

第 5 章

模型体系结构

基于第 4 章中的基本概念,现在看一个实际问题:能否使用深度学习的模型预测比特币的价格?本章将介绍如何构建一个基于深度学习的模型以预测比特币的价格。本章结束时,将把所有模型组件组合在一起,初步构建一个完整的深度学习应用程序。

本章结束时,您将能够:

- 为深度学习模型准备数据;
- 选择正确的模型架构;
- 使用 Keras(一个 TensorFlow 抽象库);
- 用经过训练的模型进行预测。

5.1 选择合适的模型体系结构

基于深度学习的研究是一项充满活力的研究。除此之外,研究人员也致力于开发新的神经网络体系结构。这些体系结构既可以解决新问题,也可以提高以前实现的体系结构的性能。本节内容将从新的体系结构和过去实现的体系结构两个方面展开。

早期实现的体系结构已经解决了大量问题,并且在开始新项目时也通常被作为合适的体系结构。较新的体系结构在特定问题上取得了巨大成功,但很难推广。有趣的是,新的体系结构可以作为下一步探索的参考,但在开始一个工程项目时,它却并不是适宜的选择。

5.1.1 常见的体系结构

考虑到体系结构的多样性,目前有两种流行的体系结构,即**卷积神经网络**(Convolutional Neural Networks, CNN)和**递归神经网络**(Recurrent Neural Networks, RNN),许多应用程序是基于它们实现的,它们都是基础的神经网络,并且都可以作为大多数项目的基础。本章还会介绍另外三个网络(出于它们在该领域的相关性):**长短期记忆网络**(Long-Short Term Memory, LSTM,一种 RNN 变体)、**对抗生成网络**(Generative

Adversarial Networks, GAN)和深度强化学习(Deep Reinforcement Learning)。这三种体系结构在解决现实问题时取得了巨大成功,但使用起来有些困难。

1. 卷积神经网络

卷积神经网络已经因处理类似网格结构的问题而声名狼藉^①,它最初是用来分类图像的,但在许多其他领域,例如从语音识别到自动驾驶,它们都得到了应用。

CNN 的本质是将密切相关的数据作为训练过程的一个要素,而不仅仅是单个数据输入,这种思想在图像方面尤其有效。在图像中,一个像素右侧的邻近像素也与该像素相关,因为它们共同构成了又一个更大的组合。在这种情况下,该组合就是神经网络训练预测的内容。因此,将几个像素组合在一起考虑比单独使用单个像素处理要更好一些。

图 5-1 所示过程的数学表示称为**卷积(convolution)**。

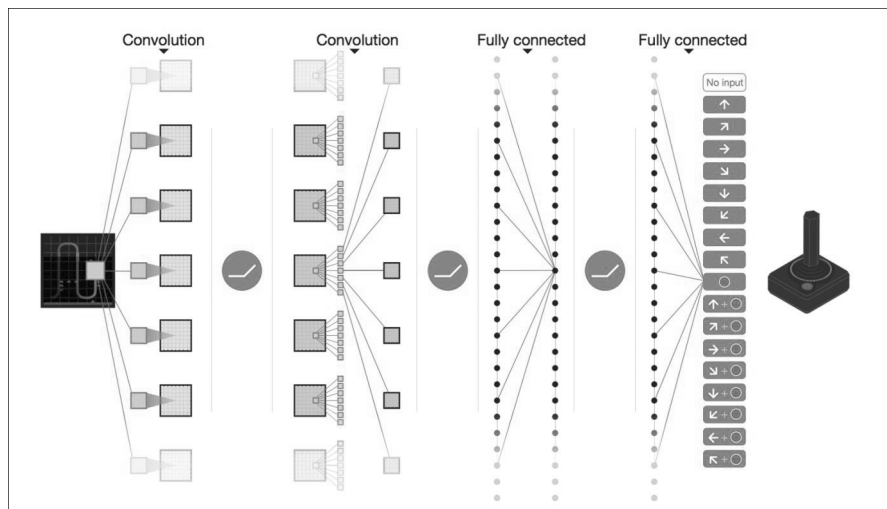


图 5-1 卷积过程图解(图片来源: Volodymyr Mnih 等)

i 如果想了解更多信息,请参阅《自然》杂志于 2015 年 2 月发表的题为 *Human-level control through deep reinforcement learning* 的文章,详见: <https://storage.googleapis.com/deepmind-media/dqn/DQNNaturePaper.pdf>。

