享读 **深度学习第二课：个性化推荐**

2017年03月13日 7158次点击 30条评论

分享人：李钊 部门：深度学习实验室

分享理由：以深度学习为代表的人工智能技术正在悄无声息地进入推荐领域，不管是电商、电影还是新闻、Feed 流，有人的地方就有个性化。人们往往喜欢花 2 个小时看一部电影，却不愿意花 20 分钟去挑选一部电影，这就是推荐系统的意义。

在 Paddle 深度学习系列 Chat 的第一课中，Paddle 官方开发组的张睿卿同学通过介绍一些深度学习的应用场景，带领大家了解深度学习的基本原理和工作方式，我们先来简单回顾下。

“人工智能”并不是一个很新的概念，它其实已经有 60 岁了，它的发展经历了三起三落，像极了数学史上的“三次危机”。作为燃料的大数据和硬件（GPU）腾兴带来的并行运算，促成了深度学习在 2012 年左右的大爆发。深度学习有很多有趣的应用，比如，搭载 GoPro 的小车的“自动驾驶”可以视为一个回归问题，普通的照片可以模仿出著名艺术家画作的风格，在机器翻译、序列生成等领域也有所突破。此外，深度学习并不“完美”，还有很多理论基础问题等待我们去解决，比如说存在可解释性的局限：很多东西不能称为“方法”，只能称为“窍门”（trick），南大周志华教授将其比作“老中医看病”。

从去年年底开始，Paddle 社区将理论与实践结合，开始撰写一份深度学习教程，其中包括：新手入门、识别数字、图像分类、词向量、情感分析、文本序列标注、机器翻译、个性化推荐。这份教程的每一章都对应一个真实问题，从背景介绍到代码实践，带领大家完整地解决问题。

本次 Chat 的主题是个性化推荐。在系列教程个性化推荐一文中，我们介绍了推荐系统的背景和经典模型，并以电影推荐为例，使用 MovieLens 数据集和 PaddlePaddle 训练了一个神经网络模型。

什么是推荐系统

> 随着信息技术和互联网的发展，人们逐渐从信息匮乏的时代走入了信息过载（information overload）的时代 。在这个时代，无论是信息消费者还是信息生产者都遇到了很大的挑战：作为信息消费者，如何从大量信息中找到自己感兴趣的信息是一件非常困难的事情；作为信息生产者， 如何让自己生产的信息脱颖而出，受到广大用户的关注，也是一件非常困难的事情。推荐系统就是解决这一矛盾的重要工具。
>
> —— 项亮 《推荐系统实践》

乔布斯曾说，“消费者并不知道自己需要什么，直到我们拿出自己的产品，他们就发现，这是我要的东西”。同样，我们也可以说，信息爆炸的时代，面对琳琅满目的商品，用户很可能不知道自己真正喜欢什么，如果没有推荐系统，用户也许永远不

最热分享

FEX 技术周刊 - 2017/07/10

【Delta Club】国际科技公司动态追踪月报-6

【Delta Club】Weekly热点回顾（Jul 3-9)

最新评论

规划很清楚，讲话也很有逻辑~希望苏宁能...

赞一个，分析的很好

回复李鹏(lipeng08)：我是看到你的评论 ...

赞小天后！

很棒

赞

顶！

一分耕耘，一分收获。

辛苦啦

外卖确实不错

知道有更喜欢、更适合的商品没有浏览到。

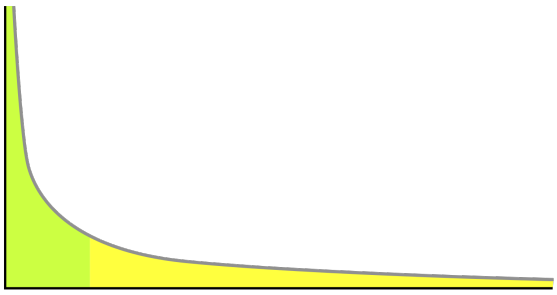
推荐系统和搜索引擎是人们获取信息的两种主要方法，与搜索引擎相比，推荐系统并不需要用户主动地寻找信息或商品，也不需要用户输入难以用简练文字描述的需求。但二者并不矛盾，在很多业务场景上推荐和搜索是相互结合的，比如说，搜索“周杰伦”时侧栏会推荐《听妈妈的话》。

从用户的角度讲，人们往往喜欢花 2 个小时看一部电影，却不愿意花 20 分钟去挑选一部电影；从企业的角度看，Data Science Central 编辑总监 Bill Vorhies 曾撰文[1]表示，“据估计，对亚马逊和 Netflix 这样的主要电商平台来说，个性化推荐的用户可能带来多达 10% 到 25% 的增量收入”，这就是推荐系统的意义。

长尾效应

长尾（The Long Tail）最初由《连线》的总编辑克里斯·安德森（Chris Anderson）于 2004 年提出，用来描述诸如亚马逊和 Netflix 之类的网站的商业和经济模式，指那些不受到重视的销量小但种类多的产品或服务，由于总量巨大，累积起来的总收益超过主流产品的现象。在互联网领域，长尾效应尤为显著[2]。

如下图所示，图中横轴表示数据类型，纵轴表示频率，大部分数据的频率都很低，但都是大于零的（图中右侧黄色部分），这就是长尾。比如，人们生活中常用的汉字其实并不多，但因频率较高，所以这些为数不多的汉字占据了左侧绿色区域，而绝大部分的汉字罕有使用，它们就属于长尾。



一个优秀的推荐系统不仅能推荐全局热点，更应该能够准确地理解**“长尾”需求**：通过挖掘某种用户群体的小众需求，将符合条件但并不热门的商品或信息推荐给用户。由于并非每个人的偏好都与主流完全一致，长尾数据的成功挖掘将带来远远高于平均的效益。

百度研究院的王益老师曾在《分布式机器学习系统》系列讲座上分享过一个真实的 case：用户搜索“红酒木瓜汤”，如果推荐系统能够理解出“丰胸”、“美容”、“减肥”等方面的语义，那点击（或购买）的几率将远远高于平均，推荐系统的任务就是将长尾需求和用户偏好挖掘出来并匹配。亚马逊高级副总裁 Steve Kessel 曾说“如果我有 10 万种书，哪怕一次仅卖掉一本，10 年后加起来它们的销售就会超过最新出版的《哈利·波特》！”说的其实也是这个道理。

