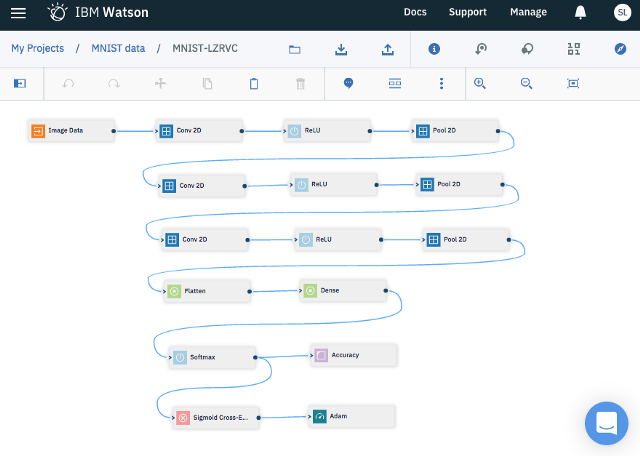
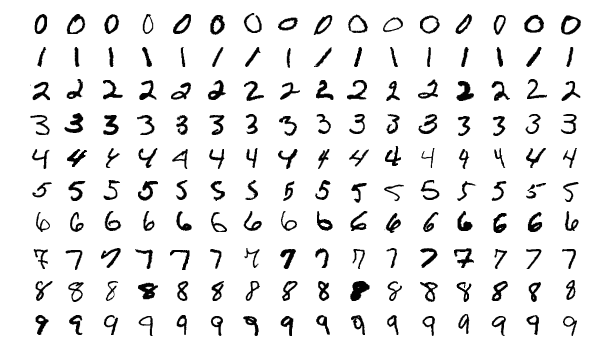
# 利用Watson工作室探索深度学习和神经网络建模器



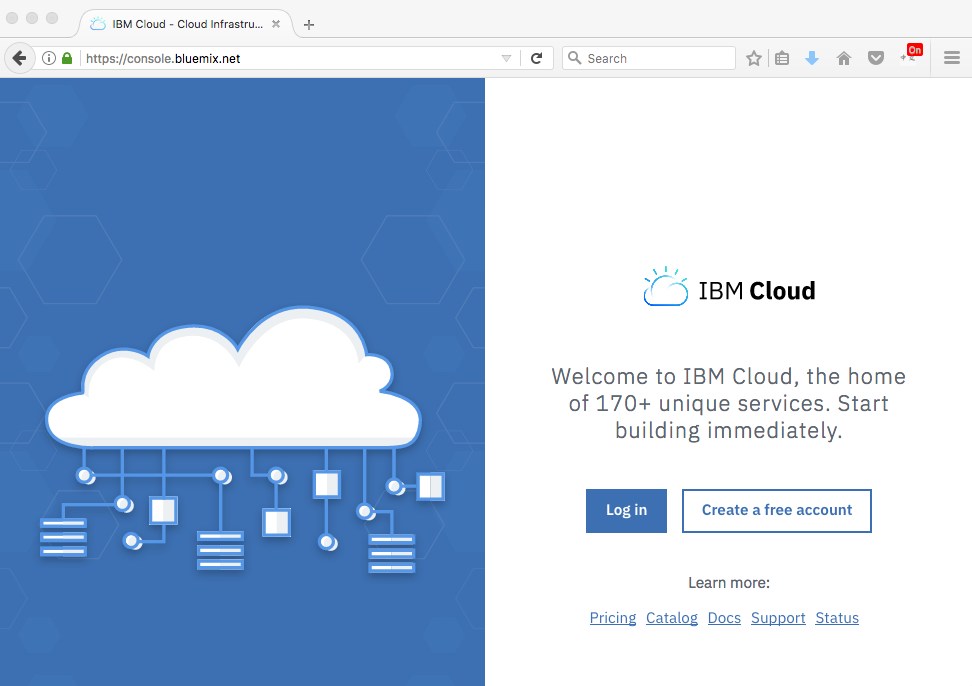
创建自己的卷积神经网络太有趣了！为构建强大的深度学习模型提供了直观的图形工具。它很好地集成了的所有方面。应用程序开发人员、分析师和数据科学家使用WML将机器学习无缝地集成到他们的工作流中。例如，只需单击几下鼠标，就可以将模型保存为REST API并将其部署为restapi，从而最大限度地缩短机器学习模型的工作时间。在这个例子中，我使用了熟悉的（手写数字的图像）并创建了一个三层深的CNN。然后，在使用NVIDAI GPUs训练模型之前，我在TensorFlow/Keras python代码中导出了算法。我还使用下载的代码在Jupyter Macbook环境中本地训练模型。在这两种情况下，我都得到了很好的结果。当然，在云中使用gpu进行训练是闪电般的快。

我们正在分类的数字图像如下：

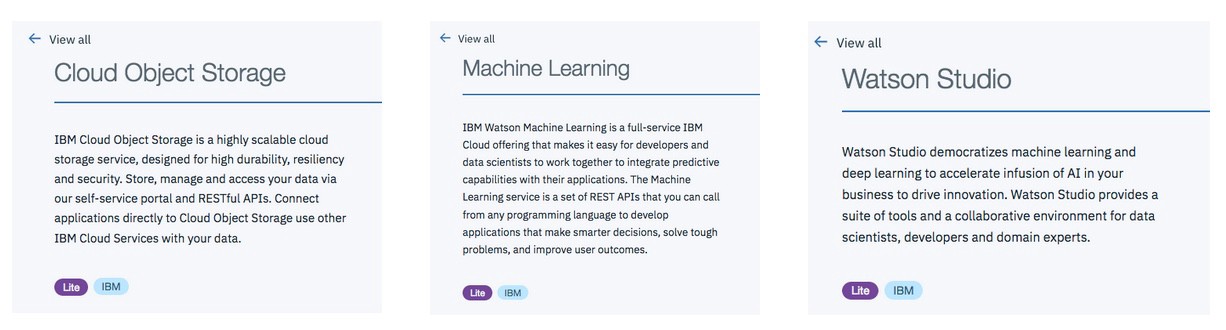


给数据科学家和人工智能开发者：去使用它。对于所有其他的爱好者来说，学习人工智能最好的方法就是自己创造它！

首先，要访问Watson Studio环境，您需要您的IBM ID。请访问，然后创建您的免费帐户。我在这里使用的一切都是免费的，因此您可以学习、开发您的模型并将其部署为高效使用，既快速又轻松。

