

# Keras中文教程 (index.html)

## [Keras中文教程](#)

---

### 简介

[Keras:基于Python的深度学习库 \(15211948875047.html\)](#)

---

### Keras新手指南

[Paras相关的一些基本概念 \(15211966310968.html\)](#)

[常见问题 \(15212128885527.html\)](#)

---

### 快速开始

[序贯 \(Sequential\) 模型 \(15217249523937.html\)](#)

[函数式 \(Functional\) 模型 \(15217255992155.html\)](#)

---

### 模型

[关于Keras模型 \(15218626696660.html\)](#)

[序贯 \(Sequential\) 模型API \(15218629950884.html\)](#)

---

### 网络层

[关于Keras的“层” \(Layer\) \(15221585677979.html\)](#)

[常用层 \(15221587162369.html\)](#)

[卷积层 \(15222414304981.html\)](#)

[池化层 \(15223314263378.html\)](#)

[局部连接层LocallyConncceted \(15224185106717.html\)](#)

[循环层Recurrent \(15224192481292.html\)](#)

[嵌入层 Embedding \(15224977160163.html\)](#)

[融合层Merge \(15225897511298.html\)](#)

[高级激活层Advanced Activation \(15226713351659.html\)](#)

[规范层BatchNormalization \(15227594692825.html\)](#)

[噪声层Noise \(15228463724267.html\)](#)

[包装器Wrapper \(15229421313123.html\)](#)

[编写自己的层 \(15230263507838.html\)](#)

---

## 数据预处理

[序列预处理 \(15231946023898.html\)](#)

[文本预处理 \(15232862634198.html\)](#)

[图片预处理 \(15233673620241.html\)](#)

---

## 网络配置

[损失函数loss \(15234585829220.html\)](#)

[优化器optimizers \(15235421820644.html\)](#)

[激活函数Activations \(15236275283380.html\)](#)

[性能评估Metrics \(15238871410252.html\)](#)

[初始化方法Initializers \(15239768106885.html\)](#)

[正则项Regularizer \(15241458542033.html\)](#)

[约束项Constraint \(15245791260462.html\)](#)

[回调函数Callback \(15247524557227.html\)](#)

---

## 协助使用Keras

[预训练模型Application \(15248386204042.html\)](#)

[常用数据库Dataset \(15253598632568.html\)](#)

[可视化visualization \(15254400186618.html\)](#)

[utils 工具 \(15257016540180.html\)](#)

---

# Keras:基于Python的深度学习库

## 什么是Keras?

Keras是一个高层神经网络API，Keras由纯Python编写而成并基Tensorflow、Theano以及CNTK后端。Keras 为支持快速实验而生，能够把你的idea迅速转换为结果，如果你有如下需求，请选择Keras：

- 简易和快速的原型设计 (keras具有高度模块化, 极简, 和可扩充特性)
- 支持CNN和RNN, 或二者的结合
- 无缝CPU和GPU切换

Keras适用的Python版本是: `Python 2.7~3.6`

Keras的设计原则是:

- 用户友好: Keras是为人类而不是天顶星人设计的API。用户的使用体验始终是我们考虑的首要中心内容。Keras遵循减少认知困难的最佳实践: Keras提供一致而简洁的API, 能够极大减少一般应用下用户的工作量, 同时, Keras提供清晰和具有实践意义的bug反馈。
- 模块性: 模型可理解为一个层的序列或数据的运算图, 完全可配置的模块可以用最少的代价自由组合在一起。具体而言, 网络层、损失函数、优化器、初始化策略、激活函数、正则化方法都是独立的模块, 你可以使用它们来构建自己的模型。
- 易扩展性: 添加新模块超级容易, 只需要仿照现有的模块编写新的类或函数即可。创建新模块的便利性使得Keras更适用于先进的研究工作。
- 与Python协作: Keras没有单独的模型配置文件类型 (作为对比, caffe有), 模型由python代码描述, 使其更紧凑和更易debug, 并提供了扩展的便利性。

