实践作业1——试用Tensorflow Playground

Tensorflow Playground是google发布的机器学习开源平台，我们可以在浏览器中可视化的训练自己的神经网络。

由下图1可以很好的看到Tensorflow Playground的组成。首先框1代表的是训练神经网络的epoch，可以通过左侧的按钮开始和暂停。框2中可以设置一些训练神经网络的参数，例如Learning rate，Activation，Regularization，Regularization rate，Problem type。可以通过设置不同的参数或者函数，来对比他们之间的区别。框3中可以设置数据，初始情况选择的是分类任务的话，会有四种分类数据可供选择，下面还可以调节训练集和测试集的比例，对数据添加噪声，调整每个epoch的batch size大小。右侧的框4和框5可以设置神经网络的结构，分别可以选择6种不同的特征提取操作，和添加不同深度，不同数量的神经元。框6则可以最终可视化的显示训练结果，并且还可以勾选show test data显示未参与训练的test数据集的情况，勾选distretize output可以看到离散化后的结果。

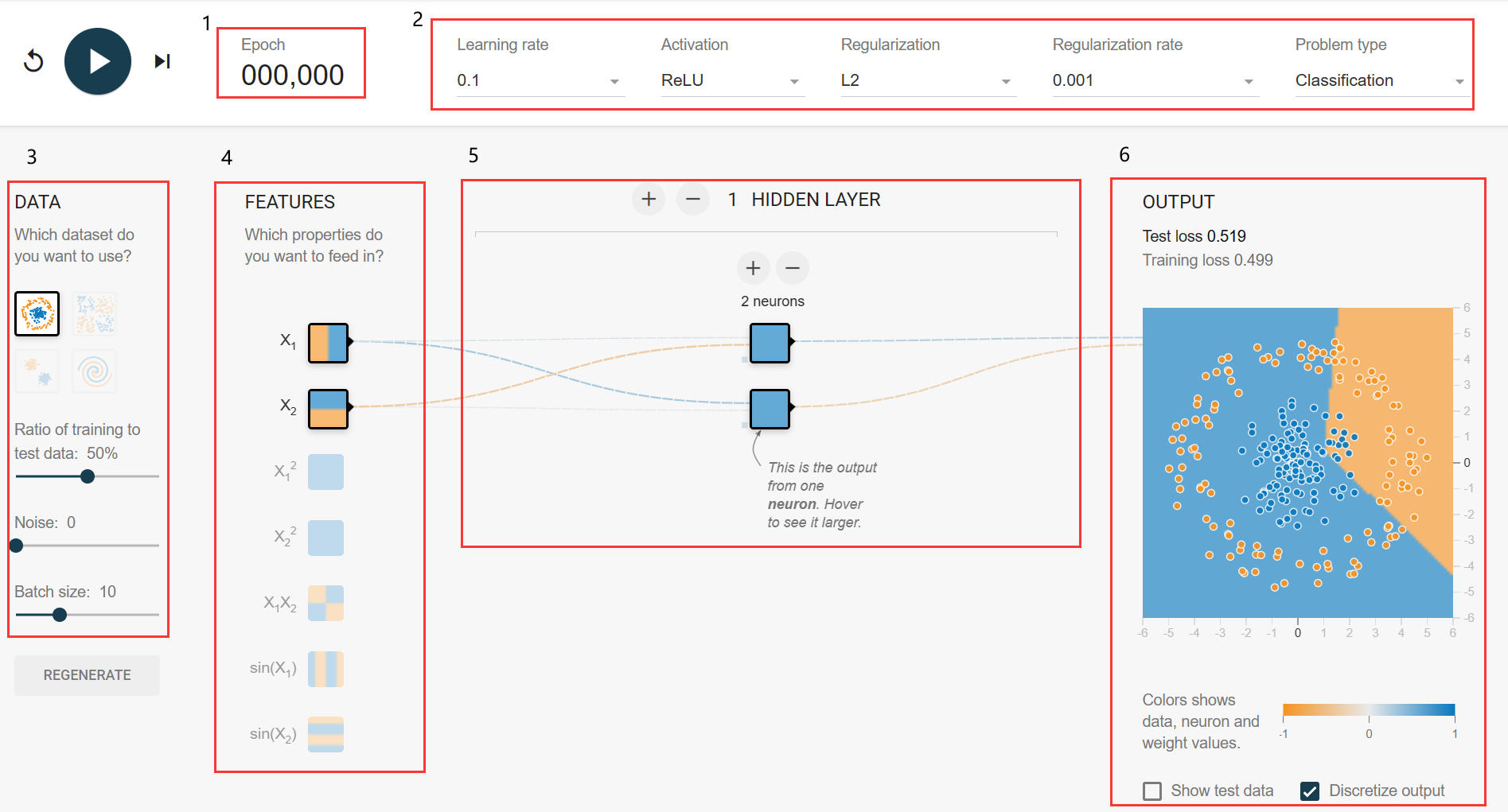


图1 Tensorflow Playground的构成

对于分类任务，数据一可以只提取横向和纵向的特征，即可以很好的分类出来。我们在采用如图2所示的参数和神经网络时，在训练200个epoch左右，就可以发现损失曲线已经可以收敛，并且达到了很不错的效果。