2. 递归神经网络

卷积神经网络与一组输入一起工作,这些输入不断地更新网络各个层和节点的权重和偏差。这种方法的局限性是:在确定如何改变网络的权重和偏差时,其架构忽略了这

^① 译者注:此句为原文直译。其实,卷积神经网络在很多应用场合的性能是不错的。

些输入的顺序。

递归神经网络就是为了解决这个问题而提出的。RNN 旨在处理序列数据,这意味着在每个 epoch 中,每层都会受到前几层输出的影响。在给定的序列中,先前观察的记忆会在后验观察的评估中起作用。

由于上述问题具有时序性,因此 RNN 在语音识别中得到了成功的应用。此外,RNN 也用于翻译问题。Google 翻译目前所用的算法 **Transformer** 就使用了 RNN,即将文本从一种语言翻译成另一种语言。

i 如果想了解更多信息,请参阅 Google Research Blog 上由 Jakob Uszkoreit 于 2017 年 8 月发表的题为 *Transformer: A Novel Neural Network Architecture for Language Understanding* 的文章,详见: <https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html>。

如图 5-2 所示,根据单词出现在句子中的位置,英语单词与法语单词是相关的,因此 RNN 在语言翻译问题中非常流行。

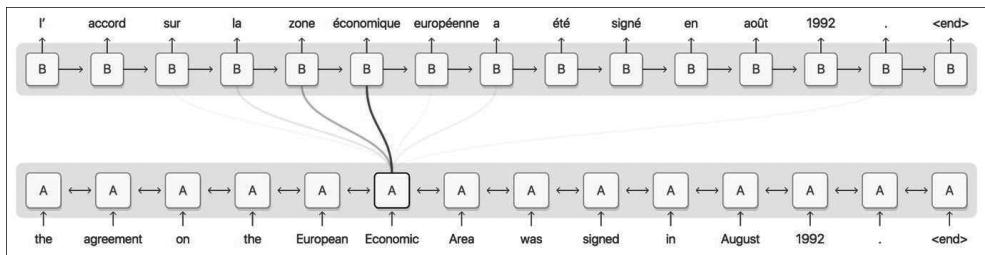


图 5-2 distill.pub 图解(图片来源: <https://distill.pub/2016/augmented-rnns/>)

长-短期记忆网络是为解决梯度消失问题而创建的 RNN 变体。梯度消失问题是由距离当前步骤较远的记忆单元所引起的,且某些记忆单元由于距离较远而导致其接受的权重较低。LSTM 是 RNN 的变体,其包含一种称为遗忘门(forget gate)的记忆单元,该组件可用于评估近期元素和旧元素对权重和偏差的影响,具体取决于序列中观察点的位置。

i 有关 LSTM 的详细信息请参阅 Sepp Hochreiter 和 Jürgen Schmidhuber 在 1997 年首次提出的 LSTM 架构。目前,LSTM 架构的实现已经进行了多次修改。关于 LSTM 的每个组件的工作原理的详细数学解释,建议参考 2015 年 8 月由 Christopher Olah 撰写的题为 *Understanding LSTM Networks* 的文章,详见: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>。

3. 对抗生成网络

对抗生成网络是由 Ian Goodfellow 及其蒙特利尔大学的同事于 2014 年发明的。GAN 认为,与其用一个神经网络优化权重和偏差以使其误差最小化,不如建立两个相互博弈的神经网络以达到同样的目的。

i 有关 GAN 的详细信息请参阅 Ian Goodfellow 等人于 2014 年 6 月 10 日在 arXiv 上发表的题为 *Generative Adversarial Networks* 的文章。详见: <https://arxiv.org/abs/1406.2661>。

GAN 有一个用于生成新数据(即“假”数据)的网络(即生成网络)和一个评估网络生成的数据的“真或假”可能性的网络(即辨别网络)。两种网络不断学习并互相博弈:一个网络学习如何更好地生成“假”数据;另一个网络学习如何区分得到的数据的真假。当评估生成数据的网络(即辨别网络)不再能够区分数据的“真假”时,两个网络将在每个 epoch 中都进行迭代,直至两者都收敛。

