1. Keras Sequential

简化tensorflow2的网络搭建与训练过程，用Sequential API将网络的各个层按照顺序叠加起来，只需要提供包含各个层的列表，Keras就可将它们自动首尾相连，并返回模型。

model = tf.keras.models.Sequential([

tf.keras.layers.Flatten(),

tf.keras.layers.Dense(100, activation=tf.nn.relu),

tf.keras.layers.Dense(10),

tf.keras.layers.Softmax()

])

1. Keras 的 Functional API

为了构建更复杂的模型，使用Functional API，将层作为可调用的对象返回张量，并将输入向量和输出向量提供给 tf.keras.Model 的 inputs 和 outputs 参数。即按照网络的顺序调用各个层，最后将输入和输出放到tf.keras.Model参数中。

inputs = tf.keras.Input(shape=(28, 28, 1))

2. x = tf.keras.layers.Flatten()(inputs)

3. x = tf.keras.layers.Dense(units=100, activation=tf.nn.relu)(x)

4. x = tf.keras.layers.Dense(units=10)(x)

5. outputs = tf.keras.layers.Softmax()(x)

6. model = tf.keras.Model(inputs=inputs, outputs=outputs)

1. compile 配置训练过程

按照上面将网络模型配置好以后，可以通过tf.keras.Model 的 compile方法配置训练过程，即设置训练优化器，计算损失函数以及评估指标。

tf.keras.Model.compile 接受 3 个重要的参数:

1. optimizer : 优化器， 从 tf.keras.optimizers中选择
2. loss : 损失函数， 从 tf.keras.losses中选择
3. metrics: 评估指标，从 tf.keras,metrics中选择

model.compile(

2. optimizer=tf.keras.optimizers.Adam(learning\_rate=0.001),

Keras Pipeline \*

Keras Sequential/Functional API 模式建立模型

使用 Keras Model 的 compile 、 fit 和 evaluate 方法训练

和评估模型

Keras Pipeline \*

3. loss=tf.keras.losses.sparse\_categorical\_crossentropy,

4. metrics=[tf.keras.metrics.sparse\_categorical\_accuracy]

5. )

1. fit方法开始训练

tf.keras.Model.fit 方法开始训练模型，该方法接受5个参数：

1. x： 训练数据
2. y： 训练数据标签
3. epochs: 迭代次数
4. batch\_size : 每一批的数据量
5. validation\_data: 验证数据，在训练过程中监控模型的性能

model.fit(data\_loader.train\_data, data\_loader.train\_label,

epochs=num\_epochs, batch\_size=batch\_size)

1. evaluate 评估训练效果

训练完后，用测试数据和测试标签对模型的性能进行测试，只需两个参数：

1. 测试数据
2. 测试数据标签