

深度学习组件模板使用-更新记录

日期	版本	更新人	备注
2019/6/13	V0.1	张程伟	深度学习组件模板使用最初版本，2个快速实例。

深度学习组件模板使用

深度学习底层平台

本模块底层使用 tensorflow.keras，如使用自定义代码如需 import Keras，请用 tensorflow.keras。比如

```
from tensorflow.keras import backend as K
from tensorflow.keras import layers
```

现阶段提供 CPU 和英伟达 GPU 独立显卡加速版本镜像，tensorflow 版本号
为 1.13.1。

如需更改镜像请通过如下方式，加上或减去 “_gpu” 后缀。



修改镜像

目前提供 4 种镜像供选择。

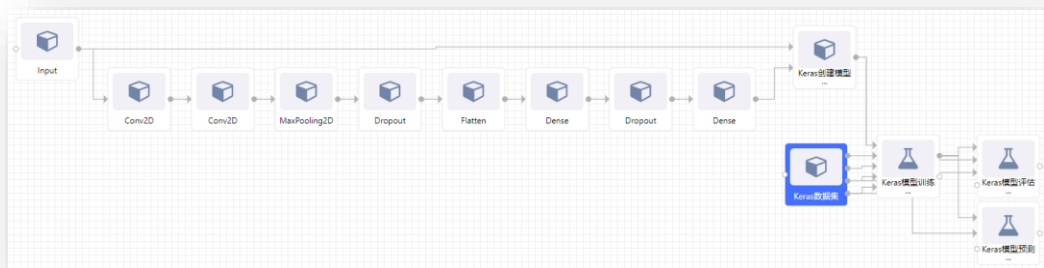
- dl: Tensorflow CPU
- dl_gpu: TensorFlow GPU
- dl_modles: TensorFlow CPU 附带打包部分 Keras ImageNet 预训练的权值文件。
- dl_gpu_models : TensorFlow GPU 附带打包部分 Keras ImageNet 预训练的权值文件。