GAN 已经成功应用于数据具有明确拓扑结构的领域。GAN 最初实现的是生成物体、人脸和动物的合成图像,并且这些图像都与这些事物的真实图像相似,如图 5-3 所示。图像生成是 GAN 最常用的领域,但在研究论文中,GAN 在其他领域的应用也会偶尔出现。

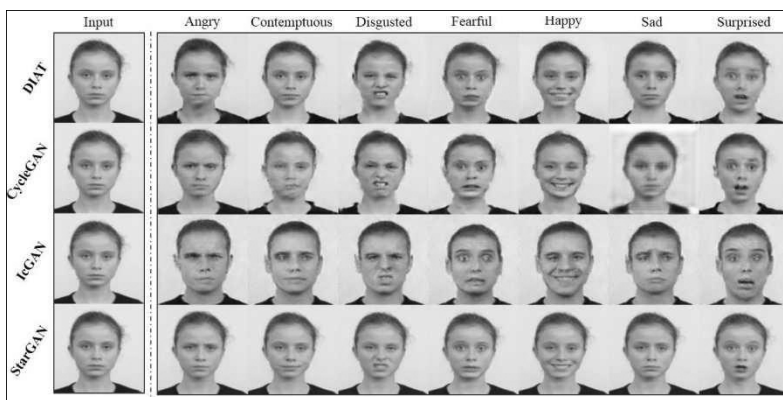


图 5-3 基于给定情感的不同 GAN 算法在改变人的面部表情方面的效果图

(图片来源:StarGANs Project, 详见: <https://github.com/yunjey/StarGAN>)

4. 深度强化学习

最初的深度强化学习(Deep Reinforcement Learning, DRL)体系结构是由位于英国的 Google 公司旗下的人工智能研究机构的 DeepMind 提出的。

DRL 网络的核心思想是：神经网络在本质上是无监督的，是从反复试验中学习的，并且只对激励函数进行优化。也就是说，不同于其他神经网络（它们使用有监督的方法优化预测的错误，而不是已知的正确方法），DRL 网络不知道解决问题的正确方法。神经网络被简单地赋予了一个系统的规则，然后它在每次正确执行一个函数后都会得到奖励。这个过程需要大量的迭代，最终由于它在许多任务中的卓越表现而使训练网络脱颖而出。

i 更多有关信息，请参阅在《自然》上由 Volodymyr Mnih 等人于 2015 年 2 月发表的题为 *Human-level control through deep reinforcement learning* 的文章，详情请见：<https://storage.googleapis.com/deepmind-media/dqn/DQNNaturePaper.pdf>。

深度强化学习模型在 DeepMind 创建了 AlphaGo（一个比专业玩家玩游戏玩得还好的系统）之后逐渐流行起来。DeepMind 还创建了 DQN 算法，可以学习如何完全依靠自己并以超人的水平玩电子游戏，如图 5-4 和表 5-1 所示。