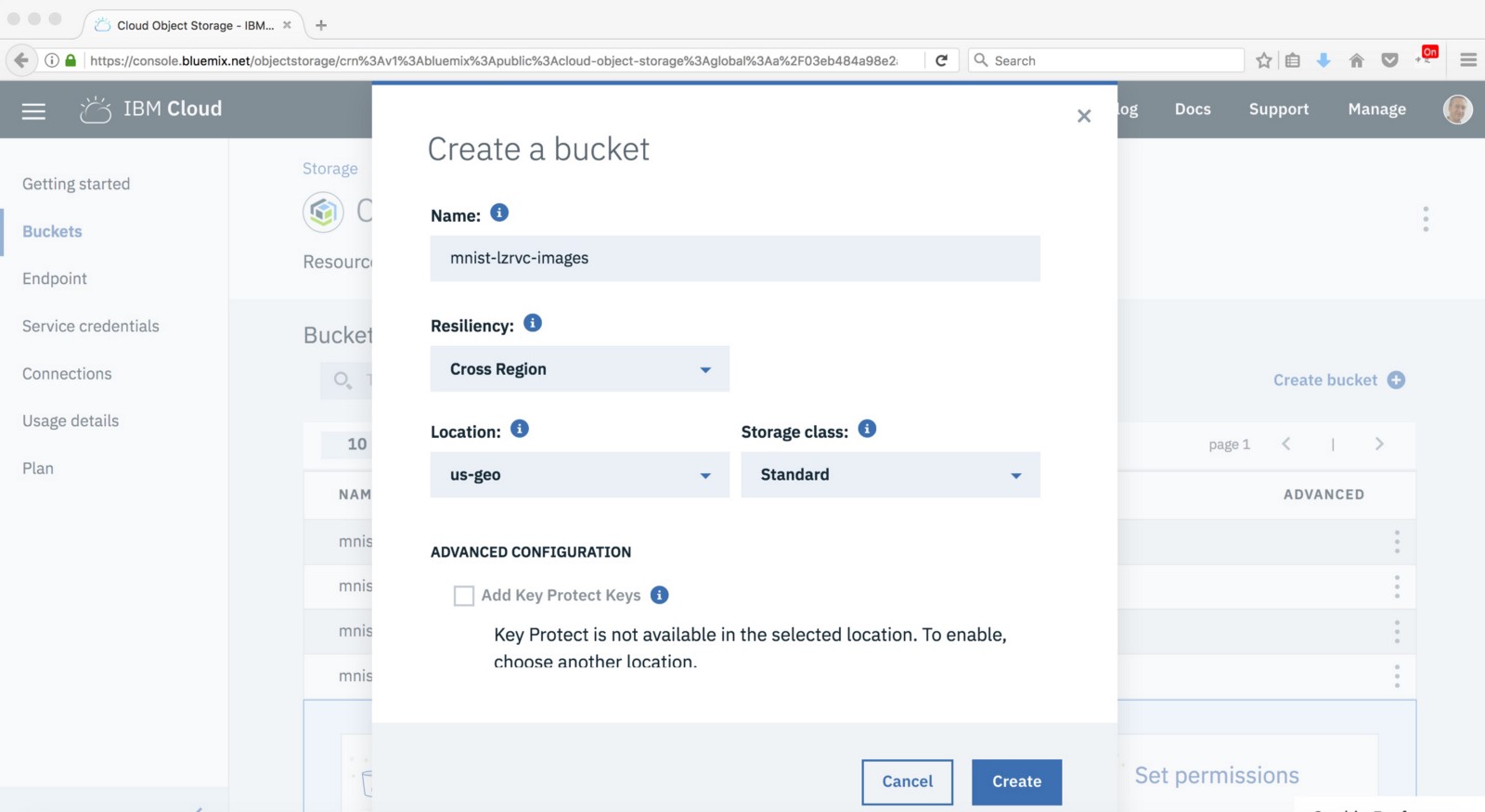


一旦您能够访问IBM Cloud，您将需要获得三项服务：\*\*\*（存储您的数据集、经过培训的模型和培训结果），（培训您的模型并从ultrafast gpu中获益）和（创建和管理您的DL模型，与您的团队协作，等等）。

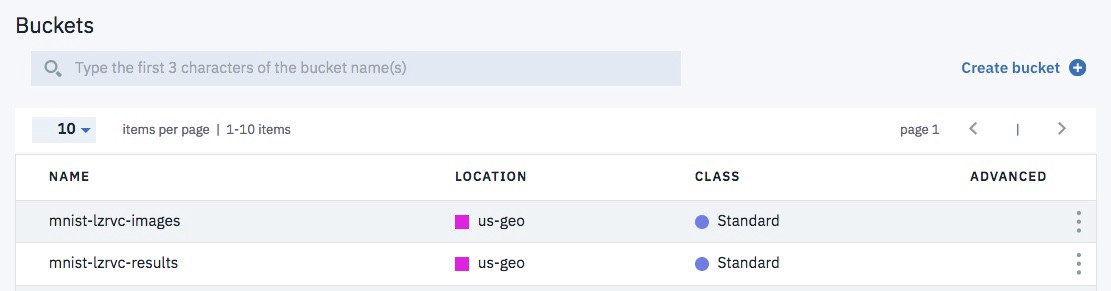


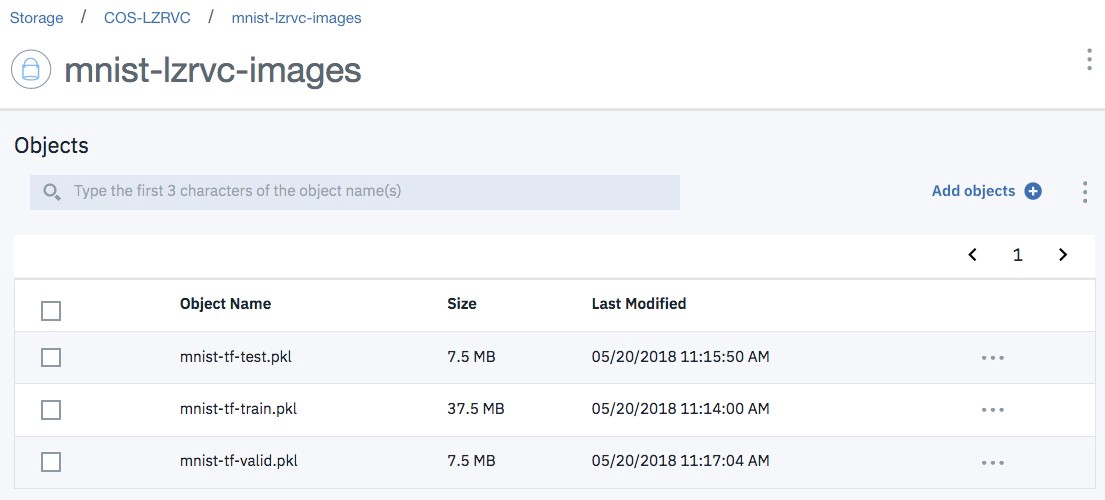
完成此操作后，请在Watson仪表板中检查是否已正确创建了所有这三个服务。你可以从\*\*\*\*或从。

下一步是在云对象存储（COS）中创建bucket并上传数据。我的建议是创建两个bucket（您也可以将它们称为数据对象的容器），一个用于数据，一个用于模型和结果。对于较大的模型，需要更大的粒度，因为Watson机器学习（WML）将从训练数据桶中加载数据，因此您希望避免不必要的等待时间和对训练运行中未使用的数据的内存消耗。