传统的推荐方法

传统的推荐方法可以分为协同过滤推荐、基于内容过滤推荐和组合推荐，其中协同过滤的应用最为广泛，我们的教程中有更详细的介绍。

	优点	缺点
协同过滤推荐	个性化程度高	冷启动、稀疏问题
基于内容过滤推荐	简单	不能发现新商品

协同过滤推荐和基于内容过滤推荐各有优缺点，所以在工业界中往往采用模型的组合方式，克服各自的缺点，达到更好的效果。在刚刚结束不久的 AAAI-17 大会上，1999 年的一篇论文因发现了将协同过滤与基于内容过滤结合起来的有效方式，被评为经典论文提名奖（Honorable Mention）。

深度学习具有优秀的提取特征的能力，能够学习多层次的抽象特征表示，并对异质或跨域的内容信息进行学习，因此近年来在推荐系统上的应用和探索也渐渐增多。

基于深度学习的推荐系统

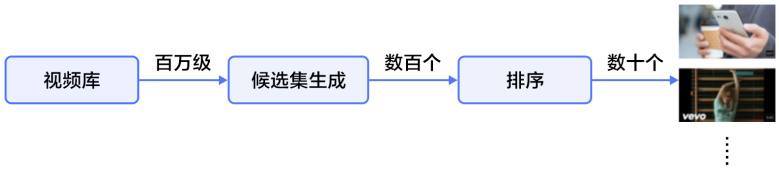
这一部分，我们会介绍 Google 提出的 YouTube 深度神经网络推荐模型和宽度&深度学习模型，以及我们使用 PaddlePaddle 实现的融合推荐模型。

YouTube 的深度神经网络推荐系统

经常上 YouTube 看视频的同学可能知道，它的首页视频**几乎全部**是个性化的，足以见得推荐系统对这个世界上最大的视频网站的重要性。

YouTube 的推荐算法系统经历过几次改动，其团队也发布了很多相关的论文。在 2016 年 9 月的 RecSys 会议（推荐系统领域顶级会议）上，Google 发布了 YouTube 的深度神经网络推荐模型[3]。

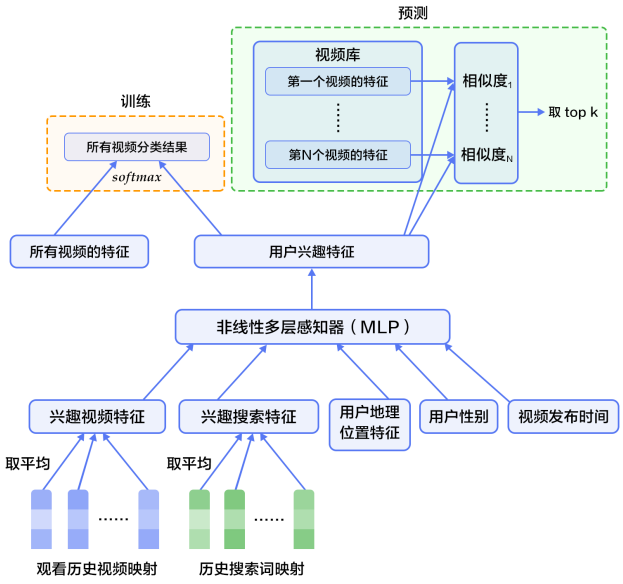
这个模型由两个神经网络组成：候选生成网络和排序网络。这样划分是一个常见的做法：为了节省计算资源，首先从大规模样本中召回候选集，降低数据规模，然后进行更精细的运算，得到 top k。



候选生成网络将推荐问题建模为一个类别数极大的多类分类问题（如下图所示），它首先将用户的历史信息（如观看历史、搜索历史）和其他特征拼接成向量，输入给非线形多层感知器（MLP）。

在训练阶段，将 MLP 的结果输入 Softmax 进行多分类，预测时计算用户的综合特征（MLP的输出）与所有视频的相似度，取得分较高的 k 个视频输入给排序网络。

这里 YouTube 团队特意介绍了“视频发布时间”（也可以称作 Example Age，样本年龄）这一特征，因为经过观察，用户更喜欢新发布的视频，哪怕有点和自己不相关，对于这样一个视频数目庞大的网站，新视频的推荐也是极其重要的。由于机器学习系统都是使用历史的行为数据来训练，这样就对过去存在一个隐式的偏差（bias），因此把 Example Age 特征加入模型后，可以发现模型结果和经验上的分布更相符。



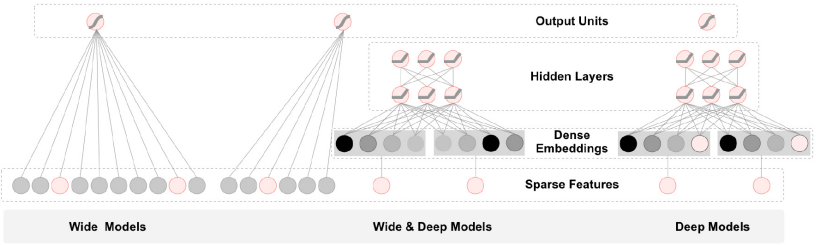
候选生成网络结构[4]

排序网络模型结构与候选生成网络类似，它添加了许多用于描述用户和视频相关性的更精细的特征，从而进行更细致的打分，比如用户很可能根据首页视频的缩略图去选择。此外，排序网络顶部使用加权逻辑回归（weighted logistic regression）进行训练，使用 e^x 作为测试阶段的激活函数。

Google 的 宽度&深度学习（Wide & Deep learning）

Google 在 2016 年 6 月发布了一篇关于“宽度&深度学习”的论文[5]，业内一些公司也在纷纷学习。这里的推荐场景是 Google Play 应用商店，但其实 Wide & Deep 的方法可以泛化应用在更广义的推荐场景上。