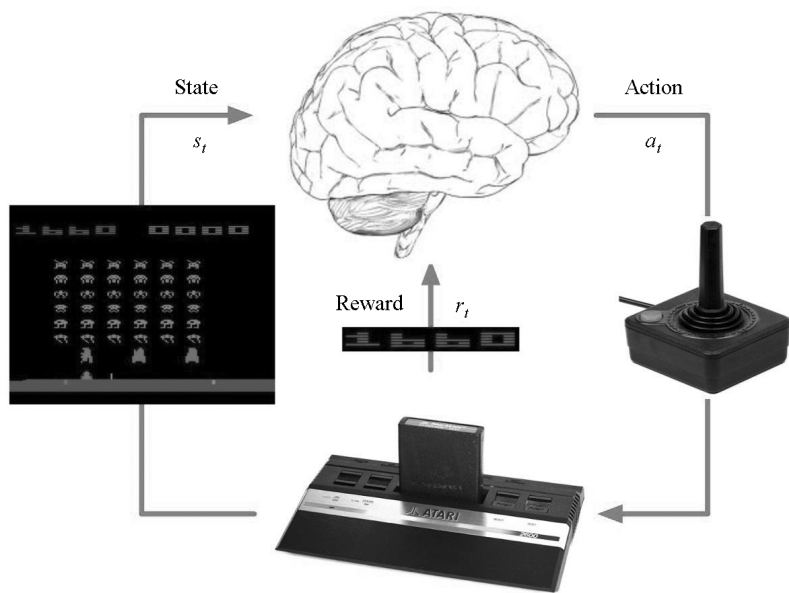


图 5-4 DQN 算法工作方式图

i 更多有关信息，请参阅 DeepMind 所创建的用来击败 Atari 游戏的 DQN 算法，该算法采用一种深度强化学习解决方案以不断增加其奖励。图片来源：<https://keon.io/deep-q-learning/>。

表 5-1 不同的神经网络体系结构在不同的应用领域都取得了成功，
采用的神经网络的体系结构通常与当前问题的结构有关

结 构	数 据 结 构	成 功 应 用
卷积神经网络(CNN)	网格状拓扑结构(即图像)	图像识别和分类
递归神经网络(RNN)和长短期记忆网络(LSTM)	序列数据(即时间序列数据)	语音识别、文本生成和翻译
生成对抗网络(GANs)	网格状拓扑结构(即图像)	图像生成
深度强化学习(DRL)	规则清晰、激励函数明确的系统	电子游戏和自动驾驶

5.1.2 数据标准化

在构建深度学习模型之前,还需要进行数据标准化。

数据标准化是机器学习系统中的一种常见做法,特别是在神经网络方面。研究人员提到,对于训练 RNN 和 LSTM 的神经网络来说,数据标准化是一项十分重要的技术,因为它减少了网络的训练时间并提高了网络的整体性能。

i 有关更多信息,请参阅 Sergey Ioffe 等人于 2015 年 3 月发表于 *arXiv* 上的题为 *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift* 的文章。详见: <https://arxiv.org/abs/1502.03167>。

应当根据不同数据和问题决定采用哪一种标准化技术,下面介绍一些常用的标准化技术。

1. Z-score 标准化^①

当数据呈正态分布(即高斯分布)时,可以将每个观测值之间的距离计算为与其平均值的标准差。

Z-score 标准化方法适用于含有超出取值范围的数据的离散数据,定义如下:

$$z_i = \frac{x_i - \mu}{\sigma}$$

其中, x_i 是第 i 个观测值, μ 是平均值, σ 是总体标准差。

^① 译者注:有的文献称之为零-均值规范化、标准差标准化。经过处理的数据均值为 0,标准差为 1,它描述的是“给定数据距离其均值有多少个标准差”,在均值上的数据会得到一个正的标准化分数,反之则会得到一个负的标准化分数。引自: https://blog.csdn.net/weixin_38706928/article/details/80329563。

i 有关更多信息,请参阅 Wikipedia 中题为 *Standard score* 的文章,详见: https://en.wikipedia.org/wiki/Standard_score。

2. Point-Relative 标准化

这种标准化方法是计算给定的观测值相对于序列的第一个观测值的差异,它对于确定与起点相关的趋势很有用。Point-Relative 标准化的定义如下。

$$n_i = \left(\frac{O_i}{O_0} \right) - 1$$

其中, O_i 是第 i 个观测值, O_0 是该序列的第一个观测值。

i Siraj Raval 在其名为 *How to Predict Stock Prices Easily Intro to Deep Learning* #7 的视频中介绍了相关内容,视频可在 YouTube 上获取: <https://www.youtube.com/watch?v=ftMq5ps503w>。

3. Min-Max 标准化^①

Min-Max 标准化方法计算的是给定观测值与序列中的最大值和最小值之间的距离,其在处理最大值和最小值不是异常值以及对于将来的预测很重要的序列时非常有用。

Min-Max 标准化的定义如下。

$$n_i = \frac{o_i - \min(\mathbf{O})}{\max(\mathbf{O}) - \min(\mathbf{O})}$$

其中, o_i 是第 i 个观测点, $\mathbf{O}$ 表示具有所有 $\mathbf{O}$ 值的向量,函数 $\min(\mathbf{O})$ 和 $\max(\mathbf{O})$ 分别表示该序列中的最小值和最大值。