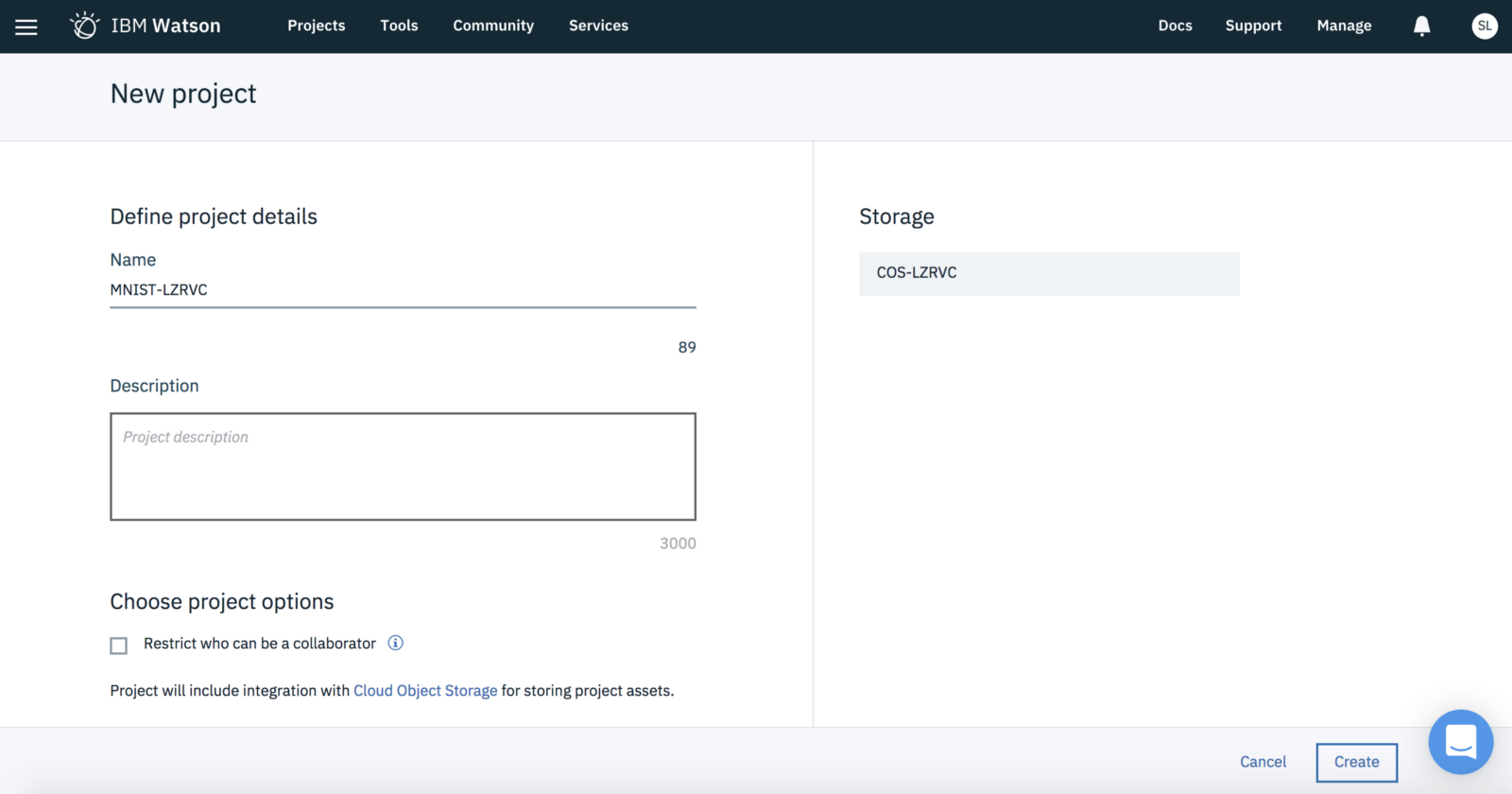


以下是创建两个bucket并上载数据时的情况：

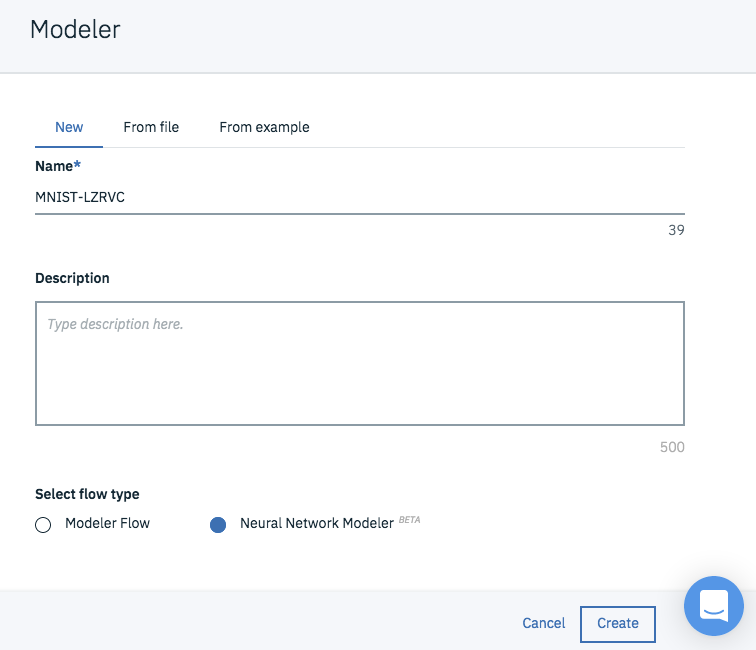




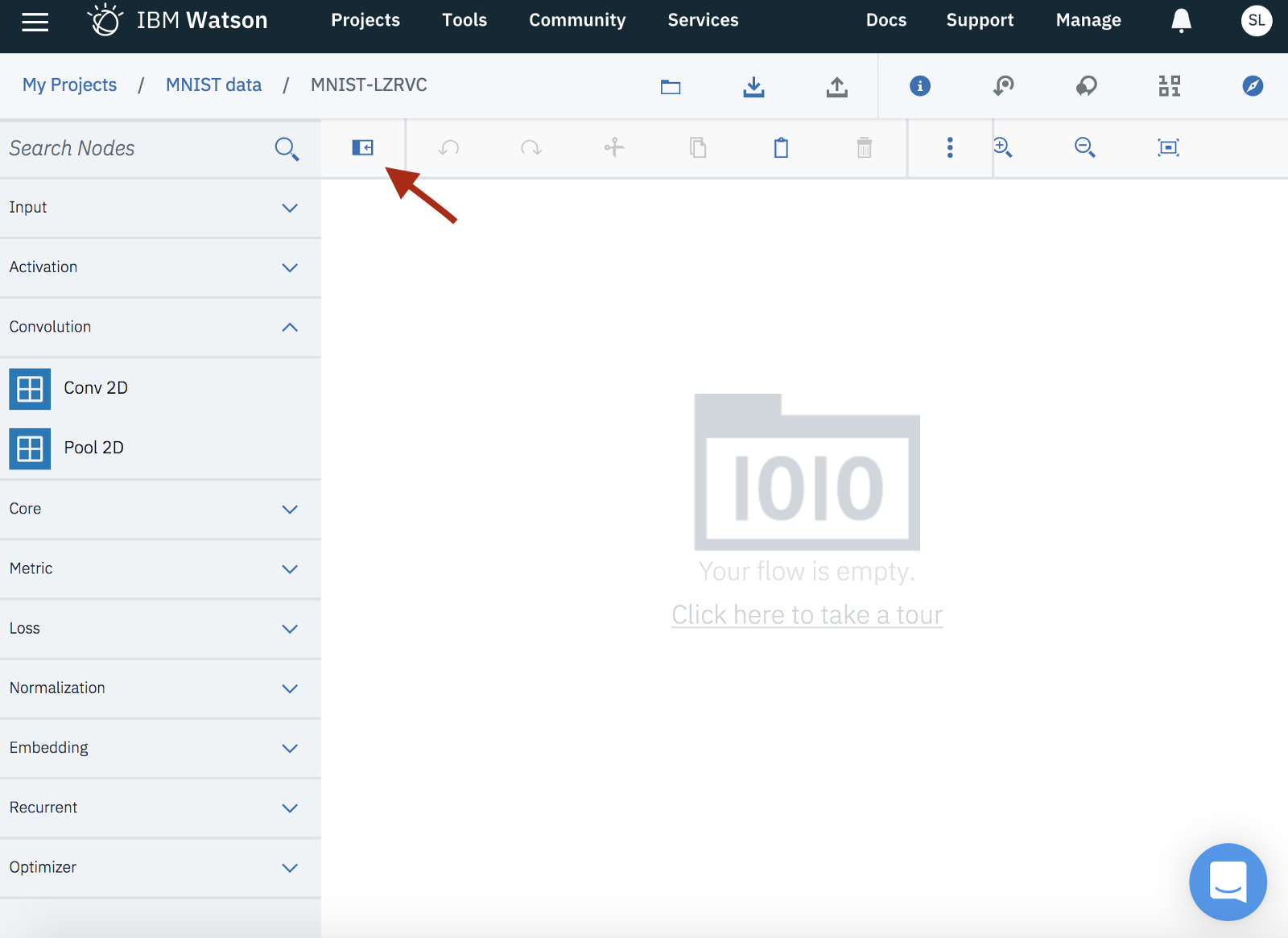
完成后，可以启动Watson Studio并创建项目。我把我的名字命名为MNIST-LZRVC。您将在右侧看到云对象存储（COS）。Watson Studio将自动创建到您的COS服务的连接。