简单来说，人脑就是一个不断**记忆（memorization）**并且**归纳（generalization）**的过程。比如说人们通过记忆“麻雀会飞”和“鸽子会飞”，归纳出“有翅膀的动物就会飞”的结论。由此获得启发，将宽线性模型（用于记忆，下图左侧）和深度神经网络模型（用于归纳，下图右侧）结合，汲取各自优势形成了 Wide & Deep 模型用于推荐排序（下图中间），这是一个非常有启发的探索。

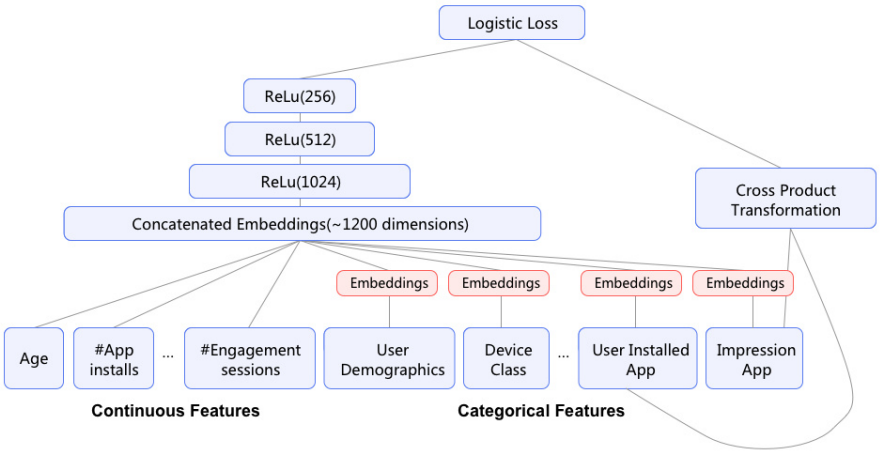


宽度&深度模型[6]

宽度模型的输入是用户安装应用和为用户展示（impression）的应用间的向量积（叉乘），模型通常训练 one-hot 编码后的二值特征（比如安装 netflix app 并展示了 pandora app 是 1，没有展示是 0），这种操作不会归纳出训练集中未出现的特征对。

基于 embedding 的深度模型可以探索出过去从未或很少出现的新的特征组合，提升了推荐商品的多样性。它可以添加小颗粒特征（比如安装了视频类应用，展示的是音乐类应用），同时也需要手动完成特征工程。高维稀疏的类别特征（如人口学特征和设备类别）映射为低纬稠密的向量后，与其他连续特征（用户年龄、应用安装数等）拼接在一起，输入 MLP 中，最后输入逻辑输出单元。

预测（服务）时，宽度&深度学习模型会将所有候选应用的分数从高到低排序后返回给用户。

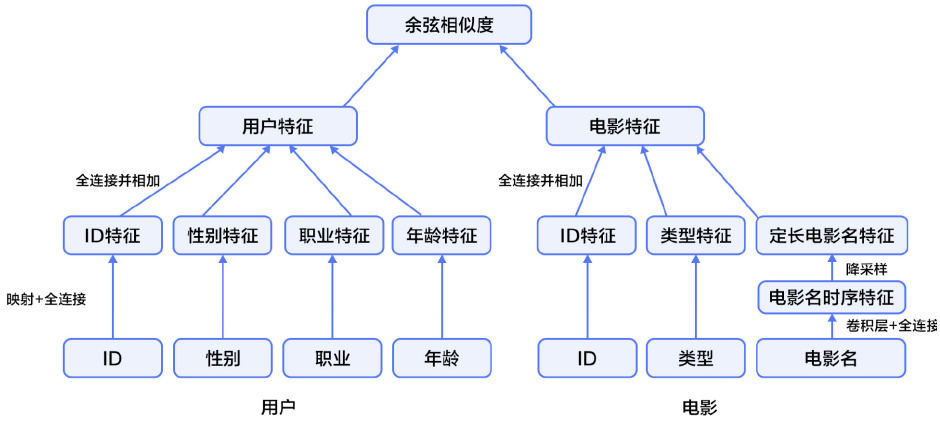


应用推荐中的宽度&深度模型[7]

尽管“ 宽度&深度学习”这一想法很简单，但经过试验，它显著提高了 Google Play 商店中应用的下载率，同时满足了训练和测试阶段的速度要求。值得一提的是，宽度&深度学习模型和集成（ensemble）学习并不是一回事，因为集成学习中的模型是分别独立训练的，互不干扰，只有在预测时才会联系在一起。

融合推荐模型

我们将使用 Paddle 实现电影推荐模型，数据集包含了 6,000 位用户对 4,000 部电影的 1,000,000 条评价（评分范围 1~5 分，均为整数），训练完成后，通过输入电影和用户的 ID，模型能够预测出该用户对该电影的评分，以代表喜好程度。这里只介绍主要的网络配置，完整版请见教程。



设置 batch size、网络初始学习率，使用 RMSProp 优化方法。

```
settings(batch_size=1600, learning_rate=1e-3, learning_method=RMSPropOptimizer())
```

构造用户、电影特征（以用户特征为例）

```
# 将用户ID，性别，职业，年龄四个属性分别映射到其特征隐层。
user_id_emb = embedding_layer(input=user_id, size=embsize)
user_id_hidden = fc_layer(input=user_id_emb, size=embsize)

gender_emb = embedding_layer(input=gender, size=embsize)
gender_hidden = fc_layer(input=gender_emb, size=embsize)

age_emb = embedding_layer(input=age, size=embsize)
age_hidden = fc_layer(input=age_emb, size=embsize)

occup_emb = embedding_layer(input=occupation, size=embsize)
occup_hidden = fc_layer(input=occup_emb, size=embsize)

# 将这四个属性分别全连接并相加形成用户特征的最终表示。
user_feature = fc_layer(
           input=[user_id_hidden, gender_hidden, age_hidden, occup_hidden],
           size=embsize)
```