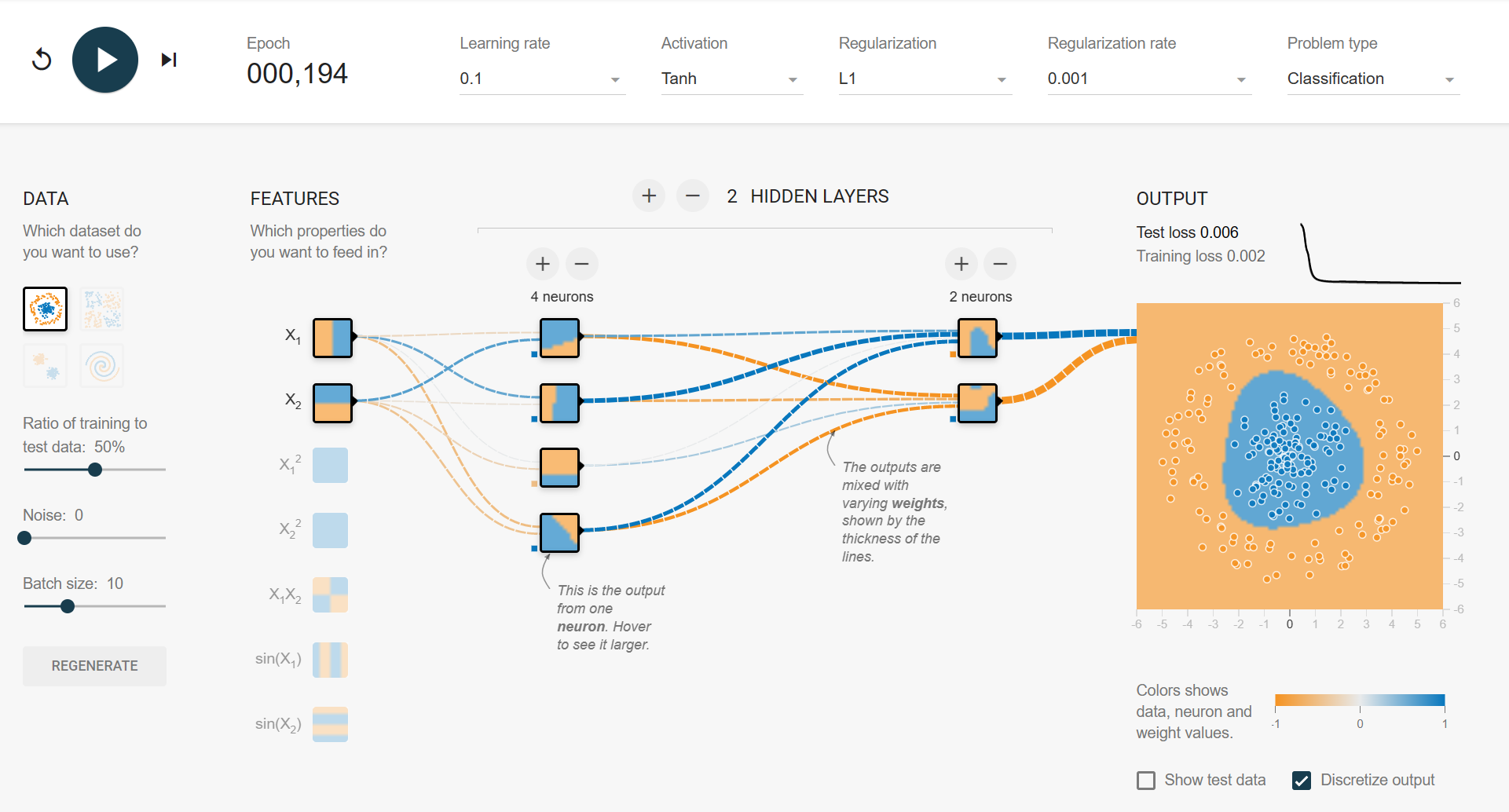


图2 数据一结果展示

对于数据二，如果使用X1X2特征提取，可以发现在训练很少的epoch下达到很好的效果。

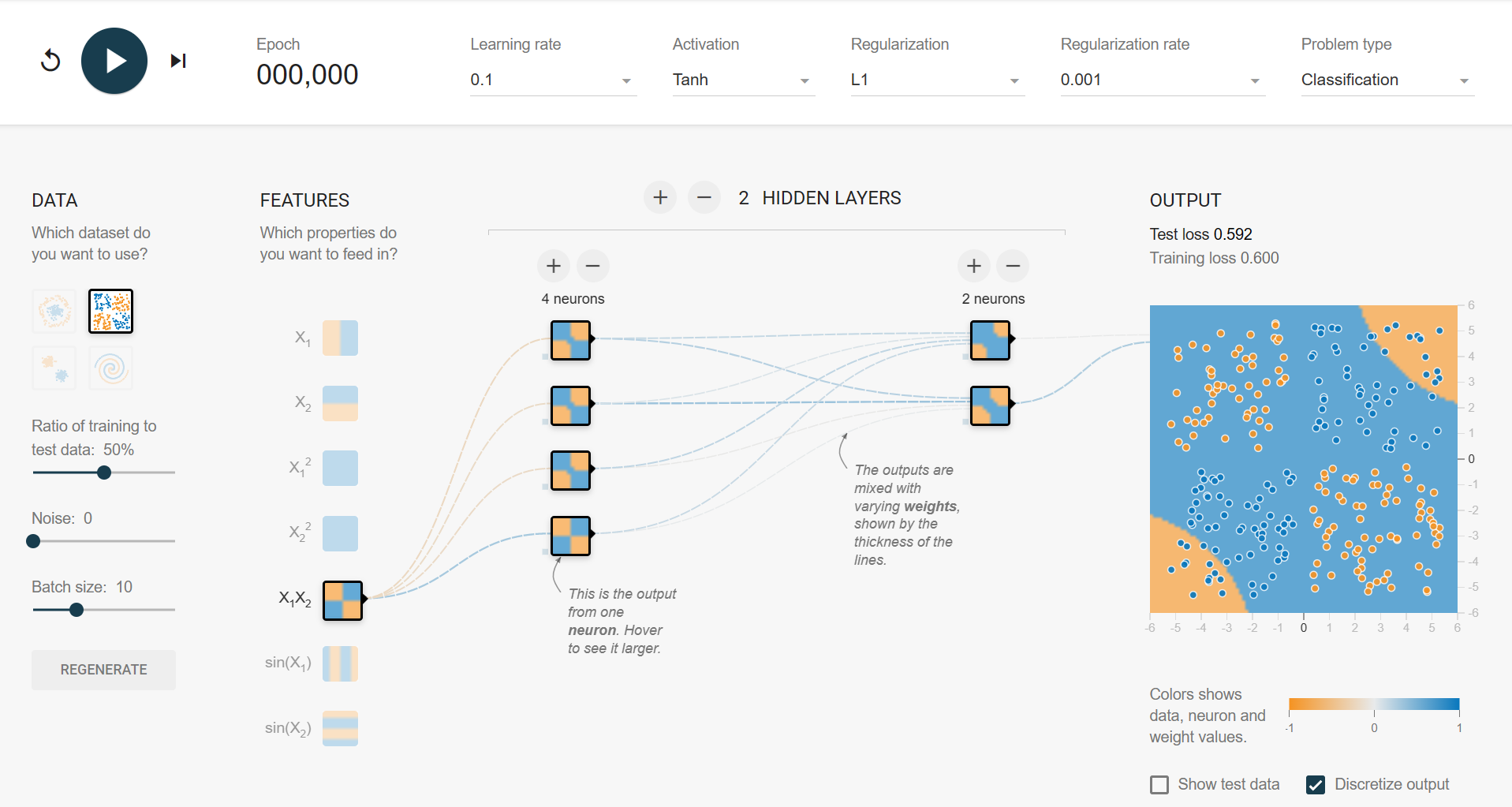


