

1、检查点

- 介绍如何保存和恢复通过Estimator构建的TensorFlow模型
- TensorFlow提供两种模型格式：
  - checkpoints：检查点，这种格式依赖于创建模型的代码
  - [SavedModel](#)：保存的模型，不依赖于创建模型的代码

2、示例代码

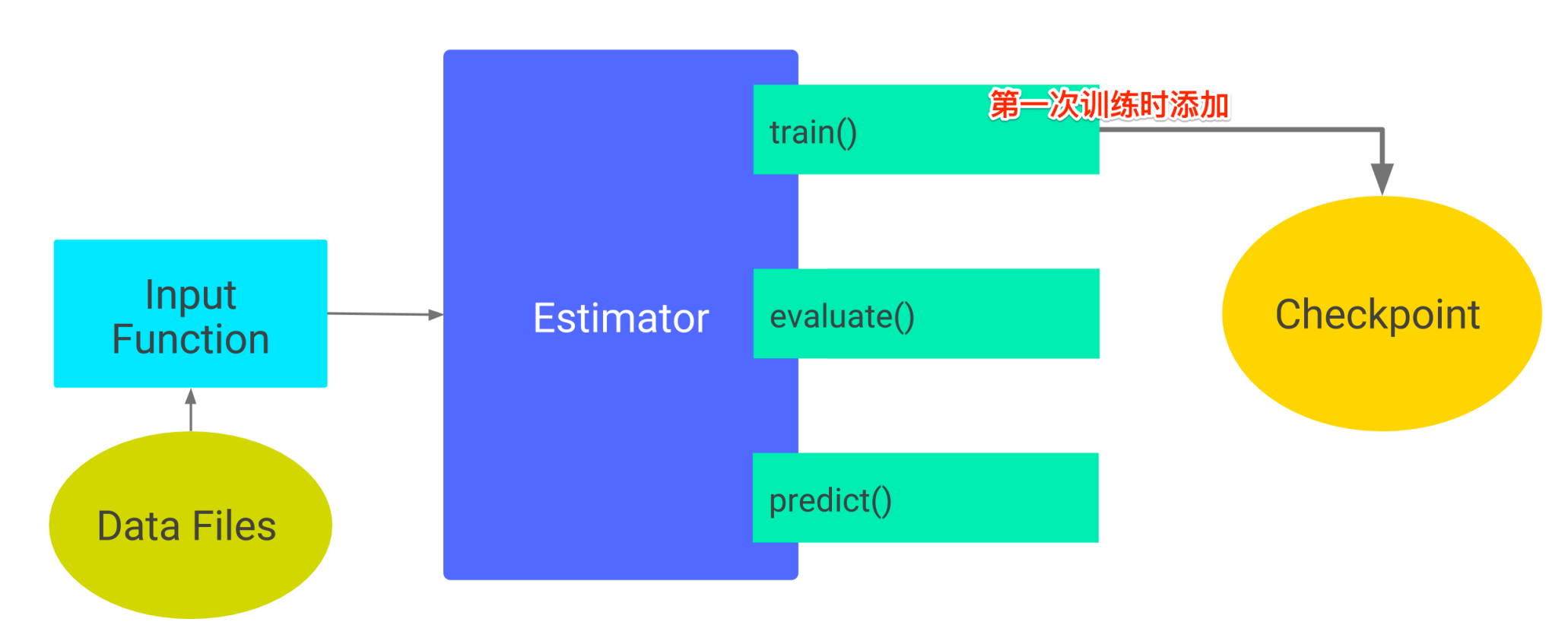
- 依赖于前面的鸢尾花分类代码
- 在premade\_estimator.py的基础上增加细微修改

3、保存经过部分训练的模型

- Estimators自动将以下内容写入磁盘：
  - checkpoints：检查点，训练期间所创建的模型版本
  - event files：事件文件，其中包含的信息可用于[TensorBoard](#)创建可视化图表
- 指定保存目录：设置【model\_dir】参数

```
classifier = tf.estimator.DNNClassifier(  
    feature_columns=my_feature_columns,  
    hidden_units=[10, 10],  
    n_classes=3,  
    model_dir='models/iris')
```

