



进阶研究：集成学习和深度学习. 第31讲



渔阳 2019-03-28 19:59

字数 7,903

阅读需 20分钟

“欢迎来到量化小学”

▲ 加入[“量化小学”校友圈儿](#)提问交流

投资全球更要投资自己

上一篇:

因子研究之机器学习

来自特辑

量化小学



我的



详细内容请在wifi环境下观看视频

<本期课程8454字, 视频31分32秒, 请合理安排学习时间>

正文阅读

大家好, 欢迎来到量化小学。

上次我们讲了机器学习的基础知识和一些基本方法, 这次我们就来谈一些进阶的话题。

先是集成学习, 然后是深度学习, 最后我们对于机器学习在量化研究中应用的一些要点做一个总结。

集成学习概论

先来看一看集成学习。它的**基本思想**其实挺简单的, 就是将一些弱学习器组合起来, 获得精度更高的强学习器。弱学习器就是我们前面讲到的决策树或者逻辑回归等等。

我们来看一看下面这张图, 这是关于弱学习器怎么样组合获得强学习器的一个著名的一个示例图。



解放你的投资动手能力

最近更新

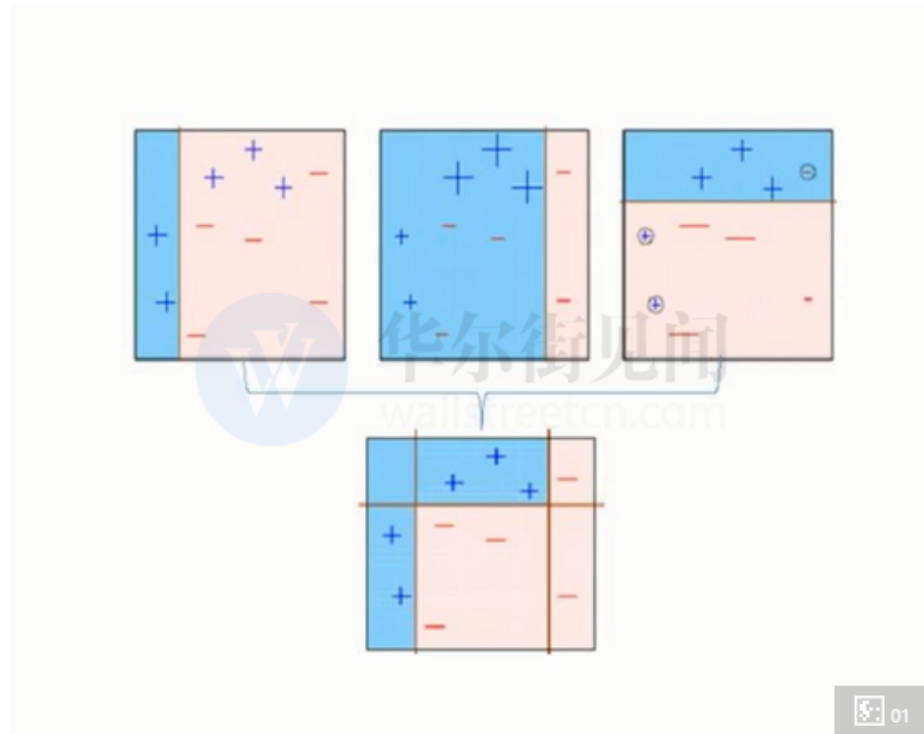
【学业总结】量化学习的脉络梳理, 以及继续学习提高的路径

2019-04-12更新

因子研究之机器学习基础. 第30讲

2019-03-22更新





(弱学习器组合示例图)

比如我们的目标是正确的区分加号和减号。那么这张图的上半部分是三个弱学习器，每一个学习器都不是特别的精确。比如最左边的学习器，它这两个加号分的是对的，但是右面它有五个减号，还有三个加号，所以它这三个加号显然分错了。

中间的也是一样，它对右边的两个减号分的是没有问题的，但是左面对于加号的分类里面又混进来三个减号。第三个学习器也是一样。但是当我们把三个学习器组合起来，用一个简单的投票的方式，就组成了一个强学习器。

下方块大家可以仔细的去看一看，三个这么一投票，所有对于加号和减号的分类都是正确的。其实这就是集成学习最基本的一个思想：三个臭皮匠顶个诸葛亮。



为什么最近几年集中学习，忽然就火起来了，其实以前也有，但是我觉得一个重要的推动力就是IT技术的快速发展，带来强大的计算能力，并行运算、包括运用图形处理器做GPU来进行运算等等。