图3 数据二结果展示

对于数据三，最后的实验结果如下图所示。

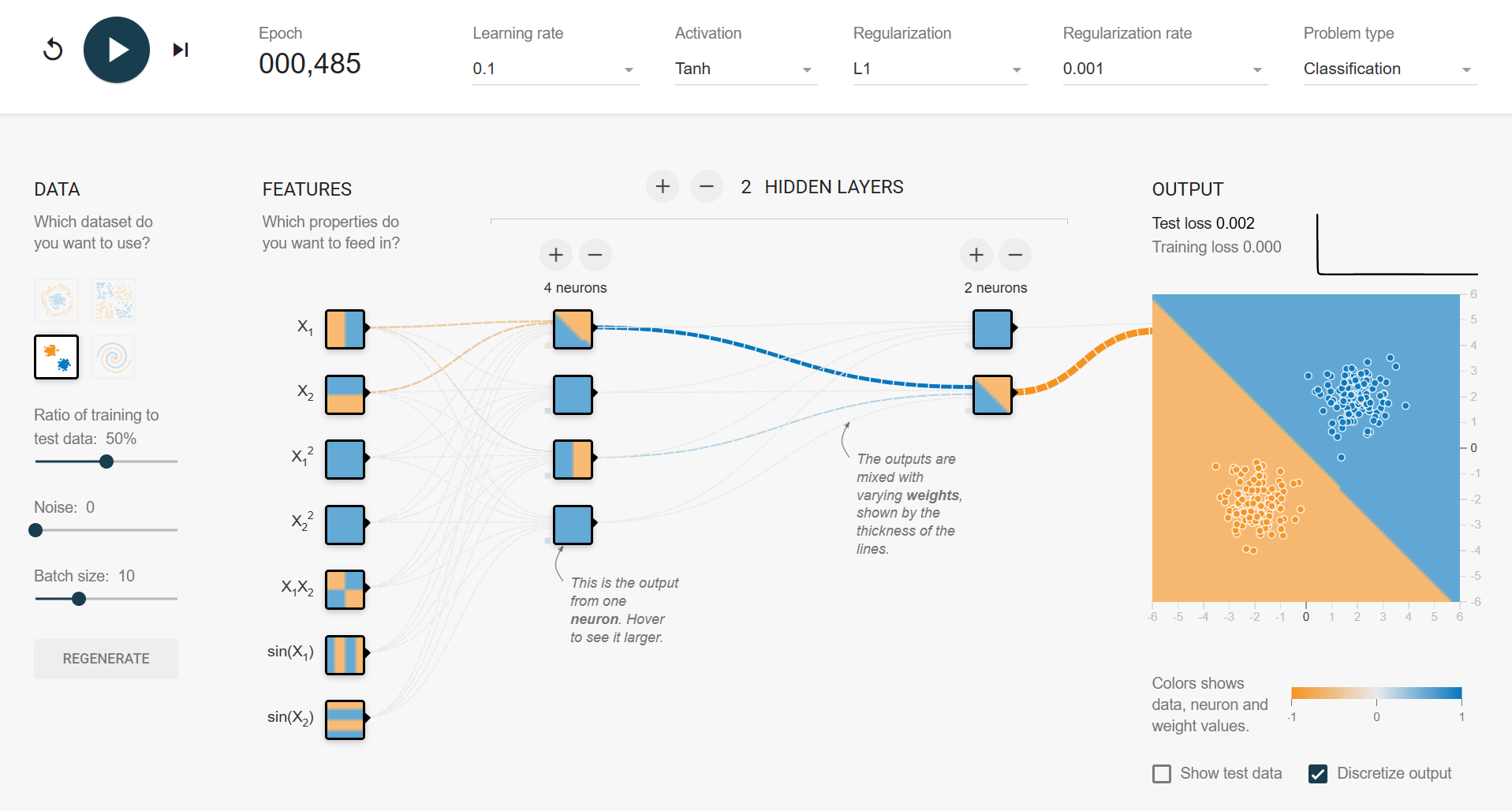


图4 数据三结果展示

对于数据四，我认为是这四个数据中最难处理的数据，因此在这里采用两种方法处理，一种将所有的7个特征都输入系统，并且只选择一层的神经网络。并且由于数据四的数据分布情况，将最后一层的输出设置成8，这样可以更好的对数据四进行分类。结果如下图所示。可以发现在200个epoch左右，就可以达到收敛的效果。