## 关于Keras中文文档

本文档是Keras文档的中文版, 包括keras.io的全部内容, 以及更多的例子、解释和建议。

由于苦李水平和研究方向所限, 无法对所有模块都非常精通, 因此文档中不可避免的会出现各种错误、疏漏和不足之处。如果您在使用过程中有任何意见、建议和疑问, 欢迎发送邮件到likuli@foxmail.com (mailto:likuli@foxmail.com)与我取得联系。

## 当前版本与更新

如果你发现本文档提供的信息有误, 有两种可能:

- 你的Keras版本过低: Keras是一个发展迅速的深度学习框架, 请保持你的Keras与官方最新的release版本相符
- 这里的中文文档没有及时更新: 如果是这种情况, 请email给苦李, 我会尽快更新

## 快速上手Keras

Keras的核心数据结构是“模型”, 模型是一种组织网络层的方式。Keras中主要的模型是Sequential模型, Sequential是一系列网络层按顺序构成的栈。

你也可以查看函数式模型来学习建立更复杂的模型。

Sequential模型如下:

```
from keras.models import Sequential
model = Sequential()
```

将一些网络层通过 `.add()` 堆叠起来，就构成了一个模型：

```
from keras.layers import Dense, Activation

model.add(Dense(units=64, input_dim=100))
model.add(Activation("relu"))
model.add(Dense(units=10))
model.add(Activation("softmax"))
```

完成模型的搭建后，我们需要使用 `.compile()` 方法来编译模型：

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

编译模型时必须指明损失函数和优化器，如果需要的话，也可以自己定制损失函数。Keras的一个核心理念就是简明易用，同时保证用户对Keras的绝对控制力度，用户可以根据自己的需要定制自己的模型、网络层，甚至修改源代码。

```
from keras.optimizers import SGD
model.compile(
    loss='categorical_crossentropy',
    optimizer=SGD(
        lr=0.01, momentum=0.9,
        nesterov=True)
)
```

完成模型编译后，我们在训练数据上按batch进行一定次数的迭代来训练网络：

```
model.fit(x_train, y_train, epochs=5, batch_size=32)
```

当然，我们也可以手动将一个个batch的数据送入网络中训练，这时候需要使用：

```
model.train_on_batch(x_batch, y_batch)
```

随后，我们可以使用一行代码对我们的模型进行评估，看看模型的指标是否满足我们的要求：

```
loss_and_metrics = model.evaluate(x_test, y_test, batch_size=128)
```

或者，我们可以使用我们的模型，对新的数据进行预测：

```
classes = model.predict(x_test, batch_size=128)
```

搭建一个问答系统、图像分类模型，或神经图灵机、word2vec词嵌入器就是这么快。支撑深度学习的基本想法本就是简单的，现在让我们把它的实现也变的简单起来！

为了更深入的了解Keras，苦李建议你看看下面两个tutorial：

- 快速开始Sequential模型
- 快速开始函数式模型

还有我们的新手教程，虽然是面向新手的，但我们阅读它们总是有益的：

- Keras新手指南

在Keras代码包的examples文件夹里，我们提供了一些更高级的模型：基于记忆网络的问答系统、基于LSTM的文本的文本生成等。

## 安装

Keras使用了下面的依赖包：

```
numpy, scipy  
pyyaml  
HDF5, h5py (可选, 仅在模型的save/load函数中使用)  
如果使用CNN的推荐安装cuDNN
```

Keras可以支持以下三种后端，使用时必须至少选择一种，我们建议选择tensorflow。

当使用TensorFlow为后端时：TensorFlow

当使用Theano作为后端时：Theano

当使用CNTK作为后端时：CNTK

“后端”翻译自backend，指的是Keras依赖于完成底层的张量运算的软件包。

从源码安装Keras时，首先git clone keras的代码：

```
git clone https://github.com/fchollet/keras.git
```

接着 cd 到Keras的文件夹中，并运行下面的安装命令：

```
sudo python setup.py install
```

你也可以使用PyPI来安装Keras

```
sudo pip install keras
```

## 在Theano、CNTK、TensorFlow间切换

Keras默认使用TensorFlow作为后端来进行张量操作，如需切换到Theano，请查看[这里](#)。