计算余弦相似度，定义损失函数和网络输出。

```
similarity = cos_sim(a=movie_feature, b=user_feature, scale=2)


# 训练时，采用regression_cost作为损失函数计算回归误差代价，并作为网络的输出。
# 预测时，网络的输出即为余弦相似度。
if not is_predict:
    lbl=data_layer('rating', size=1)
    cost=regression_cost(input=similarity, label=lbl)
    outputs(cost)
else:
    outputs(similarity)
```

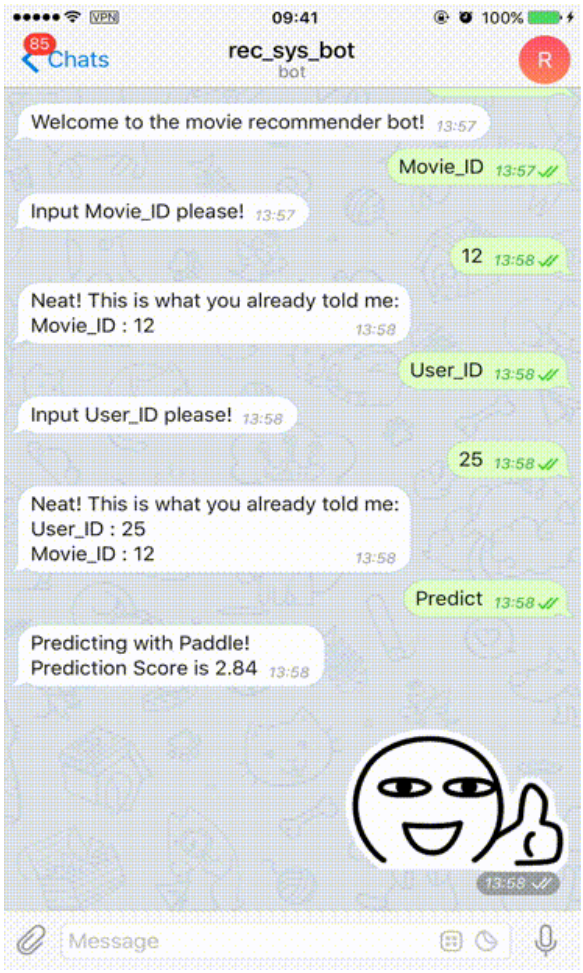
训练完成后，我们可以通过 `./evaluate.py log.txt` 评估模型，找出效果最好的模型轮数。接下来，我们搭建一个简单的 ChatBot 完成电影推荐的预测，作为融合推荐模型的应用。

融合推荐模型的 ChatBot 应用

近些年涌现出一大批聊天机器人和智能家庭设备，它们几乎全部支持个性化，比如“识别不同的人”，“根据不同人的喜欢推荐不同的内容”。Facebook 创始人扎克伯格使用多种 AI 技术为自己家里构建了一个自动控制系统，命名为 Jarvis，它能够根据家庭成员的喜好播放不同风格的音乐。所以，基于聊天机器人的个性化服务是未来的趋势。

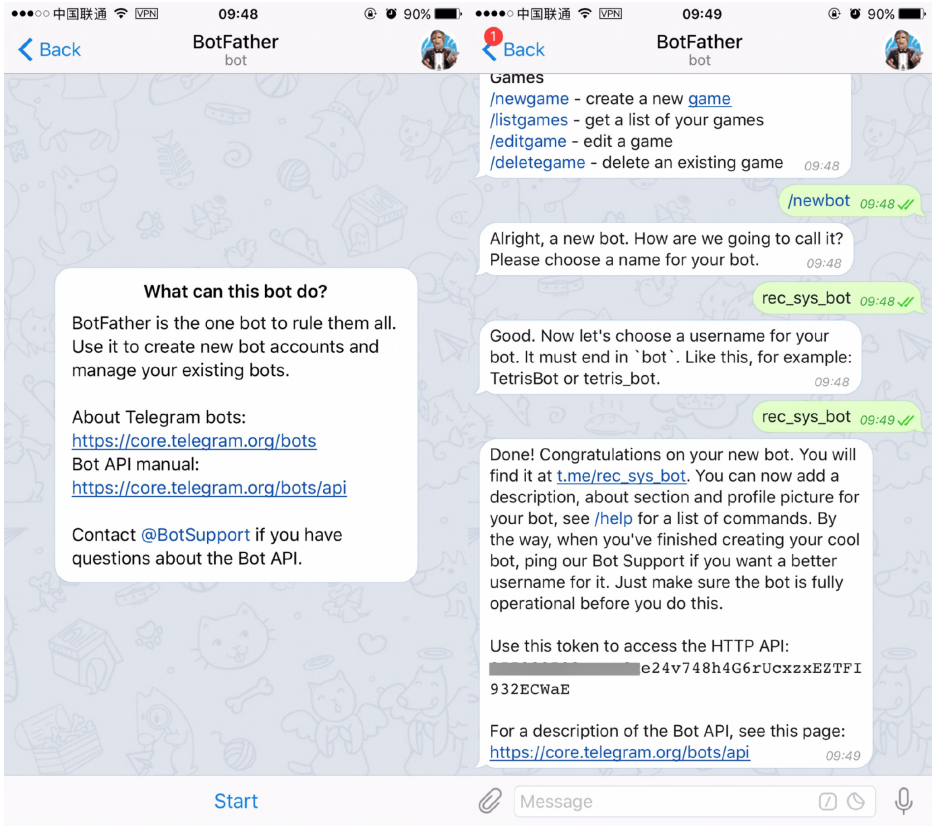
Bot 的开发**非常简单**，我们借助 Telegram 来完成这个任务。Telegram 是一款开源的即时通讯软件（类似微信、WhatsApp 等），它的机器人平台（Telegram Bot Platform）极大地丰富了生态，比如可以使用 Bot SSH 登录 VPS 、接收 RSS 订阅新闻或博客、下载 YouTube 视频、接收微信消息甚至是玩游戏等等。由于 Bot 是面向 API 的，我们可以开发某个 Workflow（比如 IFTTT）完成一系列的任务，有人为其创造了一个新名词，叫“r2r - robots 2 robots”。

由于其服务在中国的网络环境下并不容易访问，这里我推荐使用 proxychains 运行 Python 文件或 IPython 交互环境，当然也可以直接搭建在国外的 VPS 上。基于融合推荐模型的 ChatBot 最终效果如图：



