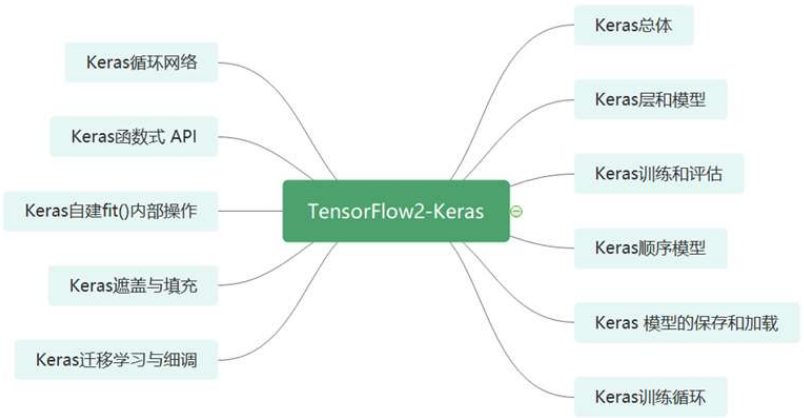
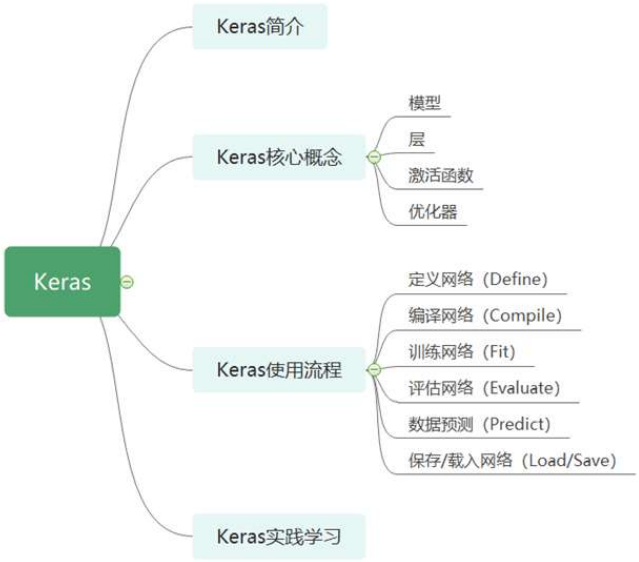
TensorFlow2-Keras

Keras
总体

TensorFlow-Keras总体介绍



本小节导学

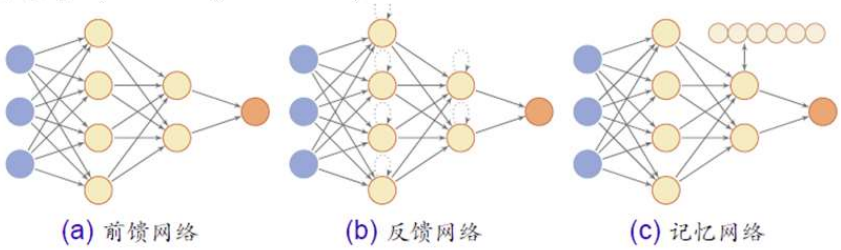


预备：网络结构

多层深度网络

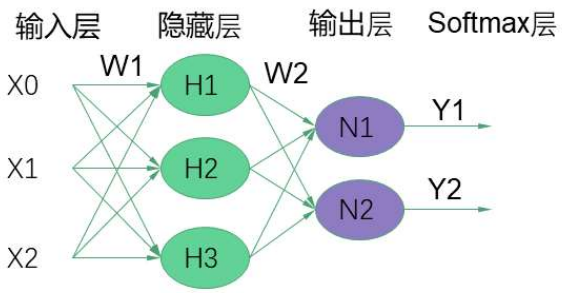
人工神经网络

- 人工神经网络，是按照拓扑连接结构，将大量的神经元之间连接起来，构成人工神经元的网络结构
- 不同神经元之间的连接关系。
  - 前馈网络（feedforward）
  - 反馈网络（feedback）
  - 记忆网络（memory network）



多层前馈网络

- 前馈网络（FNN），举例：
  - 多层全连接网络（FCN）、多层感知机（MLP）、多个密集层网络（Dense）
- 示例网络：有3个输入，2个输出；输入层、隐藏层，输出层；
- 隐藏层、输入层，共有5个神经元。



典型的网络结构

- 卷积网络CNN
- （Convolutional neural network，简称CNN）
- 前馈网络：计算快，可并行化

- 循环网络RNN
- （Recurrent neural network，简称RNN）
- 反馈网络：当前时刻的输出，可以作为下一时刻的输入

Keras简介

TensorFlow2

什么是Keras?