快速实例

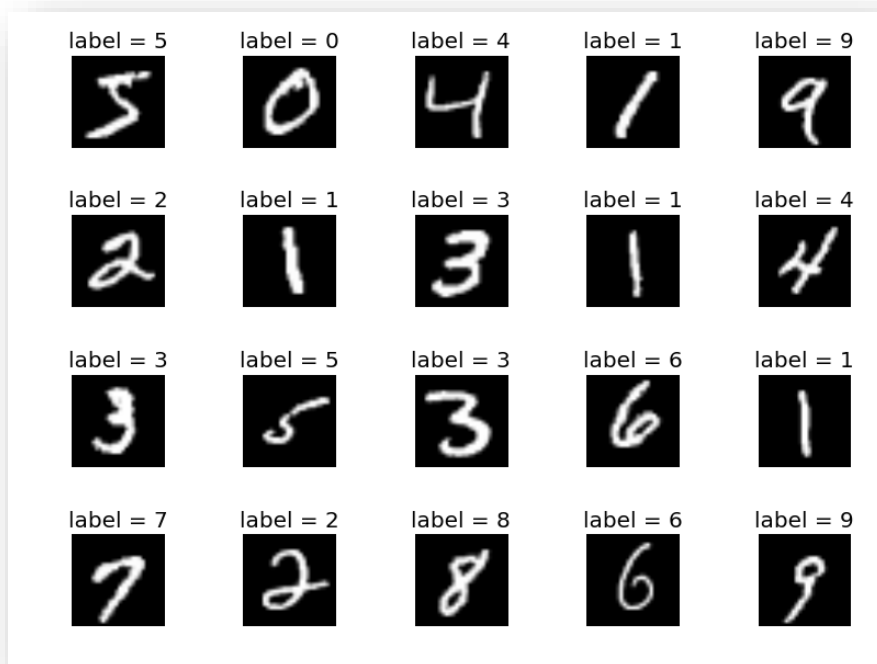
Keras CNN 手写数字识别

模板 ID: 3756

MNIST 手写数字图片分类模型, 训练一个 epoch 达到 97% 的验证数据集准确率。



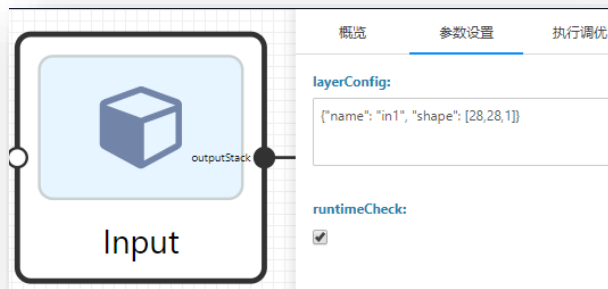
Keras CNN 手写数字识别算盘模板



MNIST 手写数字数据例子

1 添加输入层

配置模型输入层的名称和 shape，shape 需要和训练的数据对应。名称可以不填，程序中将会自动通过 UUID 生成。



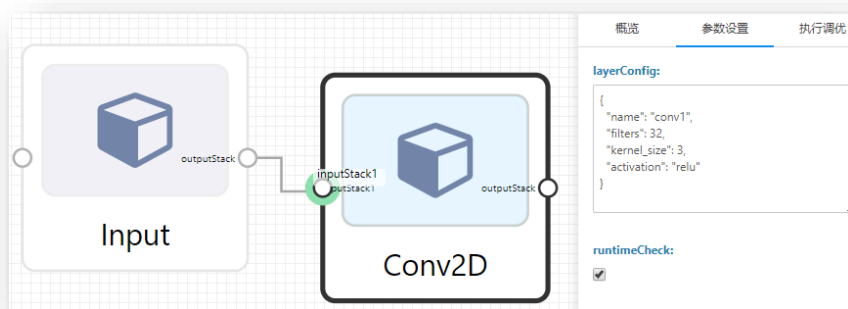
添加输入层

比如此案例中 MNIST 数据一张灰度图片为 28x28 分辨率的。

runtimeCheck 如果打钩会在运行当前节点时候尝试连接当前和已知上游网络层，如果有问题将会及时报错。

2 创建并连接一个网络层

下图创建一个 2D 卷积层，用于在网络中提取图像特征。相关的配置在 layerConfig 中定义，变量命名与 Keras API 同步，关于每个参数更多信息请参考[官方文档](#)。



