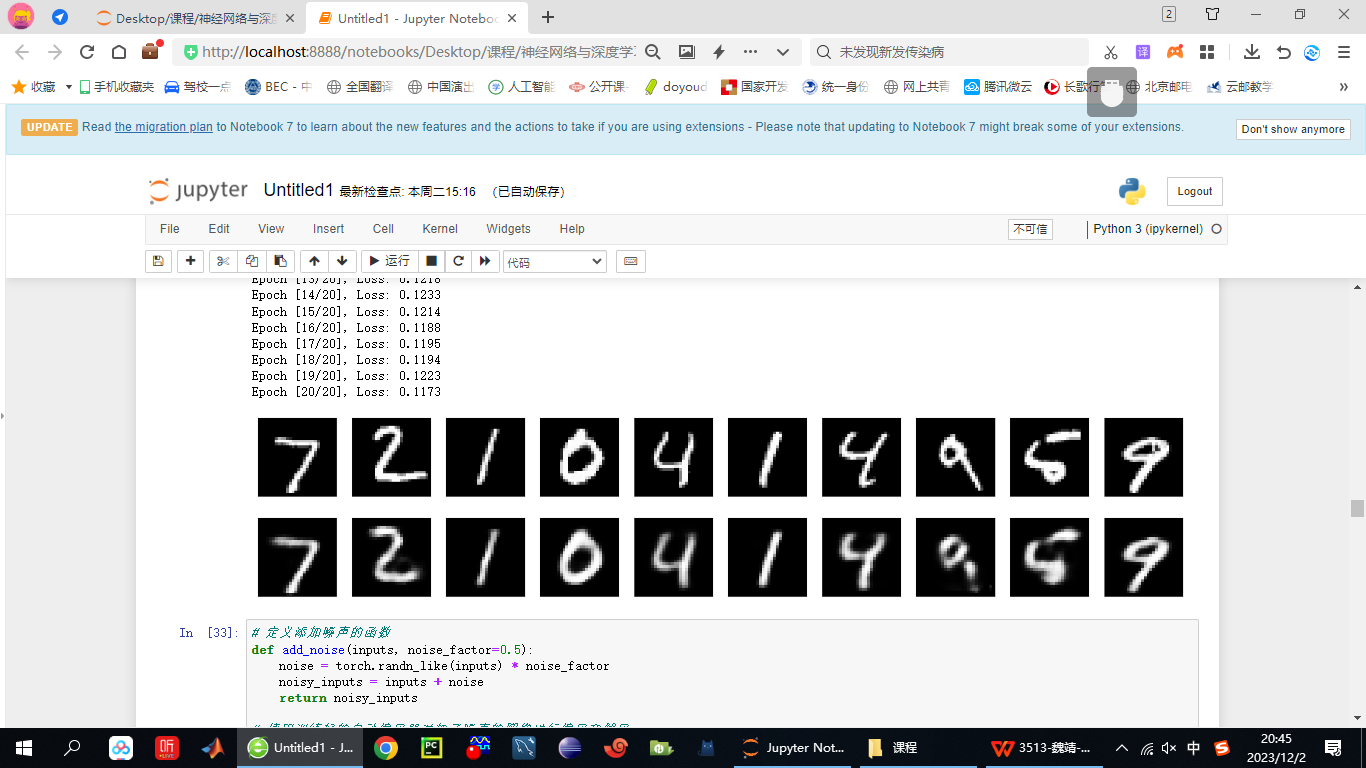
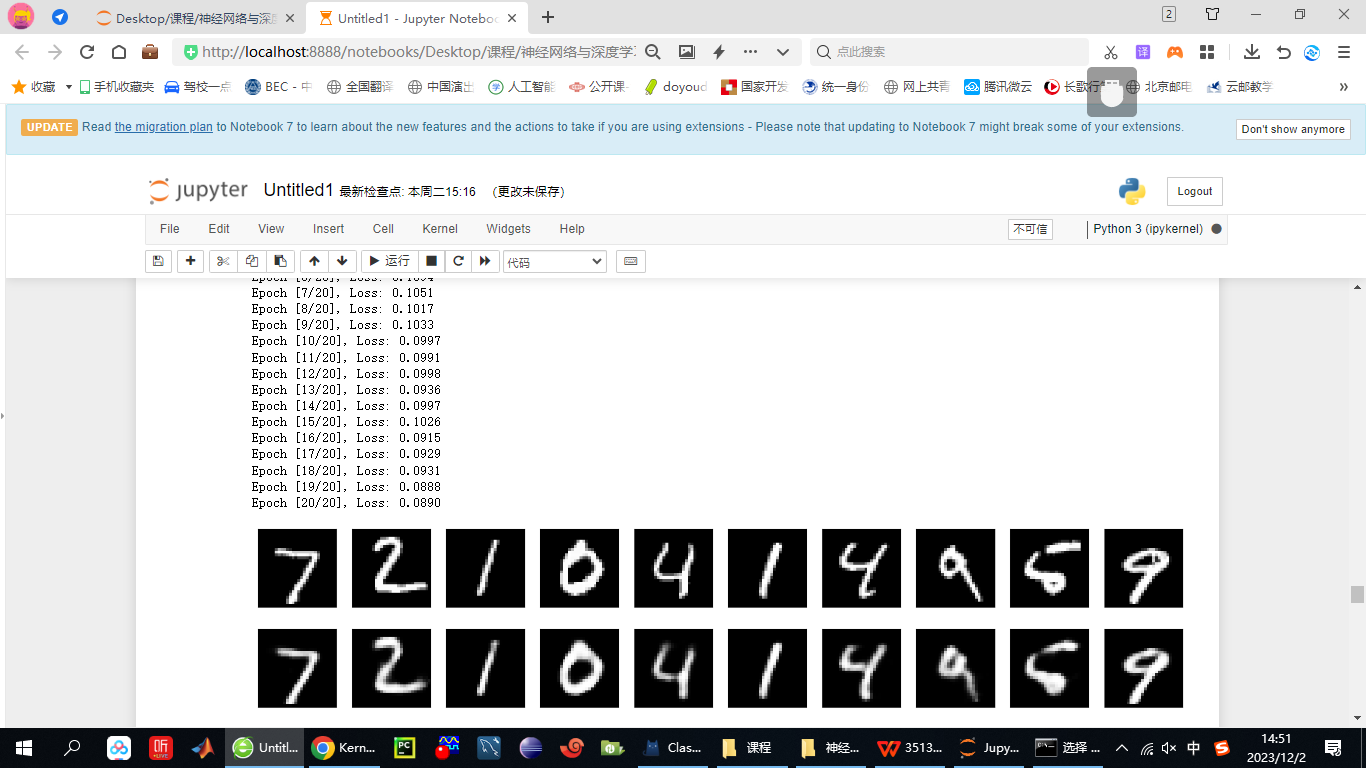