原来你要做这种集中学习，可能要非常大的计算机才可以。现在普通的服务器、甚至一些高端的家用台式机也可以做类似的研究。

所以机器学习突然火起来也是的有一定道理的，有些老的方法几十年前就有。但是在当时它不具备运用的条件。我还想起来我上中学的时候首次接触计算机，30年前的事了，那是才64K内存，不要说机器学习了，做个矩阵运算都非常费劲的。

而现在的计算机是无论是从内存还是处理能力都非常强大。另外现在开源的技术的广泛应用，使得一些比较复杂的算法，在实现层面的问题，世界上那么多聪明人都帮你解决掉了。

所以从一个量化研究者的角度而言，我们真的应该感谢伟大的时代。这么多的好的方法，好的技术都赶来给我们用，还很多还是免费的，我们真的是应该做出点好东西来。

集成学习代表方法

感想发完了，我们来看一看集成学习有什么代表的方法。

其实就两大类。第一类叫**Bagging**，它是一种并行的方法，多个弱学习器来进行投票。比如我们右图这个事例就是一个Bagging的方法。



还有一种叫**Boosting**，它是一个串行的方法，就是每一个学习器是对之前的结果来做增强。比如说我第一个学习器可能学到的是价值投资，有价值的股票就容易是好股票。但是这样你可能会遗漏一些成长股，没有价值的，成长性高的也可以，那么第二个学习器就把这点给学到了。

既没有价值又没有成长的股票就不可能是好股票吗？其实也未必，它可能是一个时髦的概念，也可以是好股票，那么第三个、第四个学习器再来学习到这样的规律。当然这是一个非常简单化的说明了，实际在机器学习的过程当中，他学到的可能是一些维度更高更复杂的规律。

总而言之，串行的方法就是一个摞一个来获得更好的学习的结果。在实践当中集成学习是常用，也是效果比较好的方法。下面我们就来简单介绍两种常用的集成学习的方法。

集成学习方法

第一个是Bagging方法的应用，也就是随机森林的方法。这个逻辑说起来是非常简单的。我们前面讲过决策，树决策，树就是一棵树来做决定。那么随机森林的主要思想就是我用多棵树来做决定，然后把预测结果来取平均。

我们下图是一段Python的示例代码，就是把**应用随机森林模型最核心的代码**给列在这儿了，其实大家看并没有多少行，我们前面也讲到了，机器学习应用，其实这个算法本身没有那么复杂，因为别人都已经帮你做好了，你只需要调用就可以了。



```
#随机森林模型的核心Python代码
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
...
model = RandomForestClassifier(n_estimators = 10,
                              max_depth = 4,
                              max_features = 10)
...
model.fit(X_train, y_train) #训练模型

y_pred = model.predict(X_test) #样本外预测

accuracy_score(y_test, y_pred) #分析结果

# 获取在线帮助
? RandomForestClassifier

# 察看Object 提供了哪些方法，有哪些变量
dir(RandomForestClassifier)
```



(一段Python的示例代码)

最关键的还是要去思考怎么样来构建这个问题。从X到Y的映射，我这个X到底是什么？用了哪些因子？我这个Y到底是什么？我是研究收益率还是研究分类问题？我们选股空间到底是什么……同学们可以去回顾前面几讲讲过的这些构建问题的基本的要点。

当你这个问题构建好了，你知道X什么Y是什么了？真正的拍档代码倒是挺简单的。我们看到这几个上面，首先是你要把一些常用的包给他import进来。



前面讲sklearn这是一个基础的机器学习的包，在sklearn里里面又有一些常用且比较有用的子包，比如accuracy_score，这是帮你来判断机器学习模型的正确率的；还有一个叫train_test_split，这个是用来做训练集合测试集，来分割的。

你这几个包都import进来之后，定义模型就是这么一句话。也就是说我是一个用随机森林来做分类法，RandomForestClassifier。当然这里参数比较多，我这个样例里面包含了三个参数，这是第一个“n_estimators”，我是设了一个“10”，我用十棵树来组成这个随机森林。然后第二个参数是最大深度，设为“4”。

我们回顾一下前面讲过的，为什么要设最大深度，主要是为了防止过拟合，同样我还可以规定每棵树最多能用到多少特征。因为我原始的输入特征可能是很多的，可能是有30个50个这样的因子。但是随机森林的一个重要想法就是每一棵树只是用到其中一部分的特征，通过这种方式来防止过拟合，去挖掘数据当中真正的可以泛化的规律。