接下来，在Watson Studio的assets选项卡中，转到Modeler流并添加一个新流。将其命名为您的项目，但要使用一些唯一的ID，以便您在开始修改和尝试不同的体系结构时能够识别。

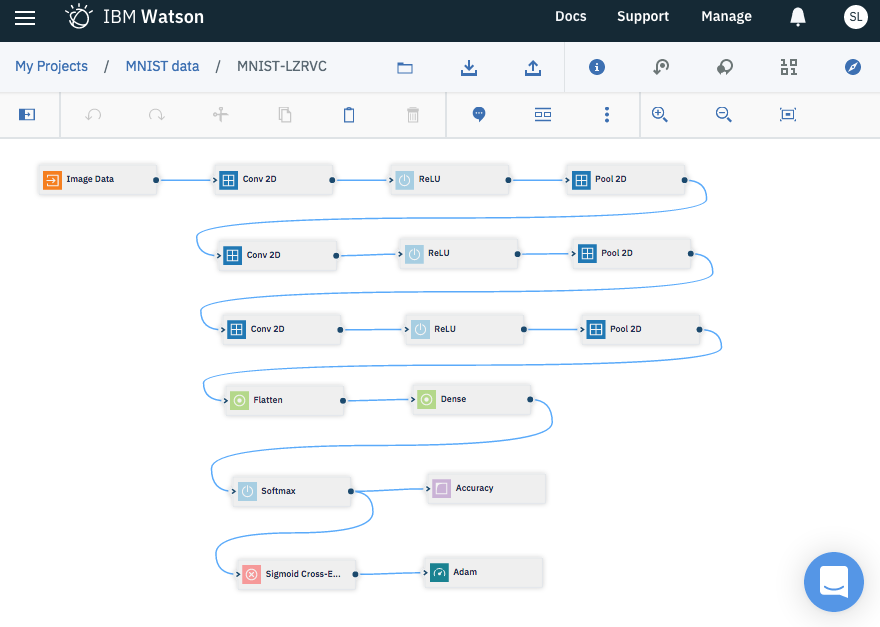


当你这样做的时候，你会在设计画布上为你的神经网络建模您可以单击左上角的按钮打开包含不同NN构建基块组的侧工具栏。

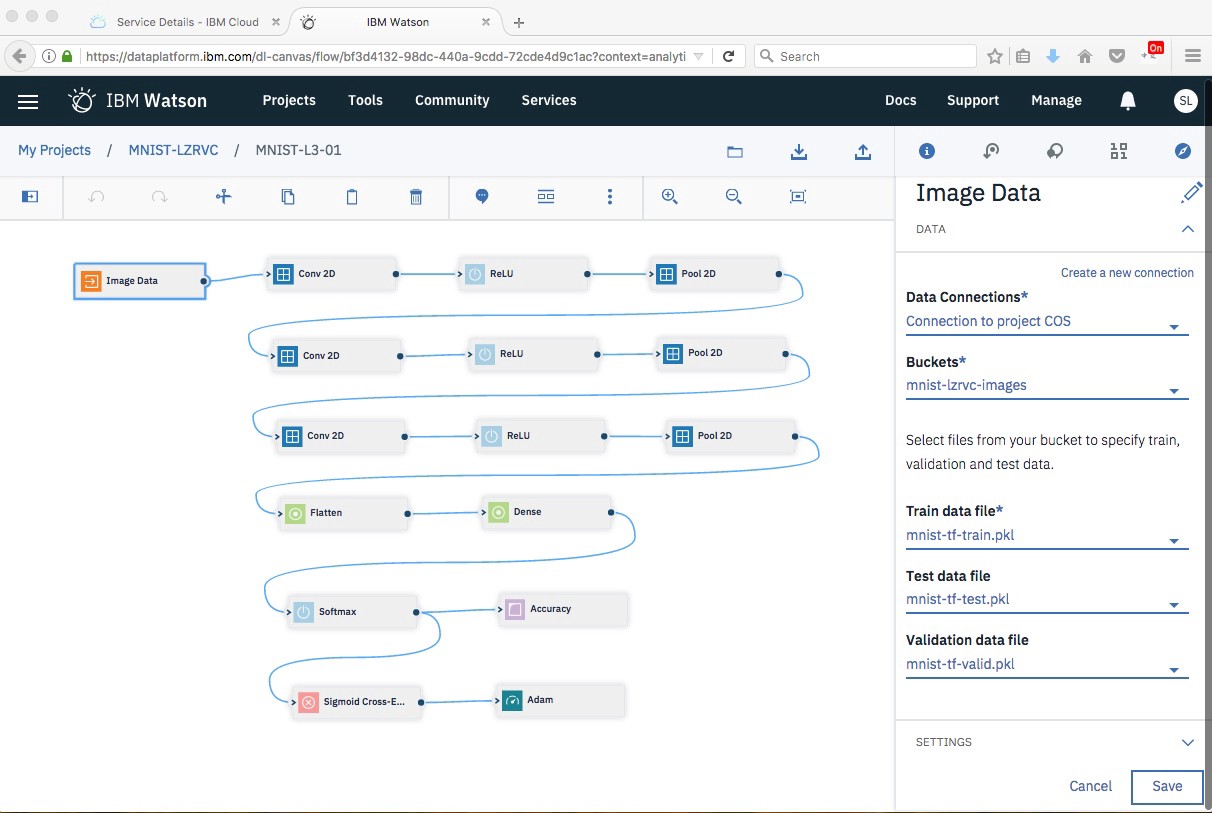


这些构建块允许您快速、轻松地从头开始构建整个深层神经结构。平台上还提供了许多样本。当然，你需要知道你在做什么，但如果你想了解更多，网上有很多非常好的学习资源。在这个图像识别示例的特定情况下，我将输入数据连接到三个卷积+ReLU激活+池层，然后连接到一个完全连接的层，并将得到的特征表示传递到具有10个类别（针对10个不同数字）的softmax分类器。最后，利用交叉熵损失函数将一切联系起来，使用Adam自适应学习速率优化器。我正在使用精度度量来测量我的模型在验证数据集上的性能。