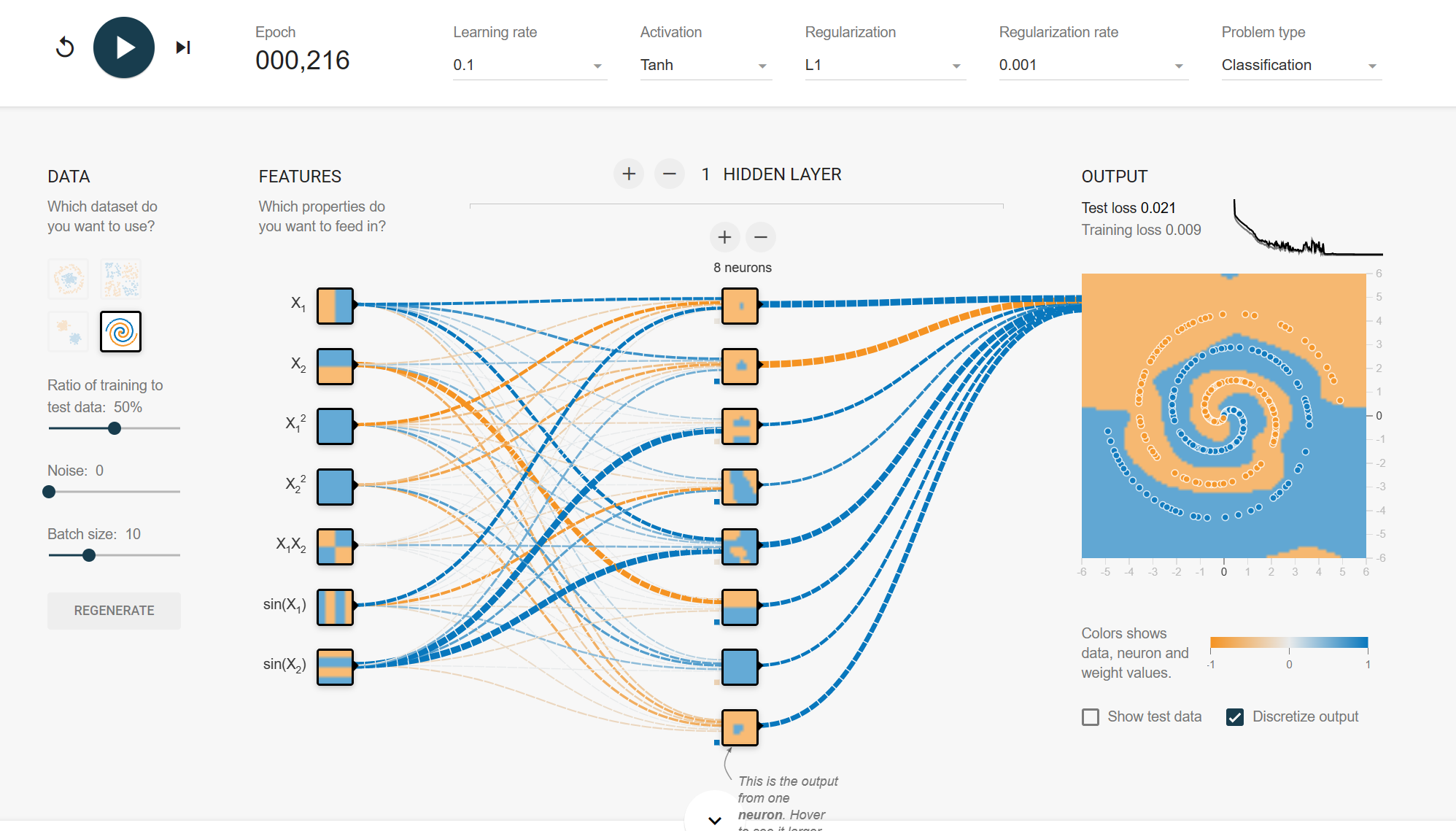


图5 数据四的其中一种结果展示

对于数据四采取的第二种处理方法，我们不需要使用各种各样的特征，用来输入给神经网络系统，通过设置足够多的神经网络和神经元。结果如下图所示。可以发现，在输入很少的特征时，经过很多层神经网络之后，也同样可以实现很好的分类效果。