- Keras 是一套**高级API（应用程序接口）**，用来**快速搭建**深度神经网络
- tensorflow.keras 是基于TensorFlow2实现的**快速搭建**深度神经网络的程序库
- http://Keras.io

Keras历史

- Keras是由弗朗索瓦·肖莱（Francois Chollet）开发的一套高层API框架
- 先前，Keras对多个深度学习框架的进一步**封装**，抽取共性，形成一致的API，而不用关心底层的框架实现
  - 底层框架，包括：TensorFlow、 Theano和CNTK。
- 目前，Keras已经完全并入TensorFlow2，TensorFlow 2.0以上版本已经集成了Keras框架，提供了tensorflow.keras包。

https://tensorflow.google.cn/guide/keras/overview

Keras优点

- Keras设计采用极简主义原则(Minimalist)，是一套高度模块化的神经网络架构库。
- Keras具有方便易用的特点，如简洁的网络定义方式，常用技巧的封装。
- Keras保留了足够的扩展性，同时具有灵活性，可以自行实现各种层（layers）。
- Keras用户众多，适合快速学习、快速搭建深度学习网络

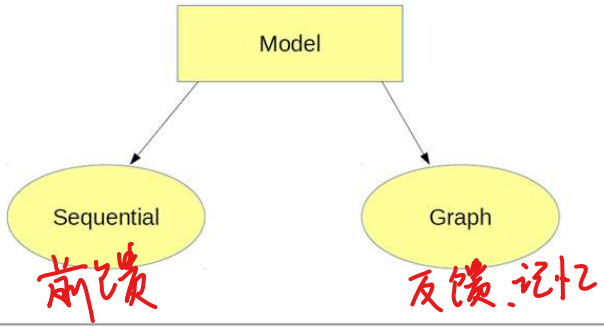
```
import tensorflow as tf

from tensorflow import keras
```

Keras核心概念

TensorFlow2

Keras 核心概念-模型

- Keras核心的数据结构叫模型（Model）
- 模型是组织层的一种方式，包括架构（连接关系）和权重参数。
- 最简单的Model类型是顺序模型（Sequential model）。
  - 顺序模型是指串接很多层的线形管道 (a linear pile of layers)。



Keras核心概念- 顺序模型

- 顺序模型（Sequential model），又称为序列模型
- 网络结构是以一个栈的形式给出，一层接一层。

```
# Define Sequential model with 3 layers
model = keras.Sequential(
    [
        layers.Dense(2, activation="relu", name="layer1"),
        layers.Dense(3, activation="relu", name="layer2"),
        layers.Dense(4, name="layer3"),
    ]
)
# Call model on a test input
x = tf.ones((3, 3))
y = model(x)
```

```
# Create 3 layers
layer1 = layers.Dense(2, activation="relu", name="layer1")
layer2 = layers.Dense(3, activation="relu", name="layer2")
layer3 = layers.Dense(4, name="layer3")

# Call layers on a test input
x = tf.ones((3, 3))
y = layer3(layer2(layer1(x)))
```

Keras核心概念- 层-1

- Module: tf.keras.layers
- Keras 层(layers) 包括:
- Dense, Activation, Dropout, Flatten, Reshape, Permute, RepeatVector, Lambda, ActivityRegularization, Masking.

Keras 核心概念-层-2

- Module: tf.keras.layers
- Keras 层还包括:
- Convolutional Layers, Pooling Layers, Locally-connected Layers, Recurrent Layers, Embedding Layers, Advanced Activations Layers, Normalization Layers, Noise layers.

Keras核心概念-激活函数

- Module: tf.keras.activations

- Keras 激活函数（activations）包括:
  - softmax, **elu**, softplus, softsign, **relu**, tanh, sigmoid, hard_sigmoid, linear 等等
- 函数表示式如下:
  - softmax(x), elu(x, alpha=1.0), softplus(x), softsign(x), relu(x, alpha=0.0, max_value=None), tanh(x), sigmoid(x), hard_sigmoid(x), linear(x).