连接网络层

请按照算盘实例模板拖出、配置、连接其他网络层。

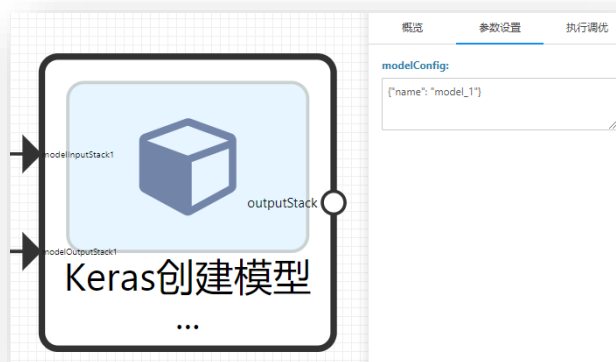
3 创建模型

在此步骤连接模型的输入、输出。

modelInputStack1 对应模型输入层的输出端子。

modelOutputStack 对应模型输出也就是连接最一层的输出端子。

modelConfig 参数中可以定义模型变量的名称。如果没有定义程序中会自动生成。



创建模型

4 加载自带数据集

在 dataset 中选择 mnist 数据集（其他数据集暂不支持）。

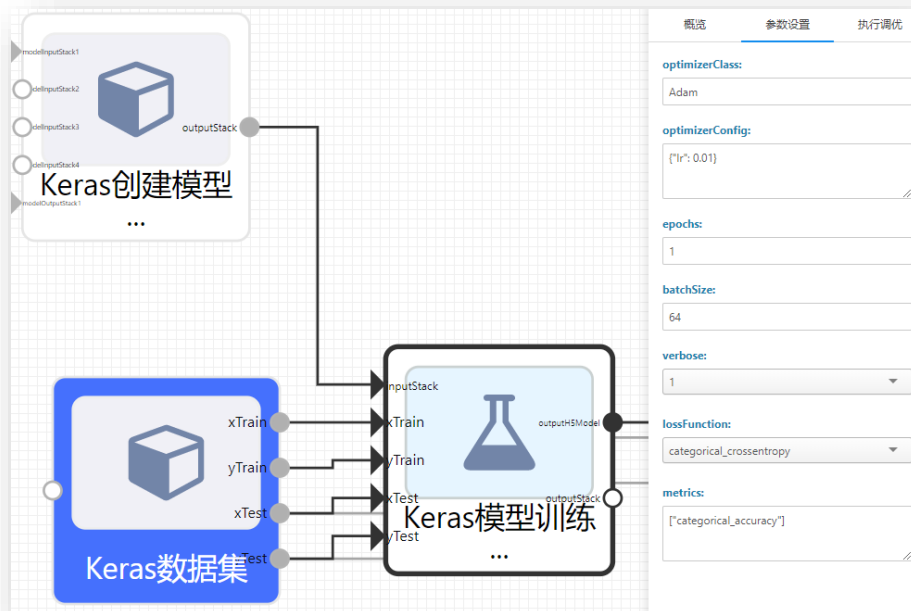


自带数据集

5 模型训练

拖出“Keras 模型训练”节点，并配置。

此节点内部将会调用 Keras model 的 compile 和 fit 类函数训练模型并输出.h5 格式模型权值文件，右面板中各个配置参数的定义可以参考 Keras [官方文档](#)。

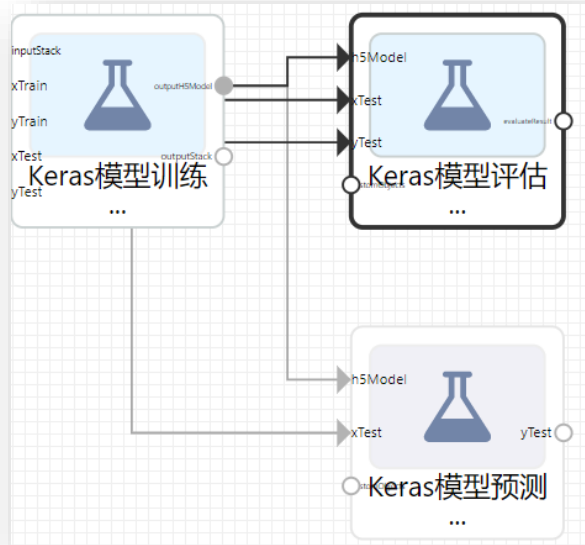


模型训练

训练进度可以在右面板的执行日志中查看。此节点建议使用 GPU 镜像运行。切换镜像方法参考上一章节。

5 模型评估和预测

这两个节点对应 Keras model 的 [evaluate](#) 和 [predict](#) 类函数。



模型评估和预测

模型评估运行完输出的是一个 Json 类型的数据，包括评估的 loss 和准确率，
比如，

```
{  
  "loss": 0.08969532091580332,  
  "categorical_accuracy": 0.9700999855995178  
}
```