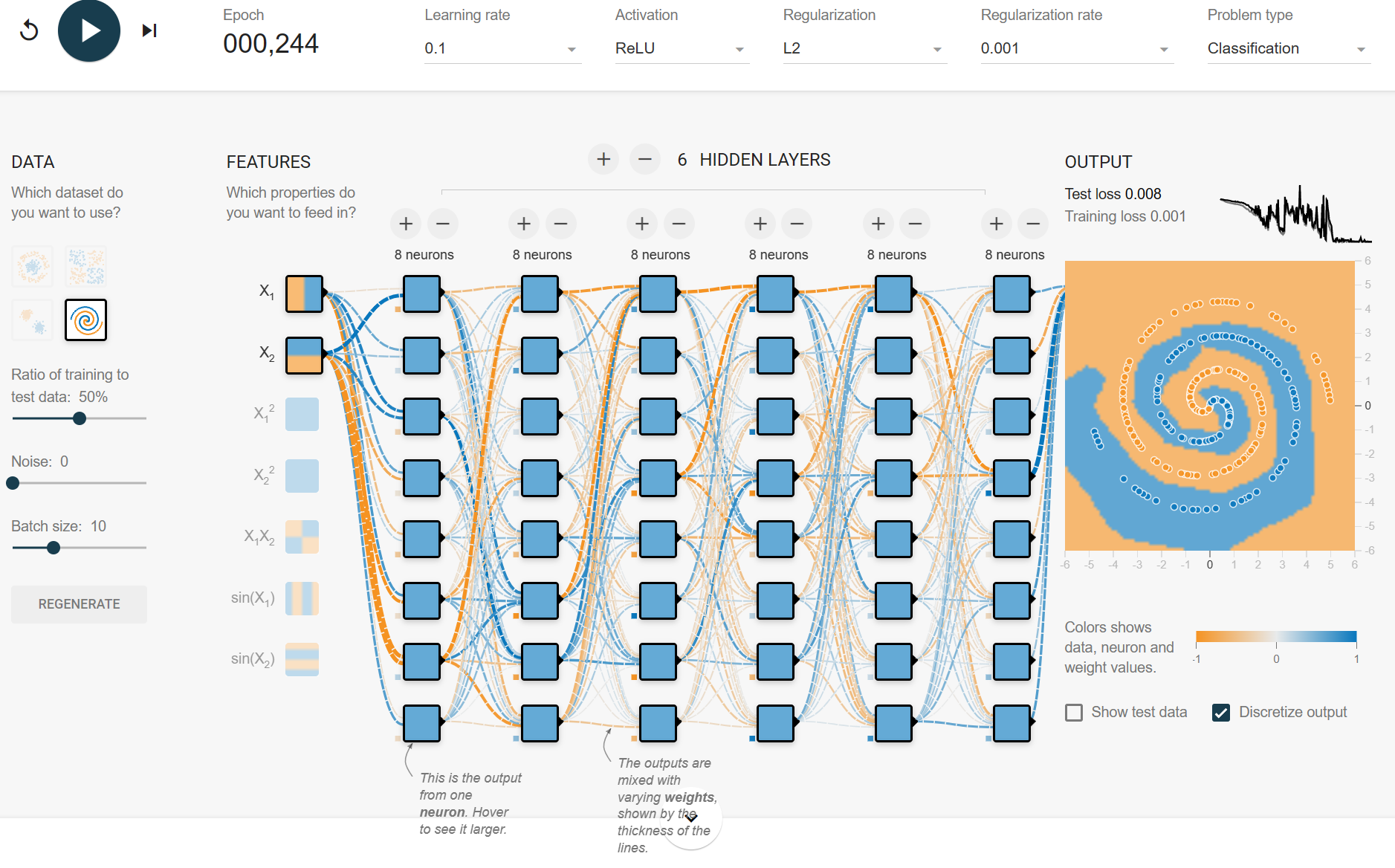


图6 数据四的深层神经元结果展示

实践作业二——使用LeNet-5在MNIST数据集上训练

1. LeNet-5基本结构

LeNet-5的基本结构包括7层网络结构（不含输入层），其中包括2个卷积层、2个降采样层（池化层）、2个全连接层和输出层。

1、输入层（Input layer）

输入层接收大小为32\*32的手写数字图像，包括灰度值（0-255）。

2、卷积层C1（Convolutional layer C1）

卷积层C1包括6个卷积核，每个卷积核的大小为5\*5，步长为1，填充为0。每个卷积核会产生一个大小为28\*28的特征图（输出通道数为6）。

3、采样层（Subsampling layer S2）

采样层S2采用最大池化（max-pooling）操作，每个窗口的大小为 2×2 ，步长为2。因此，每个池化操作会从4个相邻的特征图中选择最大值，产生一个大小为 14×14 的特征图（输出通道数为6）。这样可以减少特征图的大小，提高计算效率，并且对于轻微的位置变化可以保持一定的不变性。

1. 卷积层C3（Convolutional layer C3）

卷积层C3包括16个卷积核，每个卷积核的大小为 5×5 ，步长为1，填充为0。因此，每个卷积核会产生一个大小为 10×10 的特征图（输出通道数为16）。

1. 采样层S4（Subsampling layer S4）