在原始Linear特征表示自编码器结构基础上，降噪自编码输出作为输入训练新的网络：

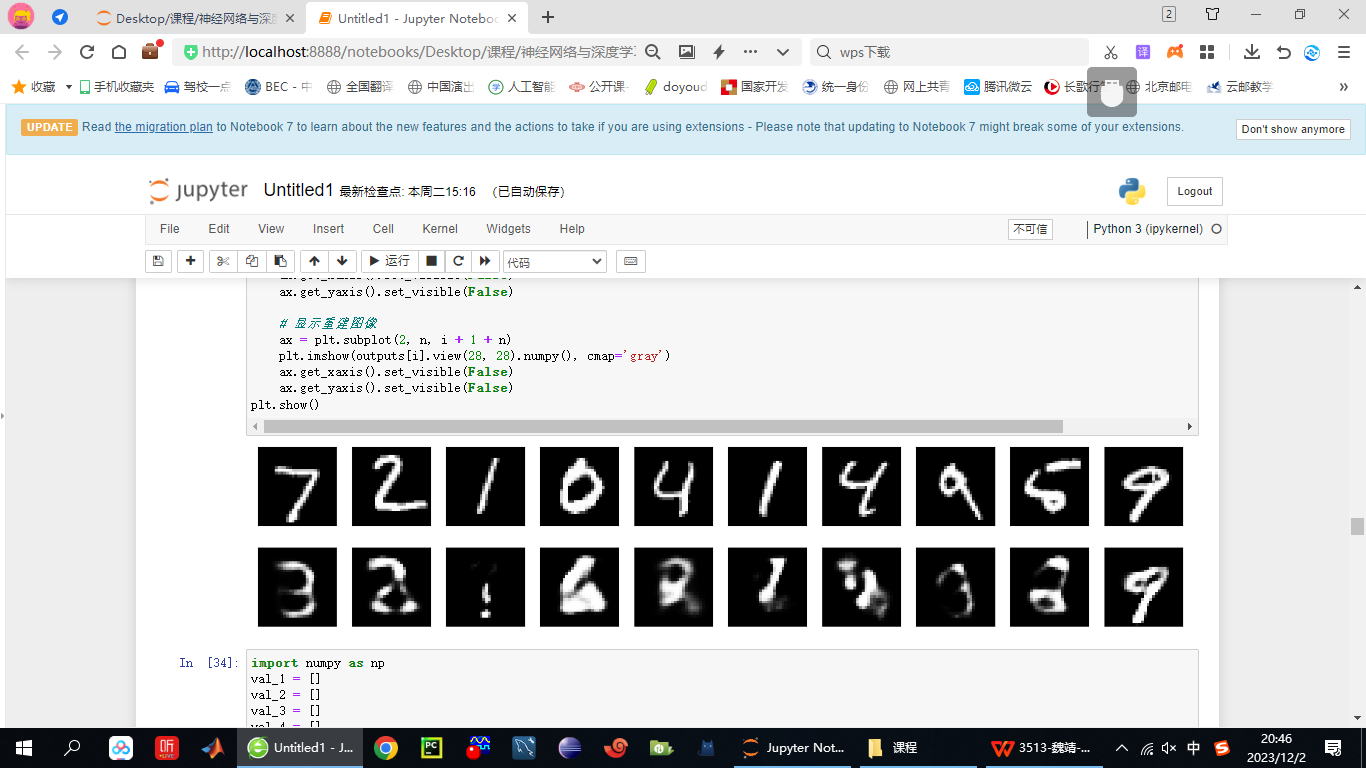