把这个模型定义好之后，底下几行红色标出的其实就看似非常简单，你的训练模型只要调用fit函数就可以了。你把训练集的X和Y扔给他，模型就会去从中企图去学习这个规律，把参数都给你配好。

有了已经调好了模型之后，你用它来做预测，怎么做呢？就是调用这个模型本身的预测函数，predict函数。你把测试集的X给他扔进去，他就得到了预测结果。

比如我们如果是选股的话，可能测试集的X就是我站在今天这个时间点上，这个因子的取值是什么样子的。我把它放到随机森林的模型当中，模型就会告诉我，每一只股票在下一期、在未来它是好股票的概率是多少，是差股票的概率又是多少。



最后我们分析结果，就可以用accuracy_score这个函数，我们把预期的结果和实际的结果放在一起，看它的正确率多少、召回率是多少等等。

关于Python的小贴士

我们开放的函数很多，方法也很多，我们在实践当中经常会遇到一些问题，就是到底有哪些参数、我们又怎么使用。那么除了在网上去获取帮助以外，Python也有简单的在线帮助，比如有任何一个函数或者任何一个类，你只要前面打个问号，它能够告诉你基础的参数是什么，也有一些基础的样例。比如你打一个问号，然后RandomForestClassifier，就能够获取基础的信息。

那么这个类里面有哪些方法？有哪些变量，你就打一个dir,然后把这个类给扩起来，他就会给你列出来，这里类里面有哪些子变量？有哪些方法？

比如说随机森林分类器，你会看到它有一个有一些基础的方法，比如刚才我们用到的fit是定模型参数，还有predict就是做预测，这些都是它基础的函数。

所以随机森林是一个常用的Bagging的方法。在实践当中有几个特点，第一个它的预测效果还是不错的，一般来说是比1颗决策树肯定是要好，比逻辑回归常常也是要好的。

但是它也有它的问题，就是在树的个数比较多时候，模型参数的确定是比较缓慢的。

XGBoost-Boosting方法的应用

接下来我们就向大家介绍一种跑起来比较快的一种模型，也是现在很流行的叫XGBoost，他就是刚才我们讲到的Boosting算法的应用。那么他的主要思想是什么呢？



它可以用决策数线性回归或者逻辑回归做继续学习。我们前面讲过了，这些都是复杂机器学习的基础组件。和其它的Boosting算法一样，它每一个学习器是在上一轮学习的基础上，通过重点学习预测错误的样本，来获取更高精度的预测。

比如说在我们简单的说明性的事例当中，我先学习价值投资的逻辑，在学习成长股投资的逻辑，最后在学习概念投资的逻辑，是这样一轮一轮来的。

说起来其实没有什么复杂的，但是XGBoost为什么会这么流行呢？

它有很多时限上的优化，比如他直接就把常用的一些正则化的方法，给你融入到算法当中了。他还通过在寻找最优质的过程当中，传统的梯度下降的方法只用到一阶导数，它是一阶导数、二阶导数一起用。

这还涉及到一些数学了，如果学过一点高等数学的同学们就都能够反应过来，这个速度就快了，他能更快地逼近极值点。

然后有一些列抽样等等其他的方法都可以包含进来了。最后还有大量的工程上面的优化，它自动的就支持并行处理，如果你的计算机上有图形处理器的话，它还会用图形处理器来帮你加速运算等等。所以XGBoost的运行效率是很高的。

从拍档的实现代码上来讲，你看这核心代码长得跟刚才随机森林还挺像的。所以我就反复说最重要的是构建这个问题本身，实际的代码调用是比较相对来说比较简单的。



XGBoost模型的核心Python代码

```
import xgboost as xgb
from xgboost.sklearn import XGBClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
```

```
...
model = XGBClassifier(n_estimators=100,
                      booster='gbtree',
                      max_depth=3)
```

```
...
model.fit(X_train, y_train) #训练模型
```

```
y_pred = model.predict(X_test) #样本外预测
```

```
accuracy_score(y_test, y_pred) #分析结果
```

03

一上来你先import一些XGB相关的包，Sklearn里面是没有这个的，但是它有一个所谓的API，就是你通过Sklearn也是可以用这个包的，我们把它都包括进来了。在定义这个模型的时候也是一个分类器，XGBClassifier。