- 第一次调用train方法会将检查点和其他文件添加到model\_dir目录中

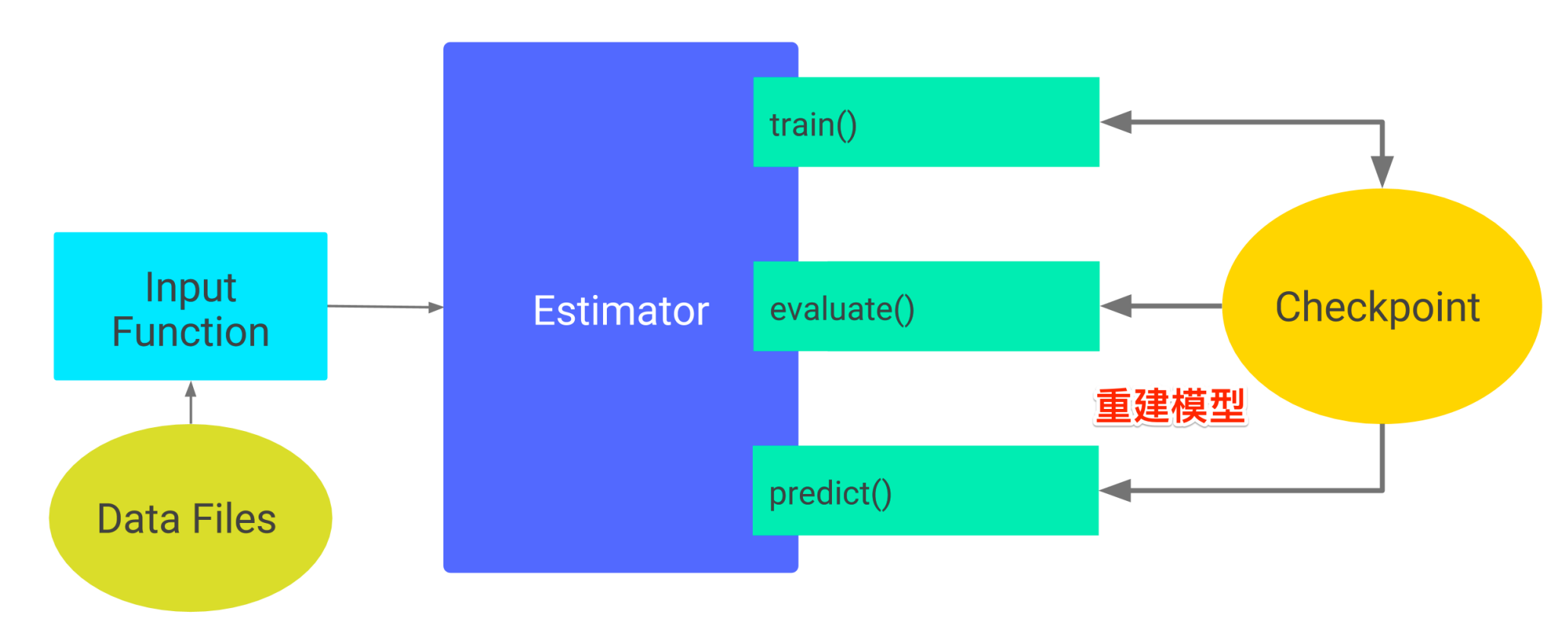


- 训练第一步和最后一步创建了检查点
- 默认检查点目录：如果不指定model\_dir，默认写入Python的 tempfile.mkdtemp函数选择的临时目录中
  - print(classifier.model\_dir)
  - 例如： /var/folders/0s/5q9kfzfy3gx2knj0vj8p68yc00dhcr/T/tmpYm1Rwa
- 检查点频率，默认情况下：
  - 每 10 分钟（600 秒）写入一个检查点
  - 在 train 方法开始（第一次迭代）和完成（最后一次迭代）时写入一个检查点
  - 只在目录中保留 5 个最近写入的检查点
- 更改检查点频率时间安排：
  - 创建一个 RunConfig 对象来定义所需的时间安排
  - 在实例化 Estimator 时，将该 RunConfig 对象传递给 Estimator 的 config 参数
- 例如，更改为每20分钟保存一次，并保留10个最近写入的检查点

```
my_checkpointing_config = tf.estimator.RunConfig(  
    save_checkpoints_secs = 20*60, # Save checkpoints every 20 minutes.  
    keep_checkpoint_max = 10,      # Retain the 10 most recent checkpoints.  
)  
  
classifier = tf.estimator.DNNClassifier(  
    feature_columns=my_feature_columns,  
    hidden_units=[10, 10],  
    n_classes=3,  
    model_dir='models/iris',  
    config=my_checkpointing_config)
```

4、恢复模型

- 第一次调用 Estimator 的 train 方法时，TensorFlow 会将一个检查点保存到 model\_dir 中
- 随后每次调用 Estimator 的 train、eval 或 predict 方法时，都会发生下列情况：
  - Estimator 通过运行 model\_fn() 构建模型图。（详见：[model\\_fn\(\)](#)）
  - Estimator 根据最近写入的检查点中存储的数据来初始化新模型的权重
- 也就是，一旦存在检查点，TensorFlow就会在后续每次调用train()、evaluate() 或 predict() 时重建模型



- 避免不当恢复：
  - 通过检查点恢复模型的状态这一操作仅在模型和检查点兼容时可行
  - 例如，假设训练了一个 DNNClassifier Estimator，它包含 2 个隐藏层且每层都有 10 个节点
  - 假如后面更改了模型，且之前的检查点已存在，则由于新模型与检查点中的状态不兼容导致报错：

```
...  
  
InvalidArgumentError (see above for traceback): tensor_name =  
dnn/hiddenlayer_1/bias/t_0/Adagrad; shape in shape_and_slice spec [10]  
does not match the shape stored in checkpoint: [20]
```

- 需要注意上述情况

5、总结

- 检查点提供了一种简单的自动机制来保存和恢复由 Estimator 创建的模型
- 更多阅读：《TensorFlow 编程人员指南》的保存和恢复一章
  - 使用低阶 TensorFlow API [保存和恢复](#)模型
  - 以 SavedModel 格式导出和导入模型，这是一种独立语言的、可恢复的序列化格式