

**计算机视觉工程实践实验报告**

|  |  |
| --- | --- |
| **姓 名:** | 高杨 |
| **学 号:** | 123106222810 |
| **学 院:** | 计算机科学与工程学院 |

**2024年 5月 1日**

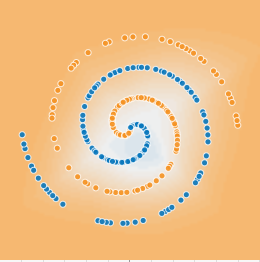
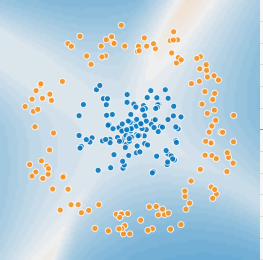
**1.Tensorflow Playground**

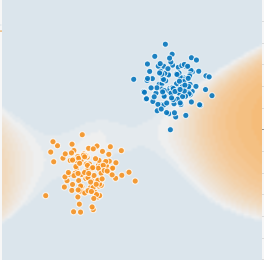
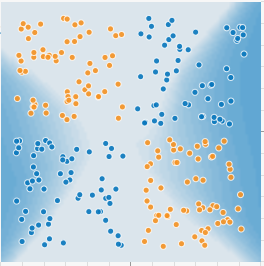
本次试用报告将介绍我对TensorFlow Playground的试用体验。TensorFlow Playground是一个基于Web的交互式可视化工具，旨在帮助用户直观地理解神经网络的工作原理。在这次试用中，我将探索其界面、功能和使用体验，并评估其对于初学者和专业人士的实用性。

**1.1 数据**

在这个Playground中，我们有4种不同形态的数据可以玩。每组数据都是不同形态分布的一群点。每一个点，都有2个特征：x1和x2，表示点的位置。而我们数据中的点有2类：橙色和蓝色。我们的神经网络的目标，就是通过训练，知道哪些位置的点是橙色、哪些位置的点是蓝色。

这个工具提供了一些经典的数据集，例如螺旋数据集和月亮数据集，方便用户快速开始实验。我们可以调整noise的大小，甚至可以改变训练数据和测试数据的比例多少。





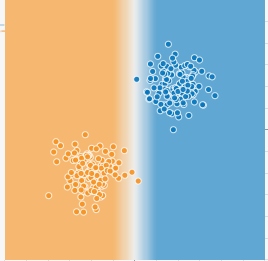
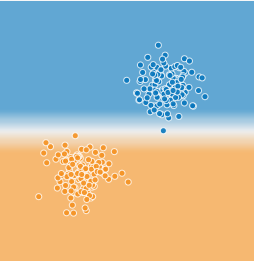
**1.2 功能**

当我们将所有数据点的信息输入机器学习系统时，进行特征提取是至关重要的。特征提取是指从原始数据中提取出对于我们所要解决的问题具有重要意义的特征。举例来说，如果我们的任务是区分蓝莓和橙子，那么大小、颜色等因素都可能成为我们用来区分的特征。在TensorFlow Playground中，每个数据点都由两个特征（x1和x2）表示。然而，除了这两个显式的特征之外，我们还可以从它们中派生出许多其他有用的特征。

例如，在一个平面上表示的数据集中，除了原始的x1和x2坐标之外，我们还可以计算每个数据点到坐标原点的距离作为一个新的特征。这个距离特征可能对于模型的分类效果起到重要作用。此外，我们还可以计算每个数据点到其他数据点的距离，或者计算每个数据点与某个预定义中心点的距离。这些都是从原始特征派生出的新特征，可以帮助模型更好地理解数据并进行准确的分类。

除了距离特征之外，我们还可以考虑将原始特征进行组合或转换。例如，我们可以将x1和x2相乘得到一个新的特征，或者取它们的平方、立方等等。这些操作可以使得模型更好地捕捉数据之间的非线性关系，提高模型的分类性能。

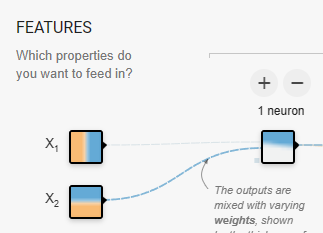
抽象来说，我们的机器学习classifier（分类器）其实是在试图画一条或多条线。如果我们能够100％正确地区分蓝色和橙色的点，蓝色的点会在线的一边，橙色的会在另一边。