我这儿列举了三个比较重要的参数，第一个我用一百颗树串行来学习，来做XGB的分类器。第二个我学习用什么？我这定义了一个gb树，这是树模型，你也可以用GB理念来线性模型等等。



第三个,参数我也定义了模型的树的深度。我们复习一下,前面讲过的不能太深,要防止过拟合,所以比如示例性的你可以用3、用4,这都可以。

定义好了,后面其实跟刚才的随机森林都是一样的,你就直接调用它的fit来训练模型,你调用它的prefict来做样本外的预测,最后你可以用accuracy来分析结果。

所以一旦问题定下来之后,使用起来还是蛮方便的。在实践当中,XGB第一个他的预测效果不错,第二个它重大的优势就是速度快,所以是很流行的集成学习的方法。

说起来XGBoost的是比较新的方法,大概是2015年、2016年才为大家所知。主要的发明者还是一位来自中国的计算机的大拿,应该是毕业于上海交大的陈天奇博士。

看到这样的方法出来之后,我一时也是觉得后生可畏。八二也是觉得其实中国人在AA的领域,我觉得未来还是有很光明的发展前途的。如果你去看一些机器学习的比较新的论文,大量的都是来自中国的作者,所以我觉得未来在这个方面,中国人还是大有希望的。

机器学习 and 深度学习对比

我们刚才讲两种方法,一个是随机森林,一个是XTBoost。那么我们简要的来谈一谈深度学习,因为平时大家都可能听说的比较多了,又是AlphaGo又是人工智能选股……

讲实话,我觉得有些也是比较夸张的。说人工智能选股的,十个里面可能有七八个都是有很大做销售或者直接是PPT忽悠的成分。

那么量化小学我们还是希望给大家介绍一些常识,让大家以后再听到这样的说法的时候,至少能大概的知道别人是在讲什么,有一个基本的判断,真的概率是多大。



深度学习到底是什么呢？我们先来和传统的机器学习做一个对比。

1.经典机器学习（统计学习）

上一讲和这一讲刚才讲过的，说实话都是一些经典的机器学习的方法，也可以叫统计学习。他其实就是通过一些统计方法来学习样本中存在的比较具体的规律。

比如说我们可以用机器学习来解决疾病诊断的问题，一个人到底有没有得过某种疾病，如果我们看过足够多的病例的话，就知道用什么样的验血指标、什么样的化验指标来作出判断。

那么在量化研究当中，我们做选股的时候，通过足够多的历史数据，我们也可以分析出来，哪些因子是具有超额收益的。所以这件事本身还是一些比较具体的规律，也比较容易理解。

训练这个模型的样本多一点，当然肯定是没有坏处的。但是如果你这个规律比较明显，其实往往你不需要特别多的样本，你就能够找到他。

比如我们反复谈到的价值投资，就是一个选股策略当中一个比较基本的规律，也用不了特别多的数据，大概这个计算机就会学会，还是买茅台、招商银行这种股票比较靠谱。

所以这个是经典的机器学习的方法，那么深度学习又是什么呢？

2.深度学习（deep learning）



他是试图从大量的样本当中学习到一些类似于逻辑和抽象规律的东西，这就和传统的经典的机器学习里面比较具体的规律不一样了。

比如说经典的深度学习的应用场景，包括图像识别和语义分析。

我们企图让机器学会怎么样区分一只猫和一只狗。这个人呢，你看了猫和狗的照片，你都知道哪个是哪个，但是如果说让你精确的说出来，你是怎么样来区分一个猫和一个狗的？其实我们不太能够说得出来。

当然可能猫有胡子、狗的耳朵比较大，但是光靠这些你其实还不能够完全的做一个区分。如果这个猫转过去怎么办？或者说狗是只露出一个狗尾巴怎么办？人当然还是能够分出来的，但是你是怎么分出来的？其实规律说的不会那么清楚，它其实是一种抽象的规律。

深度学习就是企图去掌握这样的规律。既然规律是这么抽象，既然我们学习的是他的逻辑，那你在训练的过程当中，你可想而知，你是需要非常多的样本的！

我们再一次提到，就是说之所以深度学习，在最近这几年火起来了，也有了非常多非常好的应用。第一有了非常强大的计算能力，第二个还有了很多的数据，因为各种各样的社交媒体网络越来越发达了，有了很多的数据才能够从中去学习这种比较抽象的规律。