我试过不同的层数。一个显然是不够的，但三个层次给出了非常好的结果，即使只经过两三个训练阶段。当然，你需要小心超参数，比如过滤器的数量和大小，跨步，通道等等。

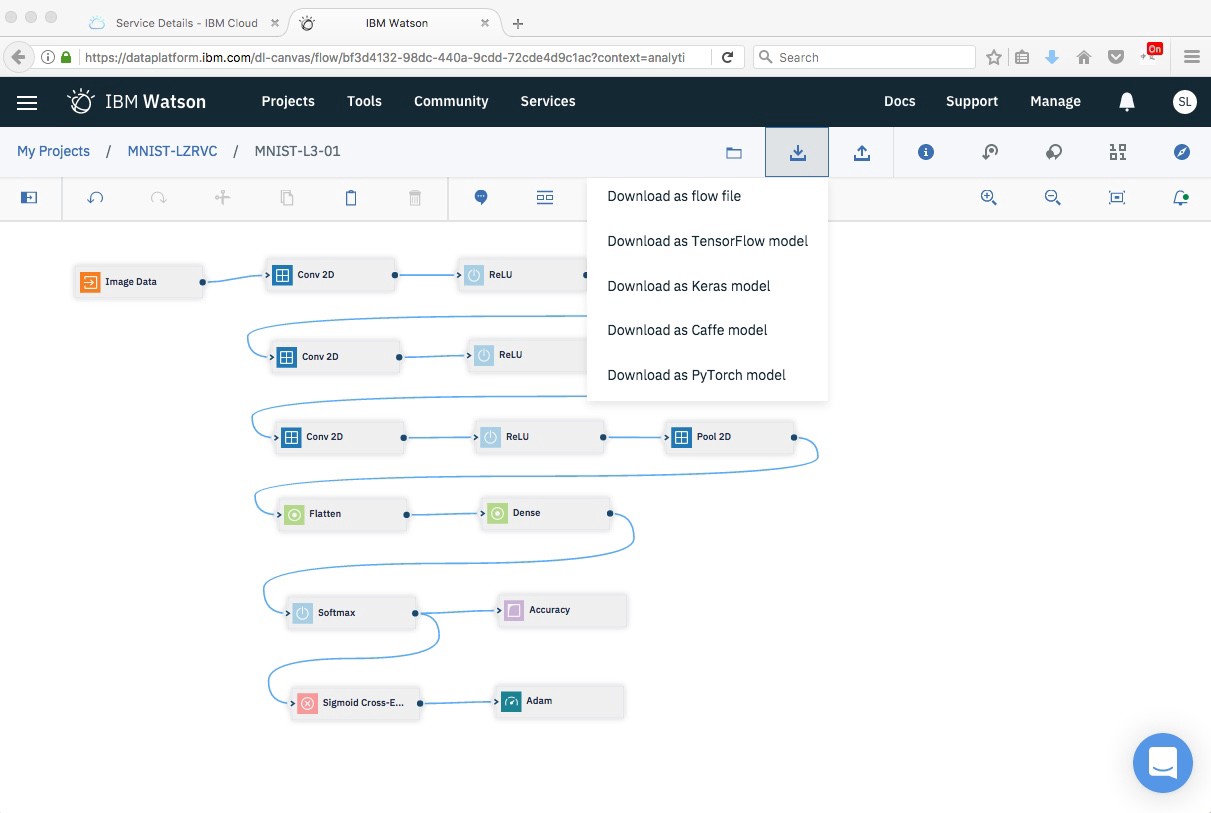


当您完成了项目中最具创造性的部分时，将您的模型与包含您的数据集的COS bucket连接起来。在这个新的数据桶中点击上传文件的三个点。我使用三个独立文件中的pickled数据进行培训、验证和测试。



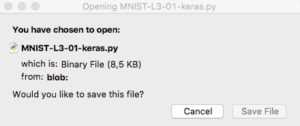
当您的模型准备好后，请查看工具栏中的两个箭头状图标。它们允许您下载代码或启动模型。

我会先把这个模型下载到我的笔记本上，因为我想玩它。

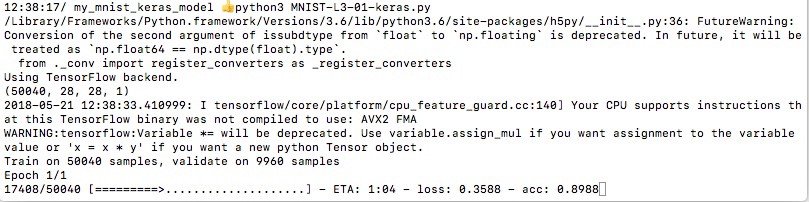


当你选择这个下载按钮，你将得到不同格式的选择。注意 - 使用Watson Studio，您可以为多个深度学习框架创建培训代码。这就是美丽的 - 使用IBM的Fabric for Deep Leaarning（FfDL），您可以在TensorFlow（带或不带Keras）、PyTorh和Caffe上运行您的深度学习算法，所有这些都在一个统一的培训和部署云环境中进行。

所以，我继续下载我的模型。我想看看它是怎么写的。



在这里，我在我的笔记本电脑python中本地运行它。不需要更改代码！



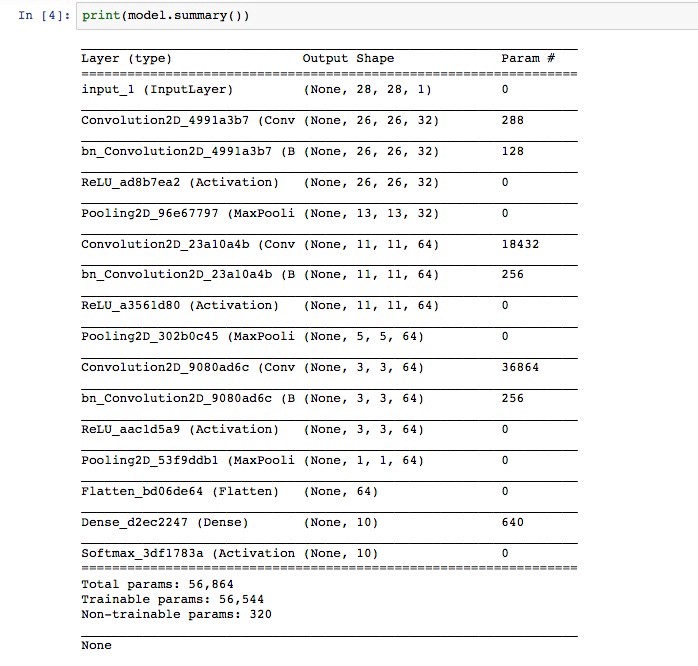
我也非常喜欢Jupyter笔记本环境，所以我只是复制粘贴到它的代码。如果您想查看，请从我的github存储库下载所有文件。