1. 创建 Bot 并安装依赖

首先需要找官方的机器人老爹 @BotFather 发送 `/newbot` 命令申请创建，设置 bot 的基本信息后会得到一串 Token，帮助访问 HTTP API，这是 Bot 的唯一标识，不能泄露出去。



在 GitHub 上，很多热心的开发者使用不同的语言封装了原生 API，让开发变得更加容易。我们使用 Python 完成开发，因此首先安装依赖 `pip install python-telegram-bot --upgrade`。

2. 接入 Paddle 预测文件

变量 `MODEL_PATH` 是模型评估 `./evaluate.py log.txt` 的结果，函数`cal_with_paddle` 实际就是教程中 `prediction.py` 的功能，输入是电影和用户的 ID，输出预测的评分。

Paddle 社区即将发布新版 API，数据读入、训练、预测的过程将变得更加简洁，因此你完全可以不关心这里究竟做了什么。

```
from py_paddle import swig_paddle, DataProviderConverter
from common_utils import *
from paddle.trainer.config_parser import parse_config
try:
    import cPickle as pickle
except ImportError:
    import pickle

# 模型路径
MODEL_PATH = 'output/pass-00004/'
def cal_with_paddle(movie_id, user_id):
    # 加载参数
    swig_paddle.initPaddle('--use_gpu=0')
    conf = parse_config("trainer_config.py", "is_predict=1")
    network = swig_paddle.GradientMachine.createFromConfigProto(
        conf.model_config)
    assert isinstance(network, swig_paddle.GradientMachine)
    network.loadParameters(MODEL_PATH)
    # 读入数据并预测
    with open('./data/meta.bin', 'rb') as f:
        meta = pickle.load(f)
        headers = [h[1] for h in meta_to_header(meta, 'movie')]
        headers.extend([h[1] for h in meta_to_header(meta, 'user')])
        cvt = DataProviderConverter(headers)
        movie_meta = meta['movie'][movie_id]
        user_meta = meta['user'][user_id]
        data = [movie_id - 1]
        data.extend(movie_meta)
        data.append(user_id - 1)
        data.extend(user_meta)
        return '%.2f' % (network.forwardTest(cvt.convert([data]))[0]['value'][0][0] + 3)
```

3. 变量声明与函数定义

替换 `TOKEN` 为实际申请的字符串，定义按键界面，和交互函数。

```
from telegram import ReplyKeyboardMarkup
from telegram.ext import (Updater, CommandHandler, MessageHandler, Filters, RegexHandler,



# 两种交互方式，分别是按键选择回复和输入文本回复
CHOOSING, TYPING_REPLY = range(2)
TOKEN ='123456789:AAG6xe24v748h4G6rUcxzxEZTFI932ECWaE'

# 选择回复界面的三个按键，分别是输入电影ID、用户ID和预测
reply_keyboard = [['Movie_ID', 'User_ID'],
                  ['Predict']]
markup = ReplyKeyboardMarkup(reply_keyboard, one_time_keyboard=True)

# 定义输出格式
def facts_to_str(user_data):
    facts = list()
    for key, value in user_data.items():
        facts.append('%s : %s' % (key, value))
    return "\n".join(facts).join(['\n', '\n'])

# `/start` 命令
def start(bot, update):
    update.message.reply_text(
        "Welcome to the movie recommender bot!",
        reply_markup=markup)
    return CHOOSING

# 记录按键（key）并要求用户输入对应文本(value)
def regular_choice(bot, update, user_data):
    text = update.message.text
    user_data['choice'] = text
    update.message.reply_text('Input %s please!' % text)
    return TYPING_REPLY

# 记录文本(value)
def received_information(bot, update, user_data):
    user_data[user_data['choice']] = update.message.text
    del user_data['choice']
    update.message.reply_text("Neat! This is what you already told me:"

                              % facts_to_str(user_data),
                              reply_markup=markup)
    return CHOOSING

# 调用Paddle预测函数并输出结果
def predict(bot, update, user_data):
    if 'choice' in user_data:
        del user_data['choice']
    score = cal_with_paddle(int(user_data['Movie_ID']), int(user_data['User_ID']))
    update.message.reply_text("Predicting with Paddle!\n"
                              "Prediction Score is %s" % score)
    user_data.clear()
    return ConversationHandler.END
```

4. 开始运行

```
updater = Updater(TOKEN)
dp = updater.dispatcher

# 定义对话handler
conv_handler = ConversationHandler(
    # 以`/start`命令作为入口
    entry_points=[CommandHandler('start', start)],
    # 定义交互方式，支持正则匹配
    states={
        CHOOSING: [RegexHandler('^User_ID|Movie_ID$',

                                pass_user_data=True),
                ],
        TYPING_REPLY: [MessageHandler(Filters.text,
                                  received_information,                                    pa

                    ],
    },

    # 预测
    fallbacks=[RegexHandler('^Predict$', predict, pass_user_data=True)] )

    # 添加对话handler
    dp.add_handler(conv_handler)

    # 开始运行
    updater.start_polling()
```

至此，我们已经完成了 ChatBot 推荐模型的基本功能。python-telegram-bot repo中还有更丰富的功能值得探索，此外我们还可以接入云服务的 API，例如使用 Google Cloud Speech API 完成语音转文字的功能。

总结

近些年来，深度学习已经极大地推进了图像处理、语音识别、NLP 等领域的发展与进步，而在推荐系统上面的应用还处于早期阶段，同时也意味着有很大的发展空间。此外，深度学习正在为医学、生物信息学、逻辑推理、量化投资甚至围棋等领域带来新的启发与思考。我曾与学校的神经科学研究所合作，使用深度学习技术来分析食蟹猴基因特征，预测 microRNA 的碱基序列，获得了不错的效果，而最基本的神经网络结构也是从大脑的生物机理获得的启发，这形成了推动学科进步的良性循环。