因此，在金融上面使用深度学习这个想法是非常好的，但是在实践当中一定要注意一个问题，金融的数据它本身就是信息含量比较少，噪音又比较多，那么深度学习又要求有海量的数据才能够学得出来。

因此在金融的应用当中，你一定要想一想，你的数据点够不够、能不能够让这个系统真的学习到规律。所以最好的应用场景还是有大量数据的时候，比如说比较偏高频的研究，比



较偏日内的研究等等。

如果你数据点本来就不多的话，用深度学习，八成是学不出什么东西来的。它不足以对逻辑或者规律做出足够多的抽象，可能还是用经典的、机器学习算法更靠谱一些。我们刚刚也讲过，它需要的训练数据的数量，一般来说是比深度学习要少的。

从方法论的角度来说，深度学习是模拟人类的思维和学习方式，是一种神经网络或者是带反馈的强化学习，我们马上会具体的看一看神经网络究竟长什么样子。

最后，模型本身就挺复杂的，参数动辄上万，甚至上十、上百万的也有。这样的模型往往就不容易直观理解。所以大家又说具有黑箱的特点。

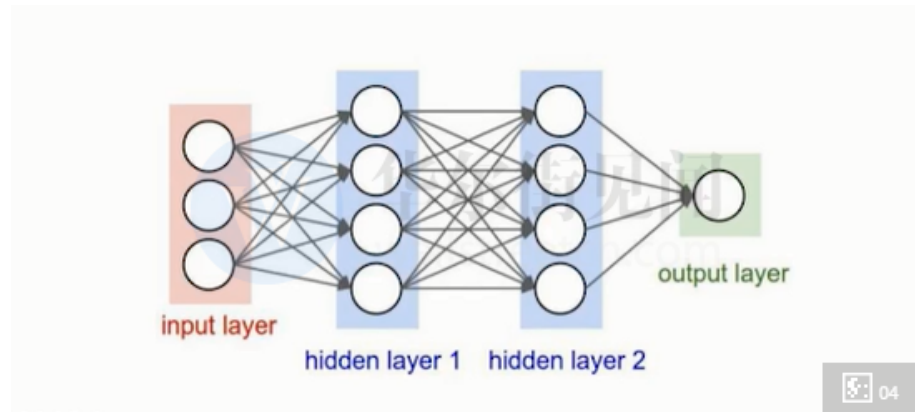
深度学习在量化研究中的应用

基础架构：神经网络

我们来非常简单地给大家介绍几种常用的深度学习方法。

首先基础的架构就是神经网络（见下图），神经网络它是模拟人类的思维方式，是用神经元来处理和传递信息。我们看到它有几层，左边这层是输入层，然后中间是隐藏层，最后是输出层。





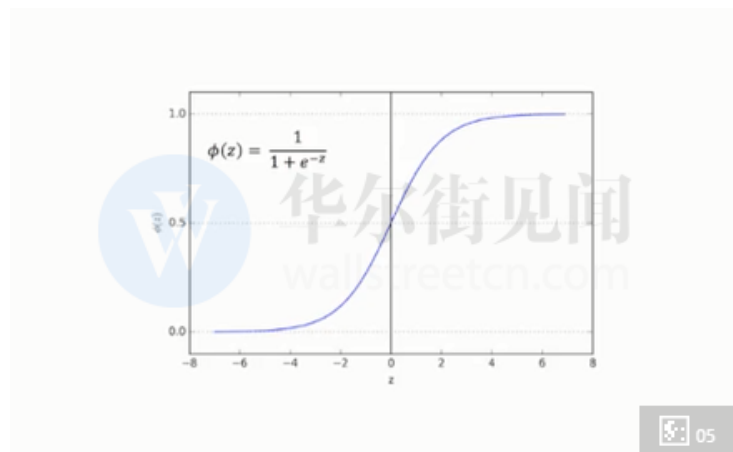
听上去挺复杂的，其实从逻辑上讲就跟人的想法差不多。每一个圆点就是一个神经元，当它受到足够的刺激的时候就会被激活，就会发出信号。比如人如果吃了一个辣椒的话，我负责感受“辣”这种感觉的神经就被激活了，就发出了信号。