Keras核心概念-优化器

- Keras 优化器（optimizers）包括:
  - Optimizer基类
  - SGD
  - RMSprop, Adagrad, Adadelta, Adam, Adamax
  - **Nadam**
- SGD是最简单的优化器之一，带有动量参数（momentum）收敛快
- RMSprop, Adagrad, Adam都是带**学习率自适应调整的算法**的优化器。
- Nadam (Nesterov Adam optimizer)是采用Nesterov加速梯度算法的优化器。
  - Nadam的是基于之前的所有梯度（全局的信息）更新模型参数，而不仅仅基于当前的梯度（局部信息）
  - Nadam可以进一步提升训练收敛的速度的

Keras使用流程

TensorFlow2

mnist数据集

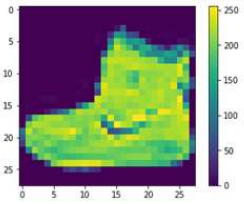
- 该数据集有7万张图片
  - 手写数字图片（hand-written digits）
  - 大小：28x28，灰度数值0~255
- http://yann.lecun.com/exdb/mnist/

```
mnist = tf.keras.datasets.mnist
(x_train, y_train), (x_test, y_test) = mnist.load_data()
```

x_train.shape (60000, 28, 28)

fashion-mnist数据集

- 该数据集有7万张图片
  - 图像是 28x28 的 NumPy 数组
  - ~~像素值介于 0 到 255 之间。~~ RGB 3
  - 标签是整数数组，介于 0 到 9 之间。
  - 标签对应于图像所代表的服装类：
  - 大小：28x28，彩色图像



标签	类
0	T恤/上衣
1	裤子
2	套头衫
3	连衣裙
4	外套
5	凉鞋
6	衬衫
7	运动鞋
8	包
9	短靴

```
fashion_mnist = keras.datasets.fashion_mnist

(train_images, train_labels), (test_images, test_labels) =
fashion_mnist.load_data()
```

https://github.com/zalandoresearch/fashion-mnist

Keras使用流程

1. 定义网络（Define）
2. 编译网络（Compile）
3. 训练网络（Fit）
4. 评估网络（Evaluate）
5. 数据预测（Predict）
6. 保存/载入网络（Load/Save）

1、定义网络

- 示例，基于顺序模型
  - 网络结构是以一个栈的形式给出，一层接一层。
- 代码解读：
  - 第一层的input_shape/input_dim参数指定了输入层的大小。
    - Input_dim是第一层的参数列表，而与第一层是什么类型的网络无关。
    - 这里 指定一个784维的输入。
  - 最后一层一般是softmax作为输出层。
    - 给出了10个输出，表示该网络解决10类分类的问题。

```
from keras.models import Sequential
from keras.layers import Dense, Activation

model = Sequential([
    Dense(32, input_dim=784),
    Activation('relu'),
    Dense(10),
    Activation('softmax'),
])
```

2、编译网络

- 编译网络
  - 给定网络的训练目标
  - 给定网络的优化算法
- 代码解读
  - 'loss'：是指网络使用多类交叉熵作为优化目标。
  - 'optimizer'：是指使用RMSprop算法进行梯度下降。
  - 'metrics'：是指在训练过程中，希望观察到准确度这个指标是如何变化的。

```
model.compile(optimizer='rmsprop',
loss='categorical_crossentropy', metrics=['accuracy'])
```

3、网络训练

- 代码解读：
  - 产生1000个符合input_shape的数据，以及标签，然后传给**model.fit()**，从而开始迭代训练过程。
- 训练过程会动态输出loss accuracy等参数。

```
import numpy as np

data = np.random.random((1000, 784))

labels = np.random.randint(2, size=(1000, 1))

# train the model, iterating on the data in batches of 32 samples
model.fit(data, labels, nb_epoch=10, batch_size=32)
```

4、评估网络 测试

- 评估网络（Evaluate）：检查训练效果
- 方法：model.evaluate()

```
model.evaluate(self, x, y, batch_size=32, verbose=1,
sample_weight=None)
```

5、数据预测（Predict）

- 数据预测（Predict）
- 方法：model.predict()

```
model.predict(self, x, batch_size=32, verbose=0)
```

6、保存/载入网络

- 将保存/载入网络结构与参数
- 方法：
  - model.save() 或 tf.keras.models.save_model()
  - tf.keras.models.load_model()