2016 年的最后一天，罗振宇在他的“跨年演讲”中提到，“人工智能不是人的延伸，它是人的替代”；英伟达 CEO 黄仁勋在《智能工业革命》中认为：“继蒸汽机（发明）、大规模生产以及自动化之后，AI 技术将引发第四次工业革命”；周志华教授在采访中说，“2017 年，机器学习技术将在更多行业带来更大价值”。各个行业的人们都在关注和见证着 AI 的发展，与此同时，很多工程师和社区（如 Paddle）正在努力着降低学习和应用的门槛。

我们有幸亲身经历了这次发展的浪潮，但仍需清醒地意识到其实还有很漫长的路等待人们的探索，我们期待更多如GAN（生成对抗网络） 一样的新思想的爆发，这需要我们见素抱朴，不忘初心。

感谢

感谢订阅本次 Chat，个性化推荐这一章节的网络结构其实很简单，更多的知识和内容，还请关注该系列的后续分享。

大家熟知的许多任务，如：机器翻译，看图说话，为你写诗，对话机器人，标题党改写等等，背后都有着共同的模型。下一课我们将会介绍这些任务背后的深度神经网络模型，一起进入自然语言处理任务中一个非常有意思的问题：自动文本生成。

我们将在下一课介绍自然语言处理任务中的重要积木：循环神经网络。围绕循环神经网络，我们会一起讨论，如何对抗梯度消失和梯度爆炸，为什么需要深度循环神经网络，如何有效地训练深度循环神经网络。在此基础上，我们会继续开发本课中的对话机器人，引入神经图灵机的概念，介绍去年最火的技术之一：“注意力机制”，利用已有的积木，让循环神经网络从数据中学习，自动生成回复与用户进行有趣地交互。最后，我们会一起讨论现有技术面临的挑战，探讨一些加速文本生成任务的技术，期待大家的参与。

Paddle 不仅属于百度，更属于开源社区，我们希望对深度学习感兴趣的研究人员、工程师和开源爱好者能够加入PaddlePaddle Tech Writer，撰写您所擅长的深度学习教程或设计有趣的示例，让更多的人感受到深度学习的魅力。如果您在使用 Paddle 过程中遇到任何问题，都可以去 GitHub 发起 Issue，社区的小伙伴们将在第一时间为您解答，希望 Paddle 与您共同成长。

参考资料

1.http://www.datasciencecentral.com/profiles/blogs/understanding-and-selecting-recommenders-1

2.https://zh.wikipedia.org/wiki/%E9%95%BF%E5%B0%BE

3.https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/45530.pdf

4.引自论文[3]中图3

5.https://arxiv.org/abs/1606.07792

6.引自论文[5]中图1

7.引自论文[5]中图4

Chat实录

2017年3月9日，周四晚上8点30分，PaddlePaddle 官方开源社区成员李钊带来了主题为“深度学习第二课：个性化推荐”的交流。以下是主持人小冰整理的问答实录，记录了老师和读者问答的精彩时刻。

问：看到你在生物信息学上使用深度学习技术，能分享一下深度学习在生物信息学、疾病预测等方面的应用吗？或者你们探索的经历？

答：我们当初研究的是一种非编码 RNA，叫 microRNA，它对调控基因表达等方面有重要作用，一个 miRNA 可以调控几十个基因的表达，生物体内有 98% 的转录产物都是非编码 RNA。我们当初探索的经历还是蛮辛苦的，因为我们学校没有专门研究交叉学科的实验室，只是两个学院老师的合作。所以当时我从高中生物开始恶补了很多，和我合作的神经科学生物所的博士学长从最基础的 Linux命令开始学习，这个过程需要我们两个人不断互相沟通和互相学习的。我们最开始是使用一些软件来预测，渐渐发现这些软件时间太久了，没人维护，虽然论文引用数很高，但用不起来，而且我们对miRNA的整个预测过程并没有一个非常清晰的认识，调研了好久才发现在预测miRNA序列之前要先预测出它的基因转录产物剪切后的前体（pre-miRNA）序列，所以这也算是一个不断探索的过程吧。最后才尝试自己构建模型，根据论文的思路，但那时其实相关的实现并不丰富，只能算是一个启发式的探索吧，但起码有了结果。预测出的结果需要做实验验证或者拿到外边去做基因测序，花钱又会花时间……后来随着我来到百度，这个项目就交给其他人跟进了。

问：想问下本科生怎么入门呀，如果入门之后，在哪能找到提高自己的方法，比如阅读一些实例代码，项目什么的还是其他更好的方法？还有，如果想在这方面找到实习的机会，需要做哪些准备呢？

答：我先把这个问题拆分成三个问题：

a. 如何入门。

感觉这已经是日经话题了，之前在一些论坛和社区也回答过类似的问题，现在尝试详细地梳理一下，我推荐的学习路线是：

Python 的基本编程能力。Python 可是说是目前数据科学最流行的语言了，因此如果不会 Python的话一定要先学习一下。学习一门新语言，可以推荐两个比较高效的方式。第一个是 Learn X in Y minutes（https://learnxinyminutes.com/），它给出了一些语言特性和基本用法，因为编程语言之间本身就有一些想通或相似的地方。比如 C和Go。第二个方法叫『koans』，它将知识点编写成测试代码，要用户类似于闯关一样地通过测试，也很有意思。

学习机器学习知识。现在网络资源这么丰富，足不出户就可以学习到名校课程，因此『学习能力』真的非常重要。技能可能会过时，但学习能力不会。周志华教授的西瓜书《机器学习》和 Andrew Ng 在 Coursera 上的 Machine Learning 课程，都非常适合新手。这两个可以同时进行学习。之后应该会对线性模型、LR、SVM、RF 等算法，过拟合、正则化等概念有所了解。同时可以接触一些 Python 科学计算的包，比如 Numpy、Pandas、sklearn，包括 Ipython、Jupyter Notebook 等工具。俗说话，『工欲善其事，必先利其器』哈。