那么从数学的实现角度，我们怎么样来做这个事呢？给它分层，用某种方式给它连起来。

最后神经元什么时候就被激活了呢？就是利用我们前面讲过的sigmoid函数，或者是一些其他的函数来作为集合函数。

我们复习一下，sigmoid函数在逻辑回归当中也用它，就长的是[下面这幅图](#)的样子，像一个脉冲信号一样，从0“啪”一下就变成1了。





最后怎么样fit这个模型？最主要的方法就是梯度下降和反向传播，这都是一些数学上的概念了，其实也没有什么特别深奥。梯度下降就是高等数学里面的牛顿法的一种延伸，高维的一种延伸。反向传播也是在比如解微分方程当中挺常用的。

想法虽然不复杂，但是具体的过程就是工程学的问题了，也有人把它比喻为炼丹，需要很多的经验和技巧。

用一个形象的比喻，神经网络搭好，就像一个刚出生的小孩，他是一张白纸，然后你就把各种各样的数据喂给他。如果学得好的话，就学出一个聪明孩子来，他可以解决你要预测的问题。

也常常有学不好的时候，我们经常听说一个词叫“梯度消失”。我的解读，就是学傻了，他已经对外界的信号不作出反应了，你再给他新的数据，它的参数也不更新了，梯度消失了。而另外一种极端就是“梯度爆炸”。

这个模型非常不稳定，你把新的数据给他的时候来回动着，就好像人变疯了。所以深度学习这个想法是不错，但是在实践当中也是用起来也不那么容易，需要很多的经验。想想搞



出一个聪明孩子来，往往是要花费很多精力的。

深度学习的几种实用架构

所以深度学习的基础架构就是刚才我们讲的神经网络，但在实践当中它的实用的结构会更复杂一些，这里非常简单的介绍一下。

比如大家常听说的**卷积神经网络（CNN）**，主要是用于图像识别的，像围棋领域，AlphaGo也是用到了很多深度卷积神经网络的技术。在量化研究当中，理论上来讲可以用它来做类似于“读图”这样的事情。

还有一种叫**递归神经网络（RNN）**，这就是说他也是在模拟人类的思维方式，因为我们有时候是带有记忆的。比如说我们在传统的股票研究中，我们讲过截面研究，我只取一个截面，用截面的信息来决定好股票还是坏股票。

但人类的思维它不是这样的，它除了这个截面之外，我还对历史有记忆。比如说历史上这类股票可能涨过，那么对于我做决策是有影响的。涉及到深度学习呢，它就用在递归的神经网络来解决这个问题。也就是说，信息不但是从前向后传播，它还可以转圈的。

在实践当中，现在大家常用的是一种叫做**长短记忆网络（LSTM）**的东西，或者是与之类似的，比如说GRU等等，其实他是就是一种大幅改进的RNN，具体的数学结构大家可以自己去看。

他还是很巧妙的，它就用了各种激活函数来实现了一些遗忘、长期记忆等的功能。在量化研究当中，我们可以用它来做高危的时间序列分析。时间序列它是带有记忆的，这个股票如果三天前表现好，这两天调整了，那么是不是明天又该涨了？



就是类似于这样的规律，理论上讲可以用LSTM这样的网络来学习。当然我们前面也讲了，一个聪明孩子在实践当中弄成傻孩子和疯孩子的概率也不小。

深度学习在Python实现

好，非常简单的介绍了几种深度学习的结构，那么在Python当中的实现没有刚才我们讲的随机森林和XGBoost这么方便，但是也还还算可以，因为底层的工具包别人都帮你做好了。

两个比较流行的，第一个是**Tensorflow**，这个是google做的，第二个是**pytorch**，这个是Facebook做的。两者功能都很强大，是深度学习的神器，实践当中其实你掌握一个就可以了。

那么既然是神器，它使用起来就是比较复杂的。比如Tensorflow，就是“张量”的意思，就不是一个二维的数据结构了，它是一个高维的数据结构。你在这上面要在做一些操作，对于初学者来讲，它还是比较不容易掌握的。

所幸的是世界上还是有很多雷锋的，有人也体会到了初学者的难处，所以就提供了框架式的工具包，就是Python里面的“Keras”。还是能够大大简化对深度学习的门槛的，因为他对Tensorflow、对pytorch进行了封装，使用起来的难度会大大的降低。

有兴趣的同学们都可以去去看，那么后面我也会给大家推荐几篇研报，是具体地讲怎么样利用这些工具的。

机器学习应用的原则



近两期我们用不多的时间简要地介绍了机器学习和深度学习，还有集成学习，主要是希望大家先提纲挈领地来理解这些方法究竟都是干什么的，他们大概的应用路径是什么样子的？有了这样的基本的知识，再去看那些研报，再去看那些论文，就有事半功倍的效果。