- 不推荐：采用hdf5格式保存网络参数数据。

```
from keras.models import load_model

model.save('./model.h5')

model2 = load_model('./model.h5')
```

```
import tensorflow as tf

#载入并准备好 MNIST 数据集。将样本从整数转换为浮点数
mnist = tf.keras.datasets.mnist

(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

#将模型的各层堆叠起来，以搭建 tf.keras.Sequential 模型。为训练选择优化器和损失函数：
model = tf.keras.models.Sequential()

model.add(tf.keras.layers.Flatten(input_shape=(28, 28)))
model.add(tf.keras.layers.Dense(128, activation='relu'))
model.add(tf.keras.layers.Dropout(0.2))
model.add(tf.keras.layers.Dense(10, activation='softmax'))

model.compile(optimizer='adam',loss='sparse_categorical_crossentropy',metrics=['accuracy'])

#训练并验证模型：
model.fit(x_train, y_train, epochs=5)

model.evaluate(x_test,  y_test, verbose=2)
```

keras-mnist.ipynb

Keras完整的mnist示例- TensorFlow 2版本

https://tensorflow.google.cn/tutorials/quickstart/beginner

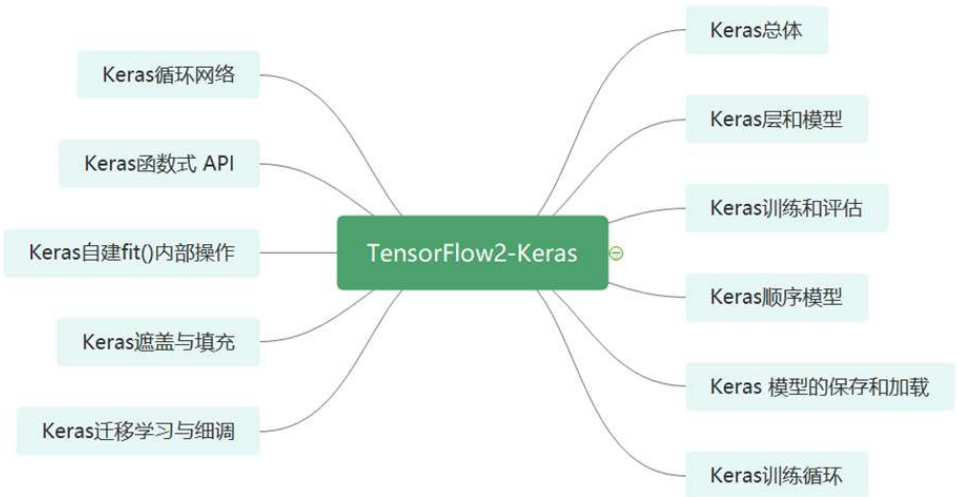
想一想，练一练（体验 Keras）

- 请大家运行一下以上代码
- 将结果投稿一下

Keras实践学习

TensorFlow2

导学



Keras总体-1

- Keras层和模型
  - https://tensorflow.google.cn/guide/keras/custom_layers_and_models
- Keras训练和评估
  - https://tensorflow.google.cn/guide/keras/train_and_evaluate
- Keras顺序模型
  - https://tensorflow.google.cn/guide/keras/sequential_model
- Keras 模型的保存和加载
  - https://tensorflow.google.cn/guide/keras/save_and_serialize
- Keras训练循环
  - https://tensorflow.google.cn/guide/keras/writing_a_training_loop_from_scratch

Keras总体-2

- Keras循环网络
  - https://tensorflow.google.cn/guide/keras/rnn
- Keras自建fit()内部操作
  - https://tensorflow.google.cn/guide/keras/customizing_what_happens_in_fit
- Keras函数式 API
  - https://tensorflow.google.cn/guide/keras/functional
- Keras遮盖与填充
  - https://tensorflow.google.cn/guide/keras/masking_and_padding
- Keras迁移学习与细调
  - https://tensorflow.google.cn/guide/keras/transfer_learning

Keras-小结

- Keras的简介，什么是Keras?
- Keras的基本概念:
  - 模型（models）、层（layers）、激活函数（activation）
  - 优化器（Optimizer）
- Keras的使用的6个步骤
  - 定义网络（Define）
  - 编译网络（Compile）
  - 训练网络（Fit）
  - 评估网络（Evaluate）
  - 数据预测（Predict）
  - 保存/载入网络（Load/Save）

参考书

- 英文版:
  - Francois Chollet, Deep Learning with Python, Manning press, November 2017.
- 中文版:
  - [美] 弗朗索瓦·肖莱（Francois Chollet） 著，Python深度学习，人民邮电出版社，2018年8月.
- 智能硬件TensorFlow实践，清华大学出版社，2017.

谢谢指正！