深度学习知识。Udacity 上有一门 Google 工程师主讲的 Deep Learning 课程，非常简短和形象。这时能接触到 CNN、RNN、Word2Vec 等。此外斯坦福 CS231n（深度学习与计算机视觉） 可以做进一步的补充。这时，你应该有比较足够的知识储备了，如果愿意可以继续学习公开课，也可以去跑跑 Paddle 的教程。

b. 入门之后的提高方法。

很多人可能看了很多课程，也读了不少书，之后却不知道要做什么。我认为进入或接触一个新领域需要两个方面的支持：兴趣和需求。前者可能是推动你去入门学习的动力，就像刚才说的那些学习路线。而后者就可能是进一步提高自己的动力了，不知道如何解决问题，因为根本没有遇到问题。最重要的是一定要知道自己想做什么。如果仅仅是作为兴趣爱好，完全可以『花 20% 的时间了解一个新领域 80% 的知识』，去学习一下开源的 interesting demo。比如别人写了个汪峰歌词生成器，你能不能用另一种方式实现一个赵雷歌词生成器。如果是学生，可以参与一下实验室的项目，读一些论文，因为那里有真正的应用场景，有数据，还有学长能够指点你。此外还可以参加 Kaggle 的比赛，因为同样作为数据挖掘类的比赛社区，Kaggle 比国内的一些类似比赛社区的氛围要更好一些，因为很多人会把他们的实现开源出来，利于新手的学习。

c. 实习的准备。

简单来说，作为学生来讲，计算机的基础知识是非常重要的，有些公司会要求写代码，因此熟悉常用数据结构算法也非常有必要。然后就是竞赛获奖、实习经历、实验室项目、开源贡献、个人技术博客都可以留下好印象。除此之外，公司很多工作需要联合其他团队的人共同沟通完成，所以有时也会考察你的沟通能力、领导能力、团队合作能力等等『软技能』。总之，如果能证明读书时没有浪费时间，最好有产出证明的话，拿到offer应该不困难。

问，具体应该怎么将特征向量化呢?

答：简单来说就是将特征映射到维度更低的实数空间上，这样可以表达更多的内容，比如可以用来计算距离以代表二者的相似度。在我们教程中有一章非常细致地介绍了词向量（http://book.paddlepaddle.org/word2vec/），之后应该也会有作者来进行Chat。

问：深度学习将在生命科学领域发挥重要作用。如何利用深度学习提高基因功能预测的精准度？尤其是对于数量庞大功能模糊的非编码RNA的预测亟待解决，希望能有机会共同探讨这些问题。

答：非编码 RNA 的预测确实有很大的发展空间，尤其是在 Deep Learning 的环境下。Paddle 是计算平台，原则上，如果能把基因预测过程抽象成为一个可以用神经网络表达的数学模型，那么就可以完成这样的训练。我参加过一些社区的活动，其中有一些创业公司就是做类似这种基因 AI 或医疗 AI 的人，所以这也是一个很值得期待的领域，如果有机会真的可以互相学习一下。

问：这个机器人我以为是从语义理解的角度来说的机器人呢，没想到只是一个接口。更希望看到知识图谱、query分析的角度的机器人的内容。

答：这篇文章中的机器人只是作为个性化推荐模型的应用场景，因为很多人训练好模型后不知道在哪能应用，你可

以在此基础上继续开发，因为现在的功能都很简单。我自己开发的 Bot （和深度学习无关）现在有 1000 多个用户，就是一个很简单的收录群组的机器人，因为确实存在着这个需求。我本身也是一个 Hackathon 爱好者，渐渐总结出来的经验就是产品的应用场景比技术能力更加重要，比如说如果有一项特别牛的技术，但是没有好的落地点，没法进入人们生活，那就很尴尬了，重要的是能用这项技术改变人们的生活。对于和语义分析有关的 ChatBot 模型，请继续关注这个系列，会有更专业的人来分享！

问：文章中说深度学习崛起是在2012年，但是很多媒体说2016年才是深度学习元年，那么到底是哪一年呢？

答：这是一个很有趣的问题。2012 年 Hinton 教授的团队在 ImageNet 上使用深度学习拿下了图像分类比赛的冠军，准确率高出第二名 10% 以上，让学术界和工业界都为之震动。而有的媒体认为 2016 年是元年，应该是从应用的角度出发，因为这一年发生了很多大事，使深度学习或者说是人工智能走出学术圈子进入大众视野。最受公众瞩目也是影响力最大的事件就是 AlphaGo 与世界冠军李世石的人机大战，包括能将照片转化成艺术家作品风格的 App Prisma、Google 的神经网络翻译系统等等一系列事件，真正让大众接触到了人工智能和深度学习，尽管可能只有个模糊的概念。

问：使用深度学习要使用大量训练数据吧？如何解决数据稀疏问题？ 个性化推荐中冷启动问题一般怎么解决？

答：embedding可以将高维稀疏的向量映射为低维稠密的向量，这个在词向量一节中有具体的介绍，刚才也提到过了。冷启动有很多可以缓解的方法。比如，新下载了一个 APP，可能要求你输入个人信息，和喜欢的话题。完善的资料越多，系统能获取的信息越多。另外这个也可以应用在反作弊中，如果用户对一个社区产生了归属感，那么他就会去装扮和丰富自己的社区属性。比如小时候玩 QQ 空间，搞的花里胡哨的。。这可能会让他被识别为垃圾用户的几率降低。在 book 中也提到了 Deep Learning 在某些情况下可以缓解特定的冷启动问题的论文，可以去了解一下。

问：输入降维size\Batch size\learn rate，隐层神经元个数和层数设置怎么选择和调节，有没有好的方法和工具，一般使用多服务器还是以服务器多显卡的，500层以上网络几个显卡比较够用。