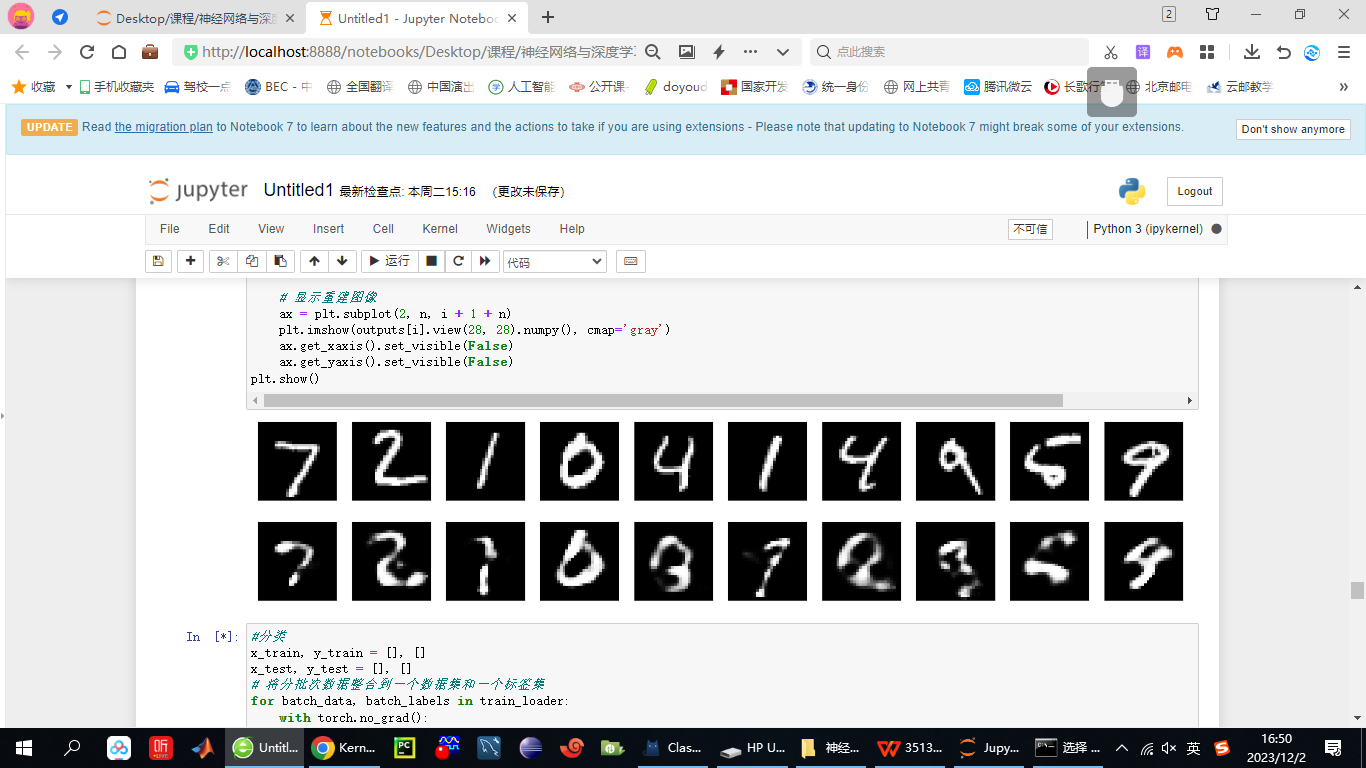
分别左右对比分析：