很显然，我们需要巧妙地结合一种或多种特征，才能成功地将蓝色点和橙色点进行分类。这种特征的智能提取，在机器学习应用中往往是最具挑战性的任务之一。然而，幸运的是，我们可以依靠神经网络来解决这个问题。神经网络具有强大的学习能力和适应性，能够从数据中自动学习到高层次的特征表示，无需手动进行特征提取。

**1.3 用法**

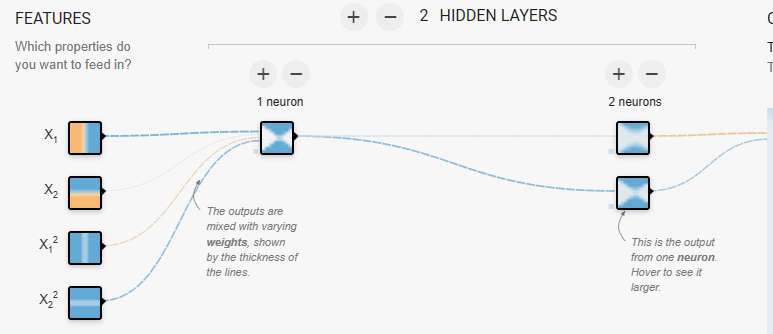
如果我们选定x1和x2作为特征，我们神经网络的每一层的每个神经元，都会将它们进行组合，来算出结果：



在神经网络中，每一层神经元将前一层的输出作为输入，并进行进一步的组合和加权。这种组合和加权过程在网络的每一层都会发生，形成了层与层之间的复杂连接。具体来说，在神经网络的前馈过程中，每个神经元将前一层的输出与其对应的权重进行加权求和，并通过激活函数产生输出。

而在训练过程中，通过反向传播（back propagation），我们不断地调整这些权重，以使得网络的输出与实际标签之间的误差最小化。这意味着，根据上一次预测的准确性，我们会针对每个连接的权重进行微调。如果某个连接在上一次预测中发挥了重要作用，我们可能会增加其权重，使得其在下一次预测中更加重要；反之，如果某个连接的作用不明显，我们可能会减少其权重，降低其影响力。

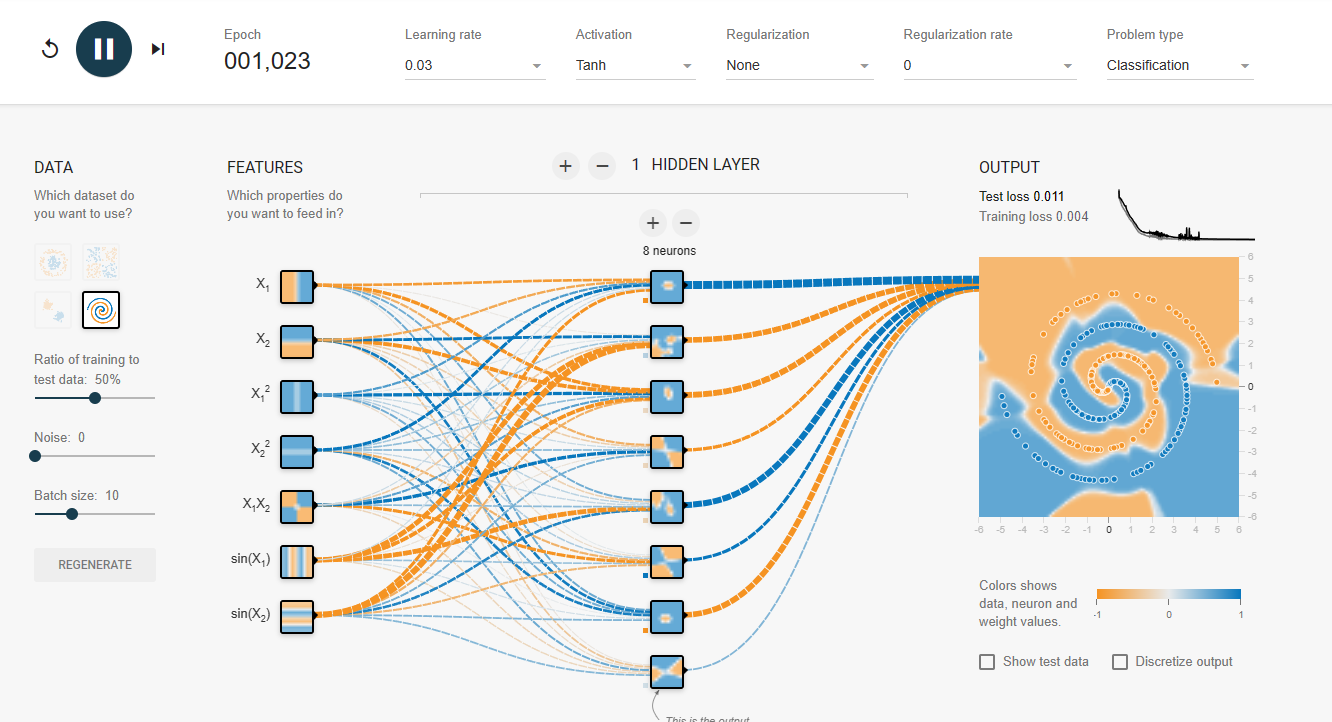
在可视化神经网络时，通常会使用线的粗细来表示连接的权重大小。线越粗，表示连接的权重越大，对于模型的预测起到了更重要的作用；而线越细，表示连接的权重较小，对于模型的预测影响较小。这种可视化方式帮助我们直观地理解神经网络的结构和权重分布，从而更好地进行调试和优化。