答：网络的配置情况还是具体问题具体分析吧，这个貌似没有一个通用或者公认的效果好的方法。你多了一层效果好，我少了一层效果好，这应该是比较常见的事情。但是可以参考论文的实现。比如 YouTube 那个论文里，就有对模型深度的探索，它分析了网络层数和 embedding size 对模型的影响。就我个人浅显的经验而言，我更喜欢在特征工程上下功夫。调整网络的话，还是需要先分析，比如是过拟合还是欠拟合了，可以通过观察学习曲线做出针对性的调整。最后可以考虑一些 trick 什么的。。前几天有一个文章讲了 18 个trick，这些都可以参考一下。工具倒是可以推荐一下，因为我在做 Kaggle 的时候接触过一点。TPOT（https://github.com/rhiever/tpot） 是一个基于基因编程（genetic programming）的项目，能够自动完成特征选取、预处理、模型选择和参数优化，最终输出效果最好的 pipeline。当然，计算量也是很大的。。适用于一些机器学习算法，貌似不支持神经网络。因为这样的话计算量更大了。此外可以使用一些训练好的模型。比如Caffe Model Zoo。论文《Large-Scale Evolution of Image Classifiers》提出了自动选择神经网络的进化算法，我简单看了一下有点像遗传算法，都是从生物进化论获得的启发，这也是一个非常有意思和启发的探索。500 层网络的情况，建议去社区求助一下，可能会有类似情况的用户。

问：1）用户id，电影id也可以作为特征吗？特征不必须是随机变量吗？2）如果可以作为特征.又比如用户id有几百万或是更多，又该怎么编码？

答：融合推荐模型中的ID是用户和电影的唯一标识，比如一个人对多部电影做出了评价，如果不使用的话怎么表达出对这些电影评价的用户是同一个人呢。ID 可以通过 embedding 映射，这个在 book 中就有介绍。

问：1）智能推荐需要依赖多大量级的数据才能做到很好的效果呢？2）学习该智能推荐和机器学习对硬件有什么特别的要求吗？比如，运算要求比较高，还是存储方面高？ 3）有java版本的api可以使用吗？4）目前业界也有好多智能推荐的东西，比如电商，京东，淘宝，网易云音乐，推荐的东西质量也很一般，这个和大数据量级有关还是 和算法有关呢？

答：1）『智能』和『很好』，都是比较模糊的概念吧。能依赖多少取决于有多少数据，现在有很多开源框架，但开源的数据很少，这些其实都是财富。

2）如果单纯学习的话，并不一定需要很强大的配置，如果需要，GPU应该就可以了，有需求上云平台也可以。对于新手而言，与其琢磨一个工作站，倒不如关注一下模型论文和理论基础。否则有点本末倒置了。这有点像很多同

学搭建个人博客，买了域名、VPS，搭建了Nginx，CDN、CMS、主题、评论系统、反垃圾统统折腾了一通。最后一篇博客也没写。

3）目前Paddle没有Java的API，以后可能会有。如果非要使用Java开发的话，可以尝试一下deeplearning4j（https://deeplearning4j.org/）。

4）个性化可以达到『千人千面』的目的，但毕竟众口难调。这些平台的数据应该都很丰富，谁让中国这么多人呢。。算法是重要的一点，但肯定不是全部，算法要和具体的业务结合，不同场景的推荐策略也不太相同。此外，对业务和商业的理解也非常重要，能不能形成一个良好的平台生态，也会在一定程度上影响推荐的效果。

问：有没step by step入门级进阶案例演示？怎样制定机器学习路线？怎么去攻克高门槛的算法？

答：刚才上面回答的入门路线够 step by step 了吧，重要的还是自己能不能坚持下来。『高门槛』是怎样定义的呢，学习路线没走完的话，把基础知识理解了，就能收获很多。就像大数据的驱动下，并不是多么高深的算法才能取得好的效果了，有时一个lr就能满足业务需求。

问：我想问的是，一个管理软件产品，在记录了一些用户行为数据之后，怎么实现机器学习，实现对系统用户的提示、预警、及推荐等等，甚至给用户提供一些决策建议？

答：这里听起来貌似有几个问题。

首先对自己产品的定义是怎样呢，如果只是一个普通的管理产品，那么就没有必要走AI路线。如果定义为一个人工智能类产品，能否进行一下市场调研呢，看看市场上这类产品的哪些地方，还可以做的更好。

记录的用户行为，是否足够用来机器学习分析呢。我在学校接触过一个小项目，是对一个公司的员工的忠诚度进行预估，但是数据就一个Excel表，所以这种分析好像也没啥意义。我觉得采用合适的数据科学的分析方法是可行的，具体还是要和产品的应用场景结合，来抽象或提取出一个数学模型。

问：现在实时推荐的解决方案是什么有来源出来吗，方案有在线上应用实施吗，有没有实战的case分享呢？

答：这个我并不是很有经验，因为严格意义上讲，我之前参与的算是『个性化推荐』，还不能称之为是一个『推荐系统』。但是据我了解，业内很多的推荐都是在学习那个Wide&Deep模型，比如小米和美团。因为参加过一些Meetup，也会去请教一些，尤其是做这次Chat。工业界的线上应用，感觉还是应该多多交交朋友，参加社区分享，去具体了解一下。

问：1）deep and wide有点像highway network。wide那一部分被直接连到最后，deep那一部分做残差。目前来说哪部分特征用于wide哪部分特征用于deep是有一个什么标准来决定的呢？有没有可能加个gate来自动决定特征怎么分？2）wide用于“记忆”，deep用于“归纳”。这该怎么理解？wide是原始特征直接用于输出，deep可以提取抽象化的特征，关键是‘记忆’不太好理解。

答：这个论文的特征选择我在文章简单提到了，具体可以看下论文。如果Wide部分不好理解，Google特意写了一个博客介绍，我觉得那个讲的应该比我讲的更透彻，地址是 https://research.googleblog.com/2016/06/wide-deep-learning-better-together-with.html 。

问：候选生成网络结构能详解一下吗？

是指网络的结构吗？这个我们的文章中有一个模型的结构图。（是的。）我们对图进行了重构，也可以阅读论文的原图看看。很多内容还是推荐去读读论文原文，每次读都有不一样的收获。

原文链接：http://gitbook.cn/m/mazi/article/58a6de96f919c2152af25d1f

百度Family

主办：百度内部沟通委员会

内部沟通委员会成员：陆奇、Jennifer、张亚勤、向海龙、朱光、刘辉、梁志祥、孙云丰、王路、王海峰、熊赟

协作：BP&IT、HR、行政部、内部沟通部、财务部、内审部

协调：内部沟通部 负责人：陈琦 运营编辑：郭伟 王瑶瑶