此结果可以在右面板的查看结果中下载



查看模型评估和结果

模型预测输出的是个 Numpy 数据，即 [model.predict](#) 输出的结果，该文件可以在右面板下载。

6 运行模板

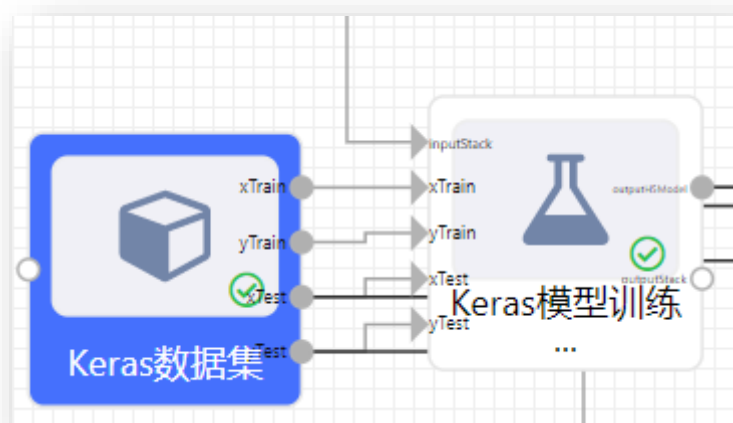
选中模型训练节点，在运行选择“运行到此处”将会运行改节点和上游节点。

如果选择“运行该节点”将会只运行当前节点。



模型评估和预测

运行成功的节点会显示绿色打钩图标。



运行成功的节点

当模型训练运行成功后，对于下游的模型评估和模型预测我们只需要选择“运行该节点”即可。你可以在右面板的查看任意运行成功节点的结果和输出数据。

Keras 图片分类（迁移学习）

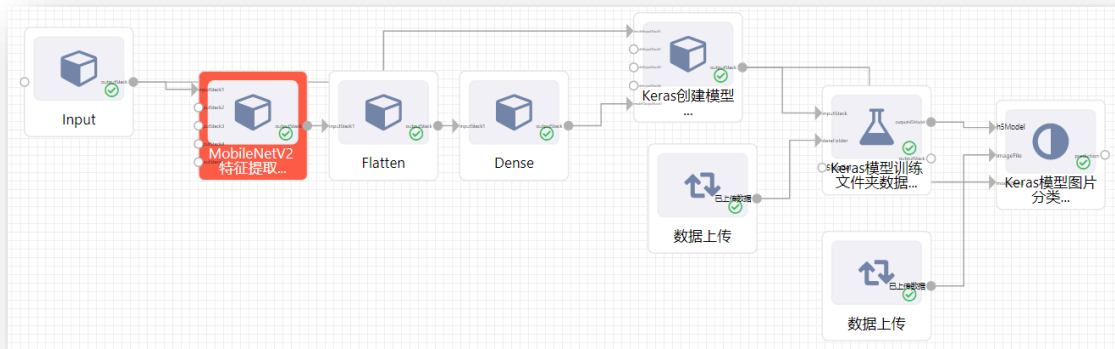
模板 ID：3894

我们在 ImageNet 预先训练的模型基础上通过迁移学习的方式训练一个可以区分猫和狗图片的模型，数据可以在微软[网站](#)下载或从算盘演示模板中的和模型训练相连的上传数据节点下载。当然你也可以通过上传自定义图片集实现其他类型图片分类的应用。



猫狗图片分类

Keras 图片分类（迁移学习）模型，上传自定义图片，训练两个 epoch 达到 92.3% 的验证数据集准确率。此案例建议在配置了 GPU 加速的机器上运行，因为训练图片分类模型需要消耗较多计算资源，GPU 可以通过并行计算加速模型训练。



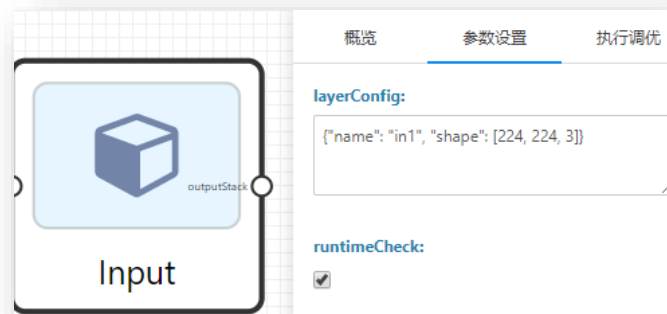
图片分类（迁移学习）实例

1 添加输入层

配置模型输入层的名称和 shape, 图片在加载模型前会被调整为指定分辨率。

如下配置, RGB 彩色图片将会被调整为 244 x 244 大小。

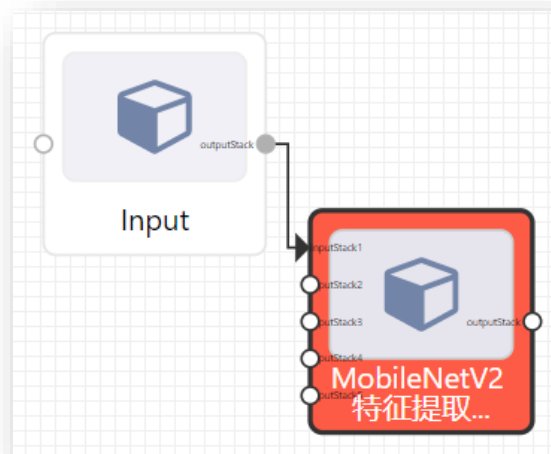
```
{"name": "in1", "shape": [224, 224, 3]}
```