本书将在 5.1.4 节的实例“探索比特币数据集以及为模型准备数据”中准备可用的比特币数据,以便在 LSTM 架构下使用。实例中包括选择必要变量、选择相关的时间段以及采用 Point-Relative 标准化技术。

5.1.3 构建您的问题

与研究人员相比,在开始一个新的深度学习项目时,实践人员往往花费更少的时间决定选择哪种体系结构。在开发这类系统时,最重要的考虑因素是获取能够正确表示给定问题的数据,其次是理解数据集的固有偏差和局限性。

^① 译者注:有的文献称之为最小-最大标准化方法,这是一种将原始数据线性化转换到 $[0,1]$ 的方法。

在开发深度学习系统时,请考虑以下问题,如图 5-5 所示。

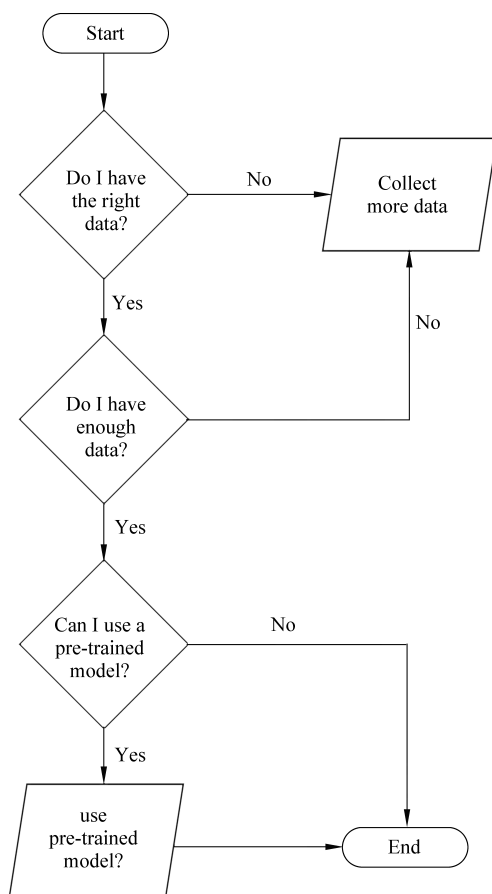


图 5-5 在深度学习项目开始时要做的关键反思问题的决策树

- **我有正确的数据吗?** 这是训练深度学习模型时面临的最大挑战。使用数学规则定义问题,使用精确的定义将问题划分为类别(分类问题)或连续值测度问题(回归问题)。那么,应该如何收集这些测度的数据呢?
- **我有足够多的数据吗?** 通常情况下,深度学习算法在大型数据集中的表现要比在较小的数据集中的表现好得多。需要了解训练高性能算法所需的数据量,当然这取决于尝试解决的问题类型,但目标还是尽可能多地收集数据。
- **我可以使用一个预先训练过的模型吗?** 如果您正在处理的问题属于一般应用程序的一部分(在同一领域中),那么请考虑优先使用训练好的模型。训练好的模型可以帮助您在解决问题的特殊模式方面领先一步,而不是让模型使用整个领域更通用的特征,可以从 TensorFlow 官方的 Github 库(<https://github.com/tensorflow/models>)入手。

在某些情况下,数据可能根本无法获得。根据具体的情况,可以使用一系列技术有效

地从输入数据中创建更多的数据,这一过程被称为数据增强(data augmentation),数据增强技术在处理图像识别问题时得到了成功应用。

i 这里有一篇很好的参考文章——*Classifying plankton with deep neural networks*, 详见 <http://benanne.github.io/2015/03/17/plankton.html>。为了增加该模型的训练样本数,文章的作者给出了一系列扩充少量图像数据的技术。

5.1.4 实例: 探索比特币数据集,为模型准备数据

下面使用一个最初从 CoinMarketCap 网站上检索到的公共数据集, CoinMarketCap 是一个跟踪不同加密货币的统计数据的网站,该数据集会同本章内容一起提供给读者,以供使用。

我们将继续使用 Jupyter Notebook 对数据集进行探索。Jupyter Notebook 通过 Web 浏览器提供 Python 会话,它允许用户以交互的方式处理数据。Jupyter Notebook 是探索数据集的流行工具,我们将在本书的所有实例中使用它。