**1.4 例子**

在这里，我们选用螺旋数据集进行测试。我们将所有能想到的几种特征都输入到系统中，并选择只有1层的神经网络。我们可以选用不同的学习率、激活函数等。

在经过多轮训练后，我们发现最后的图基本上能够区分不同的类别了。



**2.LeNet-5**

**2.1 LeNet**

LeNet是由Yann LeCun在1998年提出的一个经典的卷积神经网络结构，主要用于手写数字识别任务。它是深度学习领域中的一个里程碑，为后来的卷积神经网络的发展奠定了基础。

LeNet的基本结构包括四个部分：卷积层、池化层、全连接层和激活函数。LeNet在当时对手写数字识别任务取得了很好的效果，并且为后来更复杂的深度学习模型的发展奠定了基础。尽管LeNet相对简单，但它展示了卷积神经网络在处理图像分类等任务上的强大能力，为后来深度学习的发展开辟了道路。

**2.2 mindspore深度学习框架**

MindSpore是一个开源的深度学习框架，由华为推出，旨在提供高效、灵活、易用的工具来支持深度学习模型的开发和部署。

MindSpore采用了自动并行和动态图技术，可以充分利用硬件资源，实现模型的高效训练和推理。它支持在多种硬件平台上运行，包括CPU、GPU和Ascend AI处理器。其提供了丰富的运算算子和模型组件，用户可以根据自己的需求自由组合和定制模型。它支持动态图和静态图两种模式，用户可以根据任务的复杂度和需求选择合适的模式进行开发。  
 MindSpore致力于构建一个开放、共享的生态系统，与各种硬件厂商、学术机构和开发者社区合作，共同推动深度学习技术的发展和应用。在这个作业中，我们使用mindspore框架利用LeNet对MNIST数据集进行训练。

**2.3 MNIST 数据集**

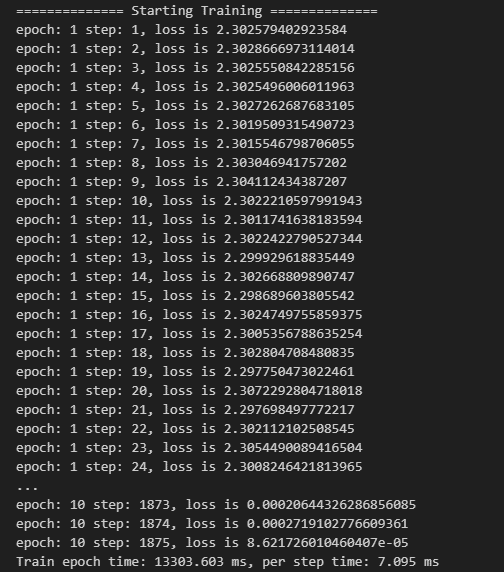
MNIST数据集是一个经典的手写数字识别数据集，常用于测试和验证机器学习和深度学习模型的性能。该数据集包含了来自 0 到 9 的手写数字图片，共计 70,000 张图片，其中 60,000 张用作训练集，10,000 张用作测试集。