采样层S4采用最大池化操作，每个窗口的大小为 2×2 ，步长为2。因此，每个池化操作会从4个相邻的特征图中选择最大值，产生一个大小为 5×5 的特征图（输出通道数为16）。

1. 全连接层C5（Fully connected layer C5）

C5将每个大小为 5×5 的特征图拉成一个长度为400的向量，并通过一个带有120个神经元的全连接层进行连接。120是由LeNet-5的设计者根据实验得到的最佳值。

1. 全连接层F6（Fully connected layer F6）

全连接层F6将120个神经元连接到84个神经元。

1. 输出层（Output layer）

输出层由10个神经元组成，每个神经元对应0-9中的一个数字，并输出最终的分类结果。在训练过程中，使用交叉熵损失函数计算输出层的误差，并通过反向传播算法更新卷积核和全连接层的权重参数。

1. 训练过程

读入MNIST数据集，并将其以dataloader的格式读入。定义好epoch，lr，optimizer，criterion。之后便可以开始训练。

1. 训练结果截图

损失的变化如下图所示。

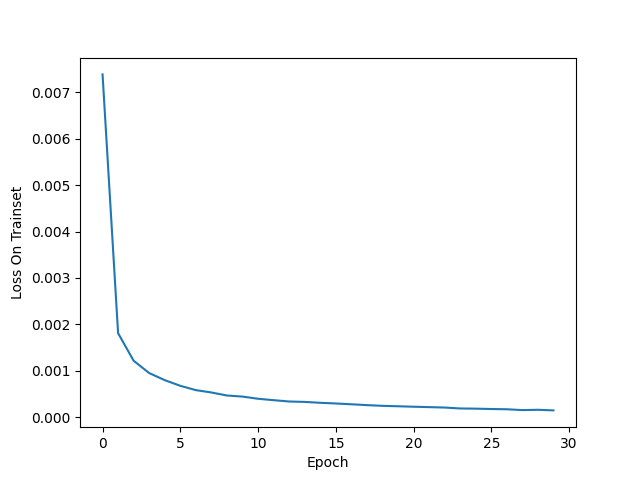


图7 训练损失

训练和测试的精确度变化如下图所示。

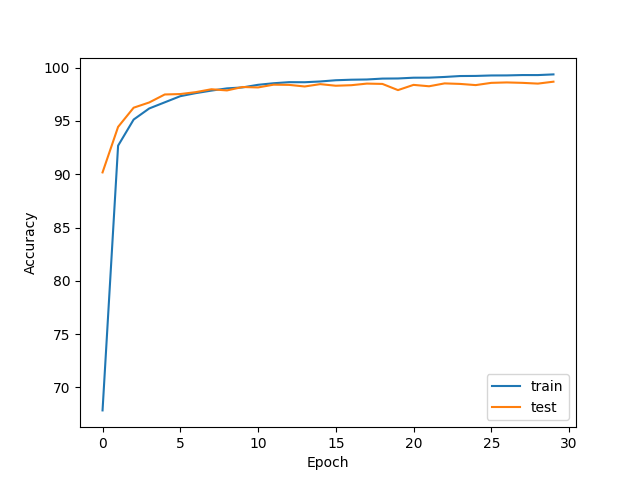


图8 训练和测试的精确度变化