添加输入层

2 添加特征提取网络

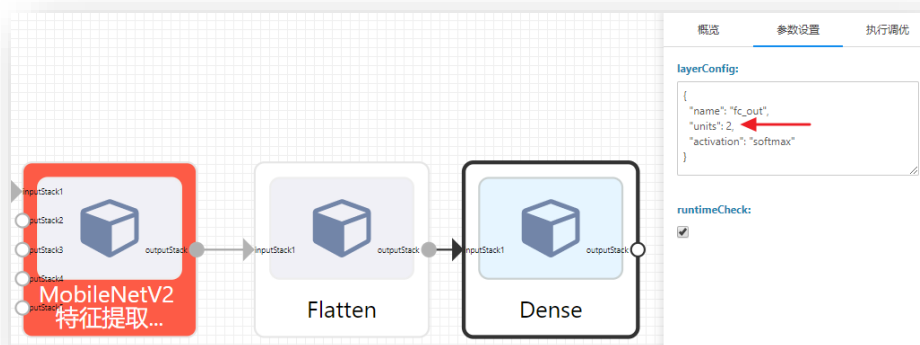
特征提取网络是已经使用大规模数据训练过的深度学习卷积网络, 可以作为迁移学习使用。特征提取网络是训练好的 ImageNet 模型去掉分类层并冻结所有层的结果。此实例中我们选择了轻量级的 MobileNetV2 网络, 相比其他参数更多的 ImageNet 模型而言, 此网络预训练和测速度更快, 消耗的计算资源更少。



添加特征提取网络节点

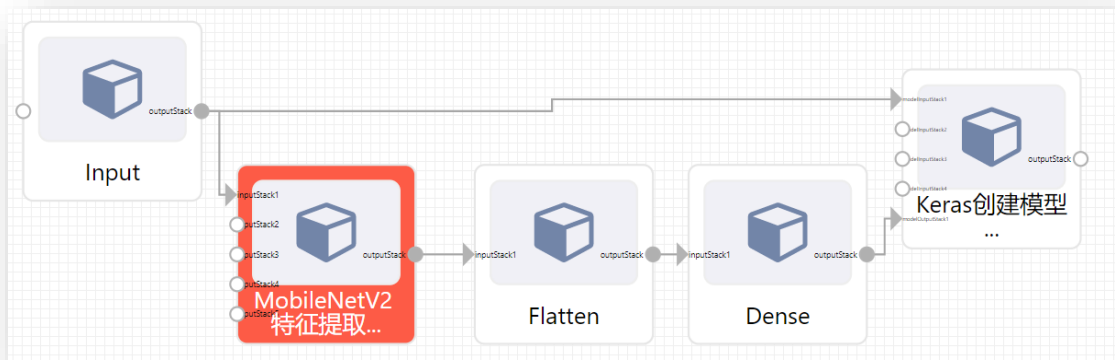
3 添加分类网络并构建模型

特征提取网络输出到 Flatten 层, 将多维度的图片特征张量平铺展开为 1 维度的张量, Dense 全连接层配置为分类。接下来我们上传的数据一共有两种分类(cats 和 dogs), 所以 units 设为 2。