每张图片都是28x28像素的灰度图像，表示了一个手写数字。图片经过了预处理，灰度值范围在 0 到 255 之间。除了图片数据外，每个样本还有对应的标签，表示了图片所代表的数字（0 到 9）。

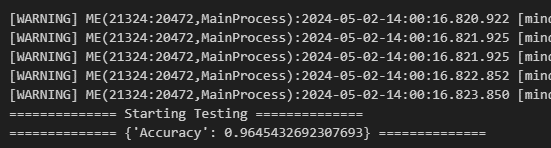
MNIST数据集被广泛用于训练和测试各种机器学习和深度学习模型，尤其是用于图像分类任务。由于数据集的规模较小，模型可以快速训练和测试，因此它常被用于验证算法的正确性和效果。同时，由于数据集的简单性，它也被用于教学和学术研究，帮助初学者理解和实践机器学习算法。

**2.4实验结果**

数据训练结果：

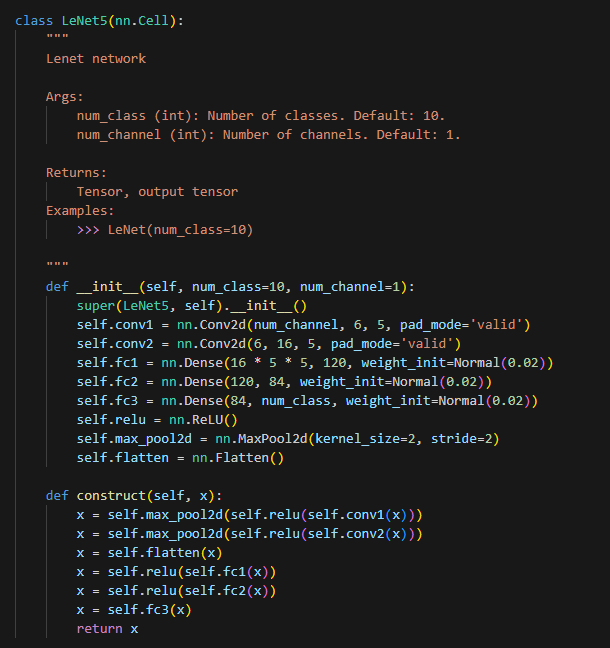


数据测试结果：

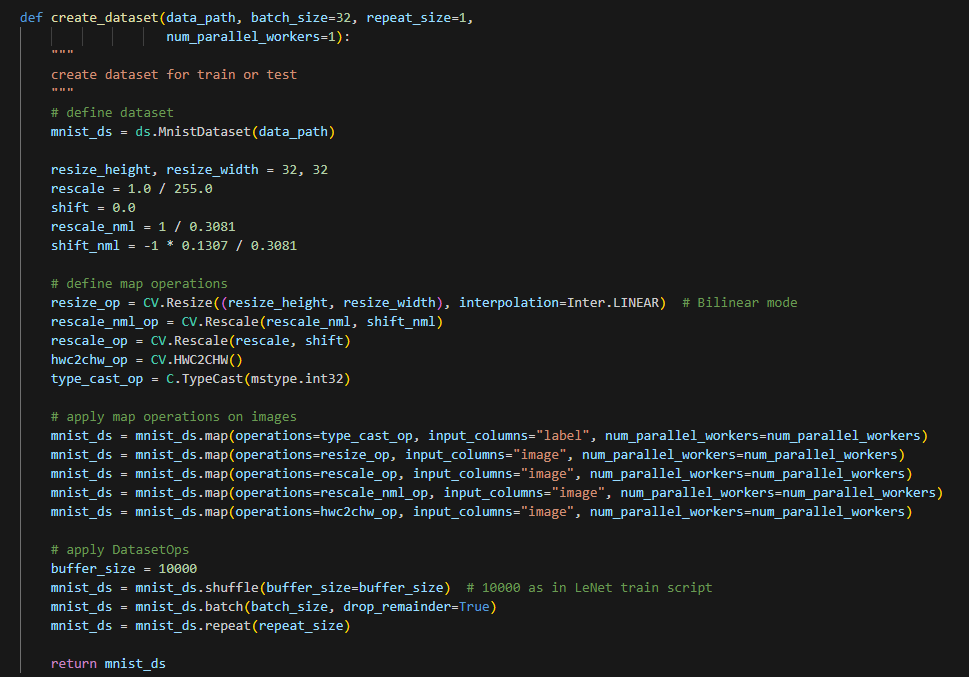


**2.4相关代码**

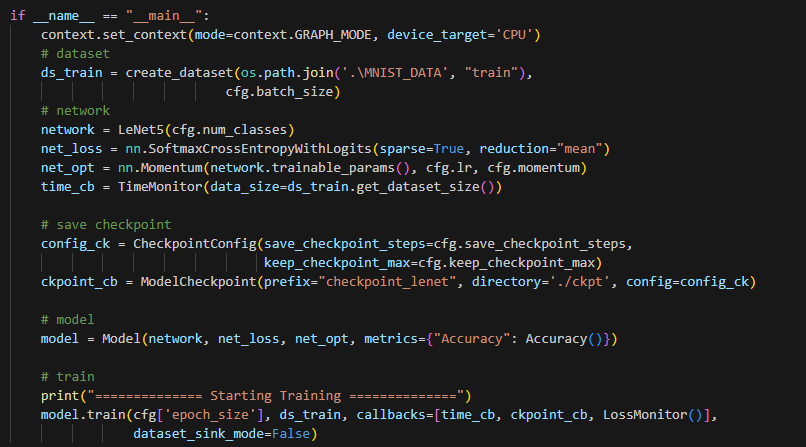
LeNet：



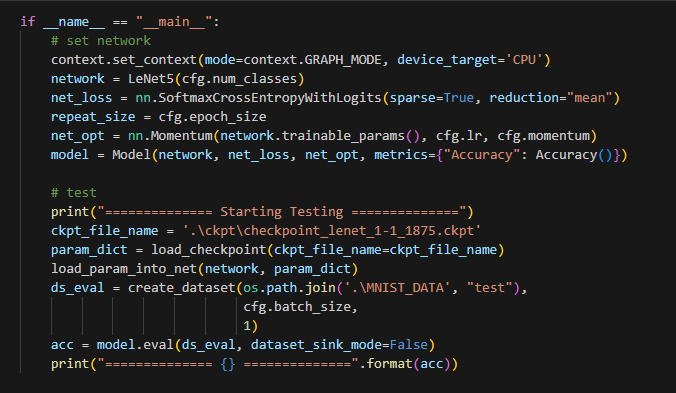
Dataset:



Train 函数：



Test 函数：