<https://github.com/LZRVC/mnist-with-keras-and-watson-studio>

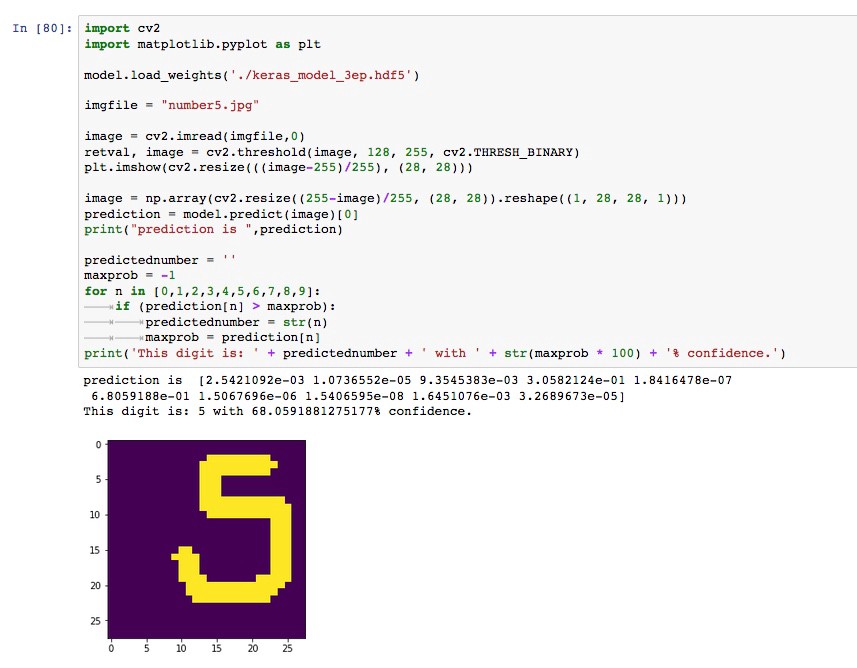
我在这里添加了两个屏幕截图来展示这个Jupyter笔记本的训练和预测。一个训练周期需要2分钟，我得到96%的准确率，包括测试数据集！



在这里可以看到模型的结构：



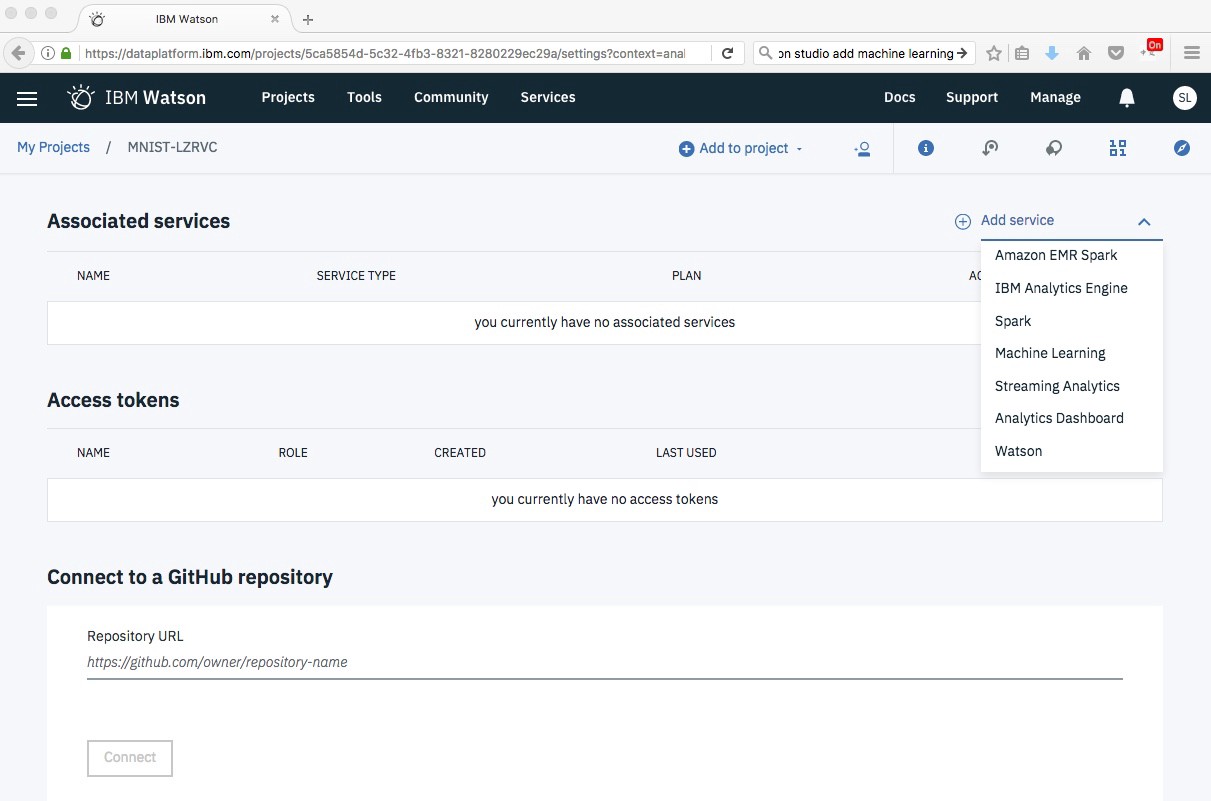
这是预测部分。对于以前看不到的图像，我将以灰度打开它们，添加对比度，规格化，调整大小和重塑它们。然后，我在这个图像上使用model.predict（）方法并从分类器中提取预测。这是一个由10个概率组成的数组，在这个数组中，我们最终选择图像传递我们的预测的最高概率。