添加分类网络

创建模型方式和上一个案例类似。

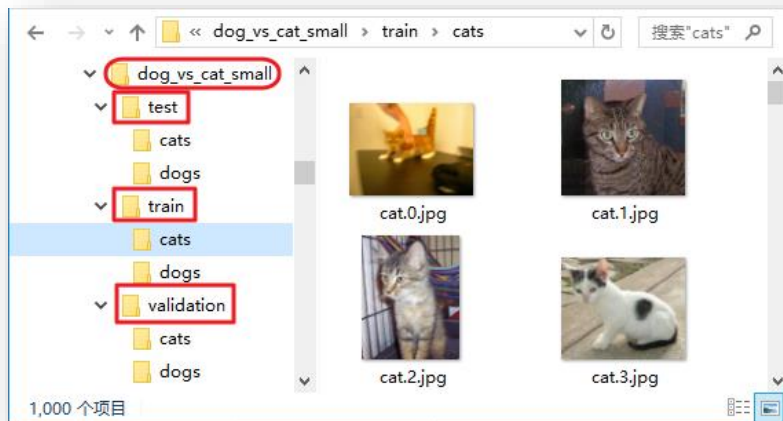


创建模型

4 添加模型训练节点

“Keras 模型训练文件夹数据”节点可以读取文件夹中的图片数据训练 Keras 模型。该节点支持加载文件夹压缩的一个 zip 包。

案例中模型训练后将可以识别 2 分类 (cats 和 dogs)，参考压缩包内部文件夹结构如下图。



参考文件夹结构

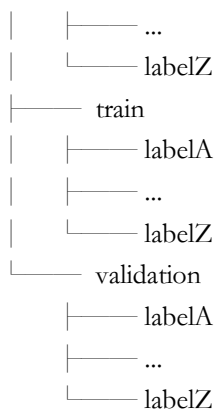
每个类应包含一个子目录。每个子目录目录树中可以为任何的 PNG、JPG、BMP、PPM 或 TIF 图像，路径中不能有中文，图片名称命名没有其他要求。

data

```

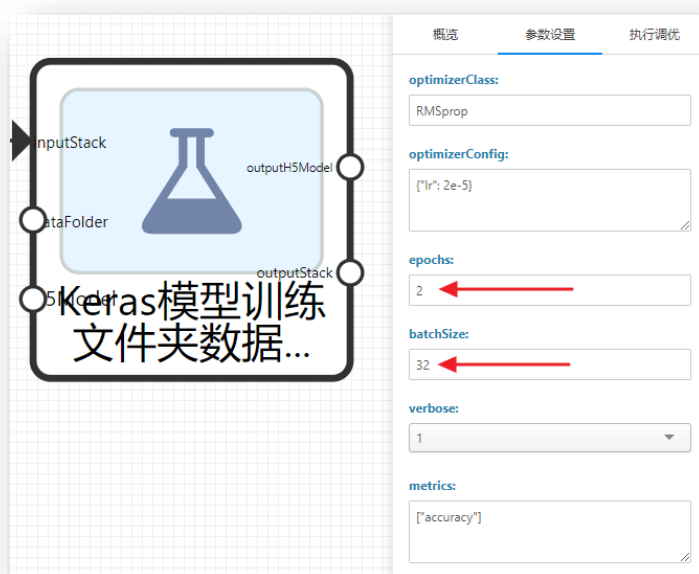
|—— test
|
|—— labelA

```



train, test, validation 分别代表训练、测试和评估集。test 和 validation 文件夹可以没有。更多细节可以参考 Keras [flow_from_directory](#) 文档。

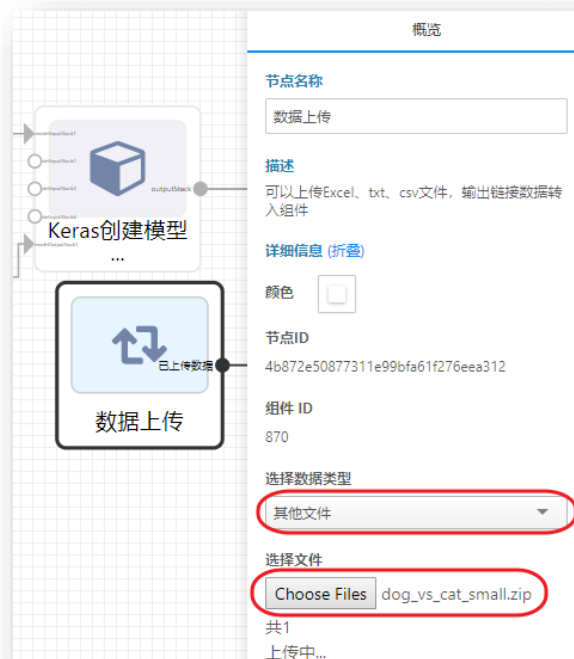
参考下图配置训练节点。epochs 决定模型遍历训练集数据的轮次，训练轮次数越多往往最终准确率较高但也可能出现过拟合情况。用户也可根据 GPU 显存大小配置适当调高 batchSize 以加快训练速度或接受默认配置。