打开终端,导航到本书配套文件目录的 Chapter_5/activity_3,并执行以下命令以启动 Jupyter Notebook 实例。

```
$ jupyter notebook
```

在浏览器中打开应用程序提供的 URL,应该能够在文件系统中看到一个带有许多目录的 Jupyter Notebook 页面,如图 5-6 所示。

```
(venv) x ~/Programs/book/lesson_2/activity_3 jupyter notebook
[I 00:25:49.252 NotebookApp] Serving notebooks from local directory: /Users/lcapelo/Programs/book/lesson_2/activity_3
[I 00:25:49.252 NotebookApp] 0 active kernels
[I 00:25:49.252 NotebookApp] The Jupyter Notebook is running at:
[I 00:25:49.252 NotebookApp] http://localhost:8888/?token=2a5f88a7e4eedadd758a342ba1310013610353560df95a32
[I 00:25:49.252 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 00:25:49.253 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
http://localhost:8888/?token=2a5f88a7e4eedadd758a342ba1310013610353560df95a32
[I 00:25:49.451 NotebookApp] Accepting one-time-token-authenticated connection from ::1
```

图 5-6 启动一个 Jupyter Notebook 实例后的终端图像。导航到浏览器中的 URL 显示,应该能够看到 Jupyter Notebook 的登录页面

导航到目录,单击文件 Activity_3_Exploring_Bitcoin_Dataset.ipynb。这是一个 Jupyter Notebook 文件,它将在一个新的浏览器选项卡中被打开,如图 5-7 和图 5-8 所示。应用程序将自动为用户启动一个新的 Python 交互式会话。



图 5-7 Jupyter Notebook 实例的登录页面

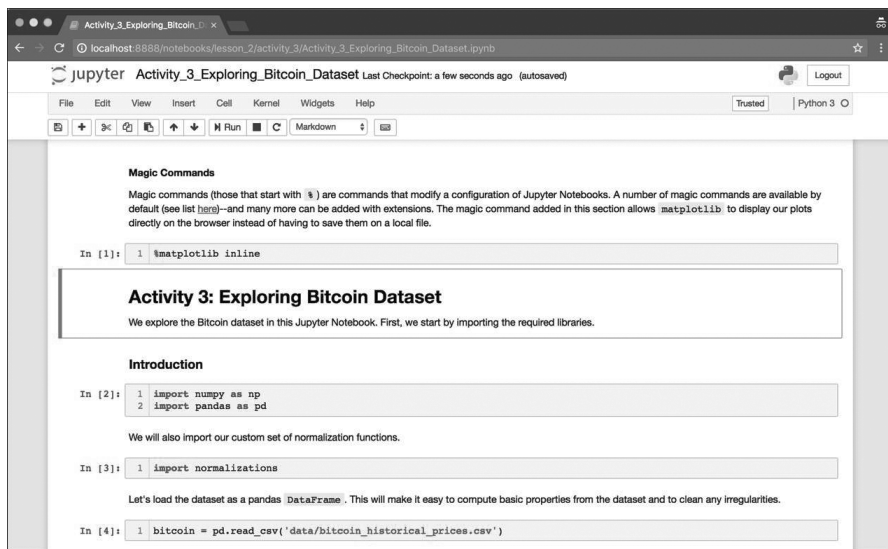


图 5-8 Notebook 中的 Activity_Exploring_Bitcoin_Dataset.ipynb 图。

现在可以与该 Notebook 互动, 并进行修改

打开 Jupyter Notebook 后, 即可开始探索本章提供的比特币数据。

数据集(data/bitcoin_historical_prices.csv)包含自 2013 年以来比特币价格的测度结果。最近的一次观察(observation)是 2017 年 11 月, 数据集来自每日更新的在线服务网站 CoinMarketCap。该数据集包含 8 个变量(如表 5-2 所示), 其中 2 个变量(date 和 iso_week)描述数据的时间段, 可以用作索引; 还有另外 6 个变量(open, high, low, close, volume 和 market_capitalization), 可以用来了解比特币的价格和其价值随时间的变化情况。

表 5-2 比特币历史价格数据集中的可用变量

变 量	描 述
date	观察日期
iso_week	给定年份的周数
open	一枚比特币的公开价值