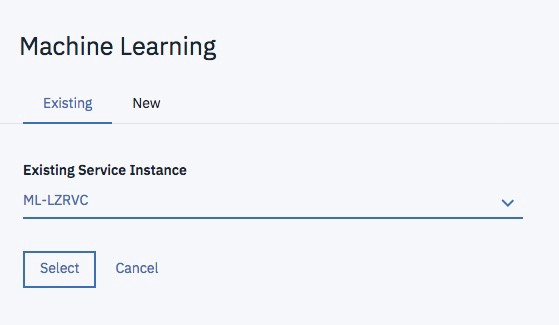


注：我注意到对数字7的预测不准确，当我在数字上划一个破折号时，对所有数字的预测都是用细铅笔写的。训练数据似乎是类似的类型，所以现实世界的结果有点偏斜。

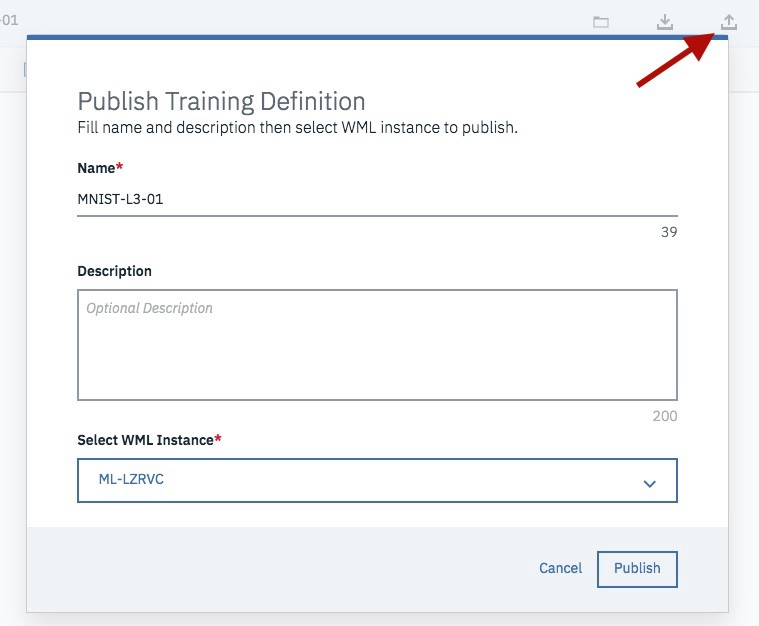
现在，回到沃森工作室。我想把我的模型直接从设计画布上传到沃森机器学习引擎。

首先，我们需要将WML与我们的项目关联起来。选择您的项目，并设置菜单选项卡选项。向下滚动查看相关服务，并添加机器学习

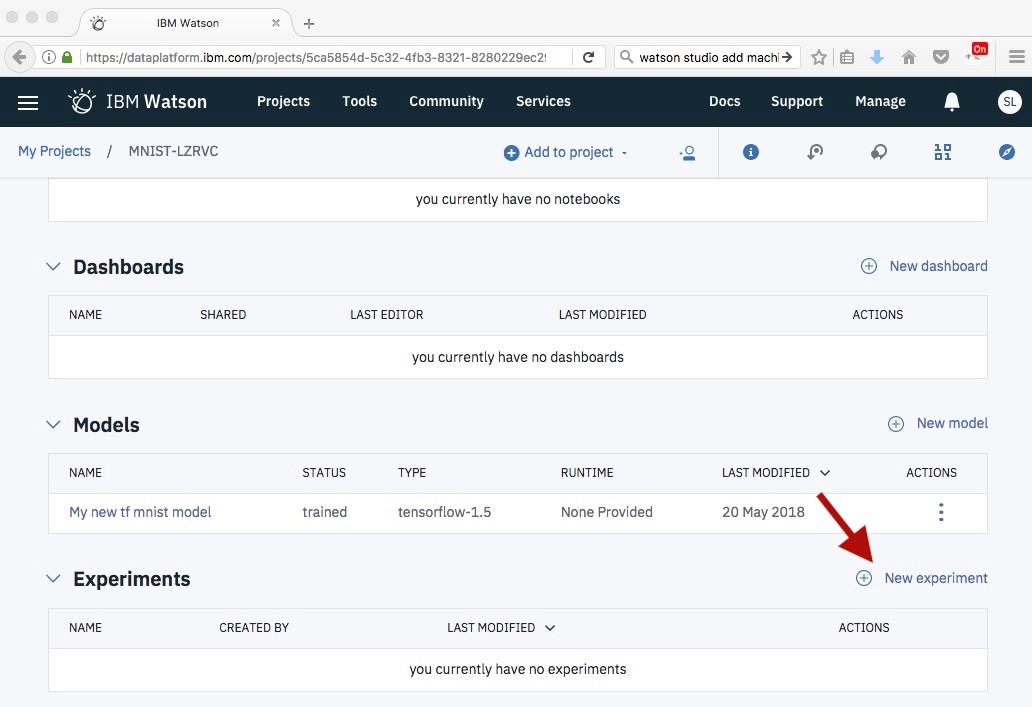




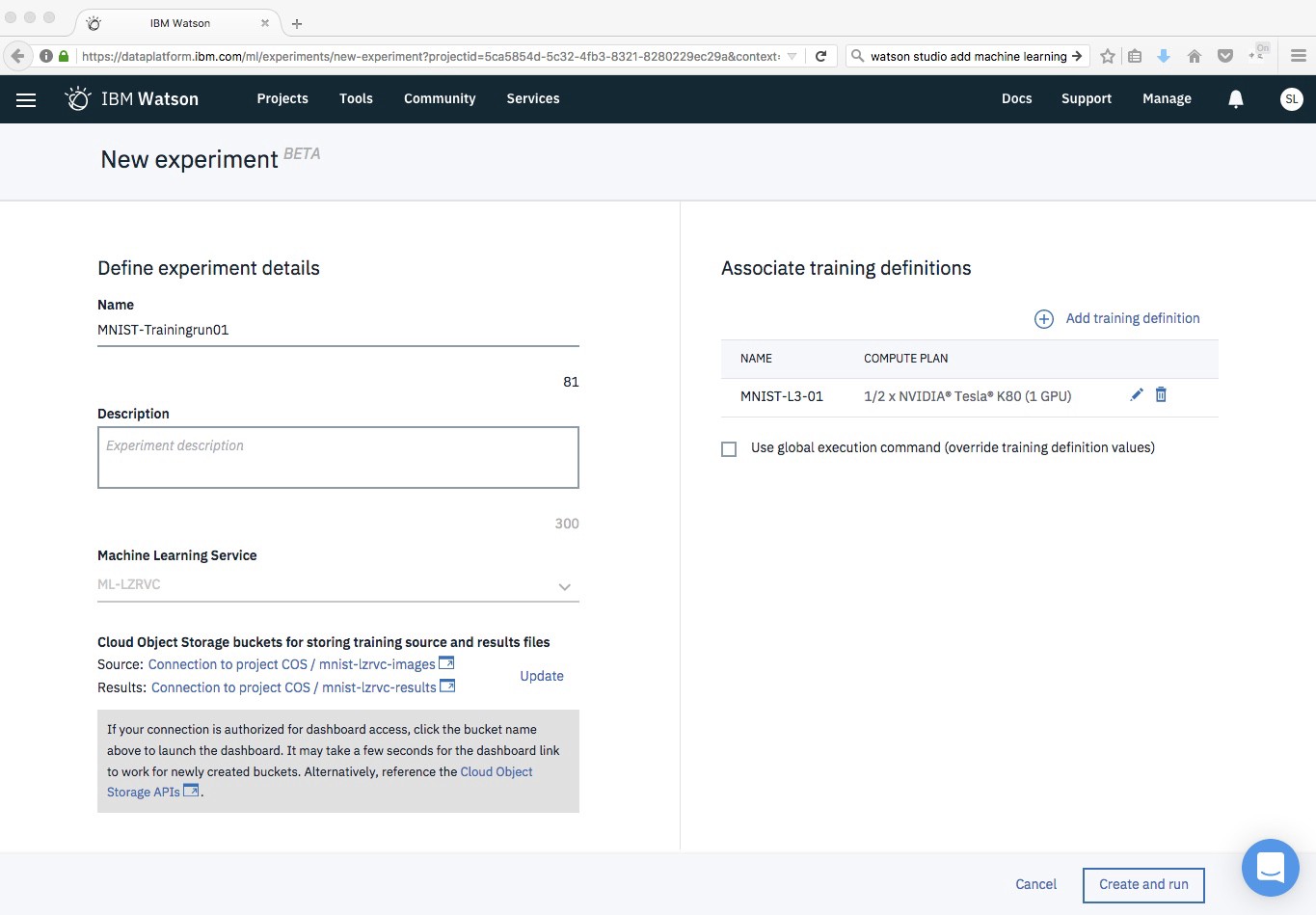
现在，转到Watson Studio的神经网络建模器部分，单击工具栏上的另一个图标将模型发布到WML。