参考文件夹结构

5 数据上传节点

如图配置数据上传节点并上传 zip 文件。

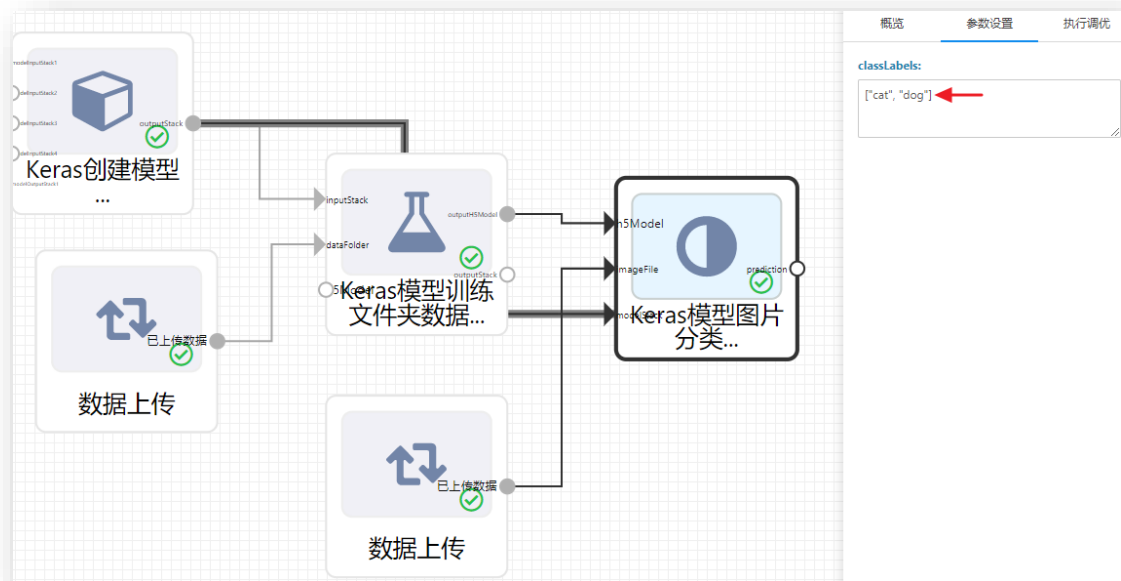


数据上传节点

6 模型图片分类(预测)

模型训练节点训练结束后会输出模型权值文件可供接下来的分类预测节点加载。

拖出 Keras 模型图片分类节点。按下图连接并配置该节点。classLabels 可选配置项，如果提供了将会在预测标签中显示对应的分类标签。



模型图片分类(预测)节点配置

在新添加的数据上传节点上传一张图片文件（cat 或 dog 的图片），在上传前请将文件命名为 file，去掉文件扩展名。

7 运行模板

如果在“Keras 模型图片分类”节点选择运行到此节点，将会运行模板中所有节点。在“Keras 模型训练文件夹数据”的执行日志可以看到 Keras 模型训练的进度和最终 loss 和准确度。

“Keras 模型图片分类”节点的查看结果可以看到模型对上传图片的预测结果。

如下图显示的模型认为上传的图片属于 cat 的概率是 99.8%。



Keras模型图片分类...

输出结果

[查看结果 - prediction](#)

执行日志

[新窗口打开](#) | [预览](#)

查看文件

全部文件夹

文件名	文件大小
 datajson	1KB

```
{
  "class_index": 0,
  "class_label": "cat",
  "probability": 0.998469889163971,
  "output": [
    0.998469889163971,
    0.0015300966333597898
  ]
}
```



模型图片分类(预测)节点结果