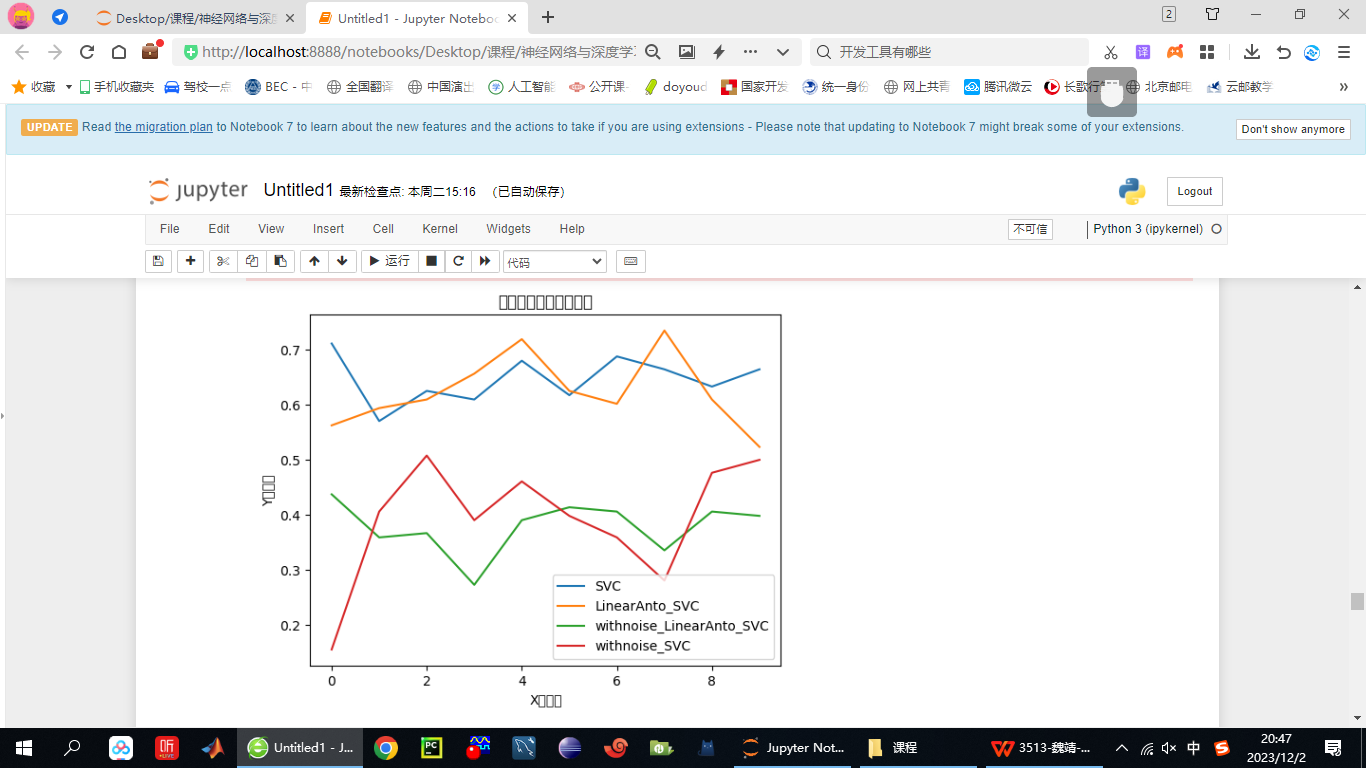
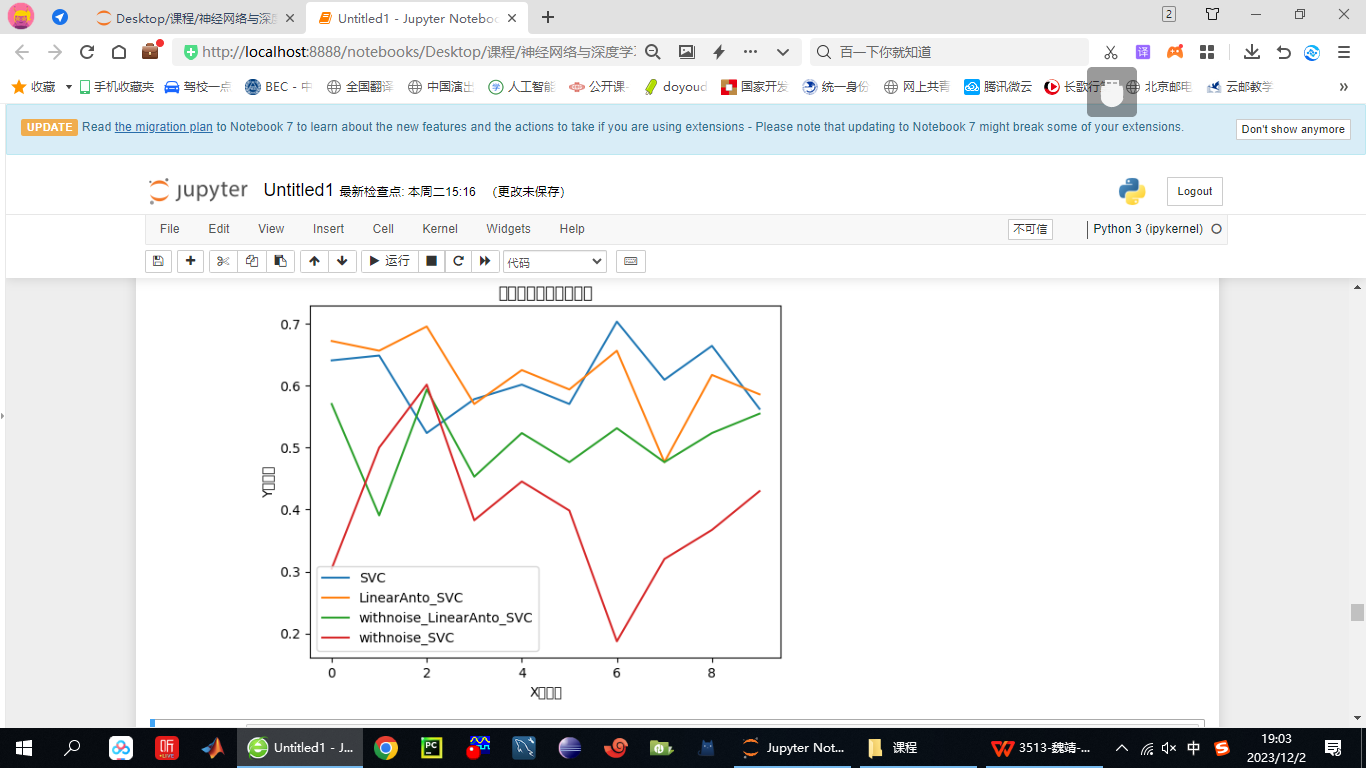
可视化：



鲁棒性：（这里我们对数据加噪，再看模型特征表示的可视化结果）



聚类效果：



1. 结果

可以认为，在没有实现Conv的情况下，原始Linear自编码器的效果更好。

4、总结与反思

PCA能够提取主成分，但用在分类时处理的效果不一定有不提取好；自动编码器学到的特征表示能够在保留主要信息的同时减少输入的维度，与PCA相比，自动编码器具有更强的非线性建模能力，能够更好地捕捉数据中的复杂结构。总而言之，特征提取之后的数据在面对干扰时的稳定性更强，鲁棒性更好。

通过本次设计，我学到了降维和特征学习的基本概念，强化了对主成分分析的认知，理解并掌握了自编码器的原理，深入理解了不同方法对于数据表征和分类的影响，体会到了无监督学习自己训练自己的奇妙之处，对以上原理的认识为今后的图像特征提取和降噪、压缩等任务提供更丰富的思路和解决方案，也为正在进行的课程设计提供了启发。