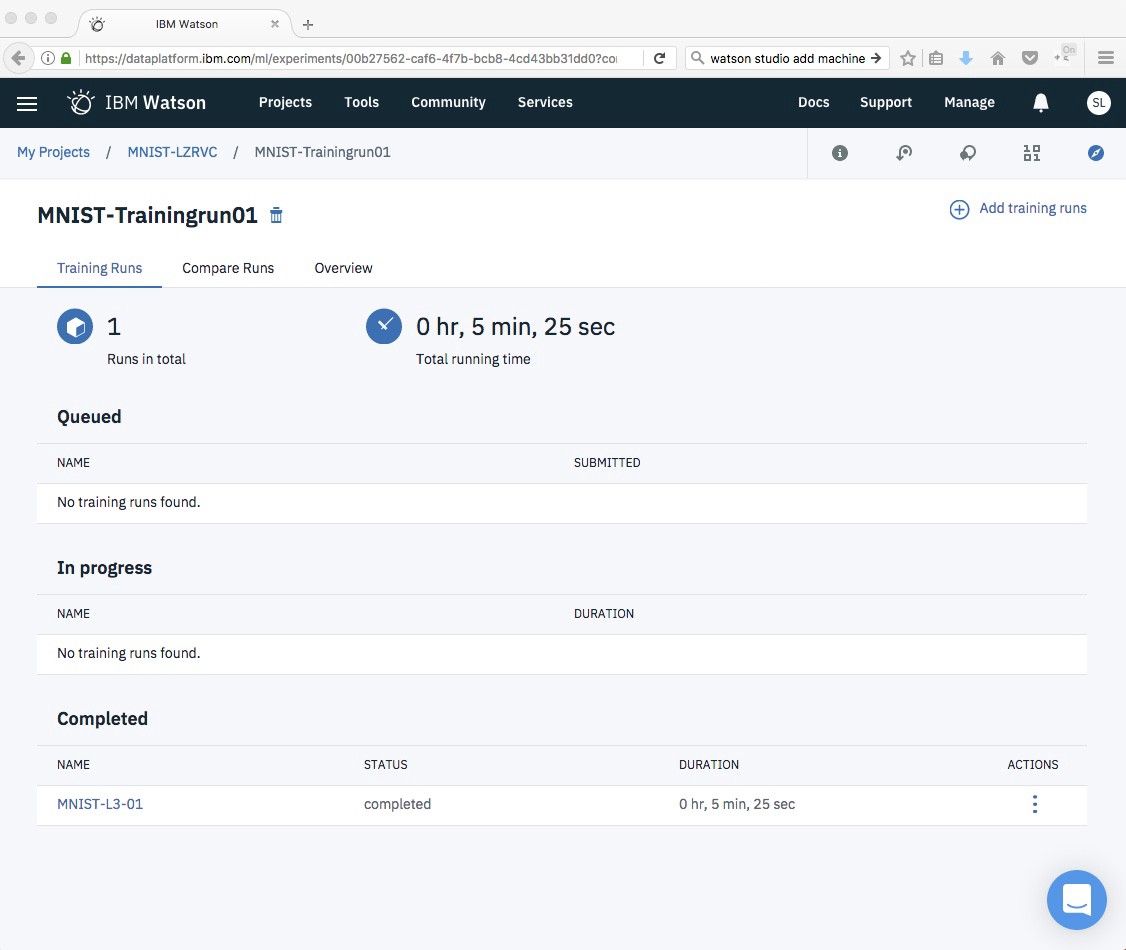
我们现在将创建一个实验（换句话说，准备启动训练运行）：



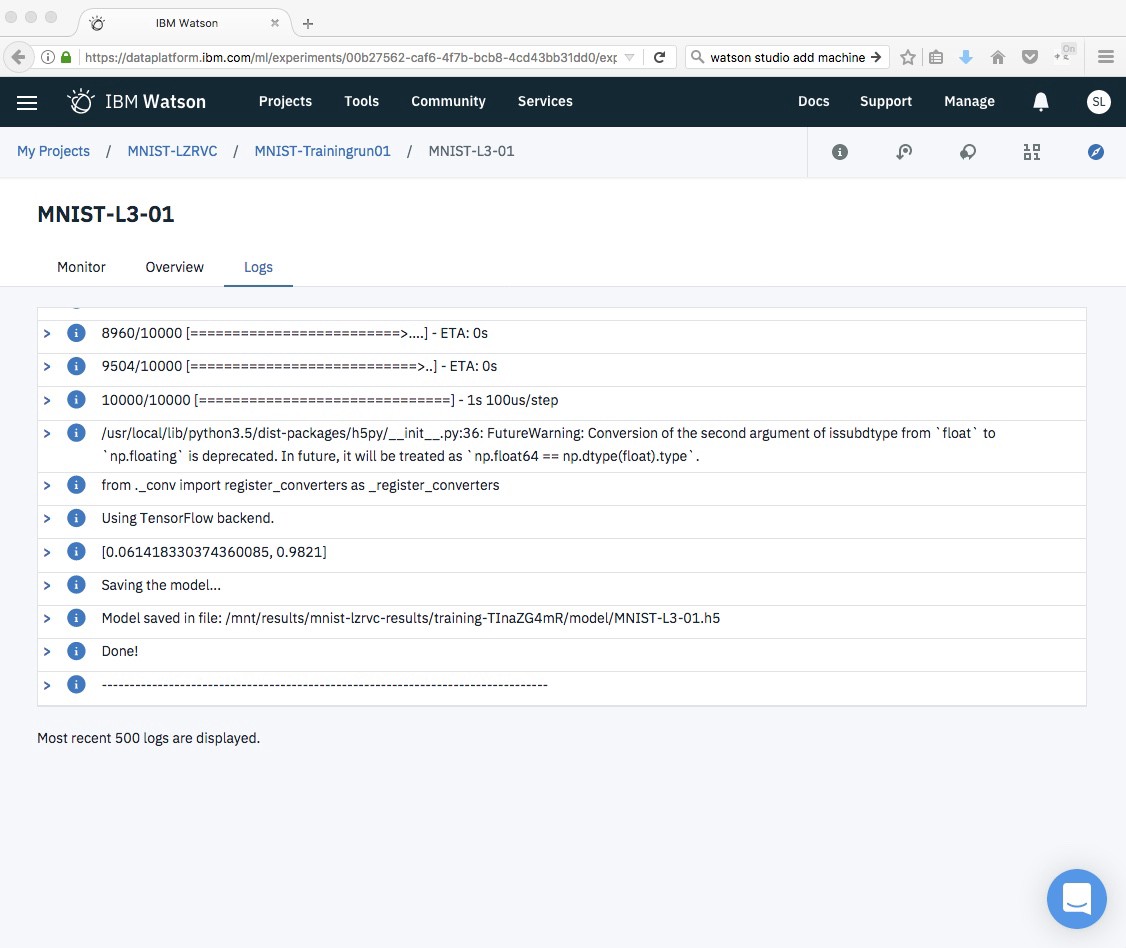
输入训练跑步的名称（注意使用id和一些有意义的缩写，以便可以追溯结果的来源）。选择并确认输入数据和结果的COS bucket，创建训练定义以选择gpu的数量，如果您想进行超参数优化（选择“None” - ；我将在以后的一篇文章中对此进行描述）。



当您单击创建并运行时，WML将自动启动您的培训过程。您可以监视进度，在系统工作时查看日志文件。有一些延迟，直到你的工作降落在GPU，这里5分25秒后，我得到的结果。考虑到我在模型定义中留下了10个指定的时间段，这段时间实际上还不错。一个训练阶段花了18秒，而且在开始之前我的工作也有一些排队。



Here in the results page you can see the location of your trained model. In the next blog post I will introduce more sophisticated concepts showing WML can be used for your Deep Learning projects, including Hyperparameter Optimization.



### Conclusion

Watson Studio, Watson Machine Learning, and Cloud Object Storage are, as you could see from this tutorial, very sophisticated, well integrated, extremely flexible and easy-to-use tools for data scientists and AI developers. DLaaS provides excellent features that make IBM Watson Studio second to none. [Get started for free today!](https://www.ibm.com/cloud/watson-studio)

Sasha Lazarevic, IBM May 21, 2018

<https://www.linkedin.com/in/lzrvc/> <https://github.com/LZRVC/mnist-with-keras-and-watson-studio>