**3.总结**

在对 TensorFlow Playground 进行试用后，我对这个基于 Web 的交互式深度学习工具有了深入的了解。它提供了一个直观的界面和丰富的功能，使用户能够轻松地探索神经网络的工作原理和训练过程。TensorFlow Playground 是一个强大而直观的工具，可以帮助用户快速理解和探索神经网络的基本概念。它的简洁界面、丰富功能和易用性使其成为学习和教学深度学习的理想选择，同时也为专业人士提供了一个快速验证和实验的平台。

另外，通过本次试用MindSpore框架和LeNet模型进行手写数字识别的实验，我们对这两者的性能和效果有了全面的了解。首先，MindSpore框架展现了出色的易用性和高效性。它提供了丰富的功能和简洁的API，使得模型的构建、训练和评估过程变得轻松而高效。同时，MindSpore框架对硬件资源的利用率也很高，可以在不同的硬件平台上实现高效的模型训练和推理。

其次，LeNet模型在手写数字识别任务上表现出了良好的性能和可靠性。作为一个经典的卷积神经网络结构，LeNet通过卷积层和池化层等操作，有效地提取了图像的特征，并通过全连接层实现了对手写数字的分类。

通过实验，我们验证了MindSpore框架和LeNet模型的有效性，并得到了令人满意的结果。我们相信，在实际应用中，这对组合将会发挥出更大的作用，为图像分类和识别等任务提供强大的支持和解决方案