在结束本章之前，我们也对机器学习的应用做几点总结，其实很大程度上也是对因子研究的一个总结。

首先还是要再次强调，**构建问题比使用什么方法更加重要**，你的选股信号最好是要有金融逻辑支撑的。

那么在因子的选择、因子的处理方面要独章功夫。我们前面讲到的选股空间、行业市场中心化等等，都是要仔细思考的。

第三点，你要想明白，究竟让计算机学习什么最容易发生效果。你是分类还是回归，你是要选牛股，还是要避免踩雷，也就是所谓监督学习当中的学习目标设定的问题。

第二个原则是，实践当中的质量控制，比精细化的调参更重要。机器学习算法的参数很多，在初学的时候常常会给你一种错觉，就像你坐进一个高大上的汽车里面，你看到有这么多的操纵的按钮都可以按。但其实初学是不应该过多地花精力在调参上面，机器学习的默认值往往就是最优的。

第二点，你有时间应该更多地花在怎么样构建问题、怎么样去解读机器学习的结果上面。还有特别重要的，就是要保证工程的质量。

首先是数据质量，数据非常容易有错漏，用的不好还会有所谓的前视偏差，所以你应该第一个注意，第二就是要利用到quant<OS>提供的像“Dead view”这样的工具来帮助你避免



这样的问题。

另外机器学习的程序往往比较复杂，也容易出bug，这也是需要仔细的地方。

在实践当中也有很多细节的问题，比如说你计算速度是否足够快，在示范当中能不能跟得上市场的节奏，数据源是否可靠。最后你有了很漂亮的模型，很好的结果，你的交易执行的滑点和完成率怎么样，能不能让你在实盘中拿到这样的结果？这都是我们更应该关心的问题。

所以最后，机器学习确实是非常有用、非常强大的工具，但是不应该过于把它神化，我们做研究的最主要的精力，我个人认为并不应该放在这些方法上面的。

学习资料

好，今天我们简要的向大家介绍了集成学习和深度学习的方法，因为课堂上的时间有限，我们可能讲得不是很详细，请大家课后去阅读一些很好的资料。

在这儿我一口气给大家推荐了三篇华泰证券关于人工智能的研究报告。第一篇是《**Python 实战**》，它提供了许多的样例的代码，稍微改一改就可以直接用到quant<OS>平台上。

第二篇是讲了我们今天谈到的**随机森林的模型**，你也可以尝试着去复现这个研究。

第三篇讲的是今天我们谈到的**Boosting的算法**，主要是XGBoost。



学习资料

1. quant<OS>金融终端教学频道：<https://www.quantos.org/courses/index.html>
2. 《人工智能选股之Python实战》- 华泰证券，2017年9月19日
3. 《人工智能选股之随机森林模型》- 华泰证券，2017年8月31日
4. 《人工智能选股之Boosting模型》- 华泰证券，2017年9月11日



我觉得量化学习就一定要动手，如果你在计算机面前把这些研究报告当中提到的模型实际给它做一遍，那么我相信对机器学习的了解就完全是不一样的水平了。所以希望走专业道路的同学，请不但要认真的阅读这些研究报告，而且还要自己动手试一试。

好，今天的内容就到这里，下一期将是我们量化小学的最后一期，我们会对整个课程做一个全面的总结。谢谢大家。

-END-

加入“量化小学”的见识圈，关注动态

感谢您订阅本特辑，扫描下方二维码或[点击圈子链接](#)，即可加入专属见识圈子提问交流





量化小学



渔生

小学而大不遗，量化师生联谊会

感谢大家订阅《量化小学》，这里是学校见识社群，你可以随时提问、随时互动，我们一起投资，一起分享！



风险提示及免责条款

市场有风险，投资需谨慎。本文不构成个人投资建议，也未考虑到个别用户特殊的投资目标、财务状况或需要。用户应考虑本文中的任何意见、观点或结论是否符合其特定状况。据此投资，责任自负。

写评论

请发表您的评论



表情

图片

发布评论

华尔街见闻

- 关于我们
- 广告投放
- 版权与商务合作
- 联系方式
- 意见反馈

声明

未经许可，任何人不得复制、转载、或以其他方式使用本网站的内容。
评论前请阅读网站[“跟帖评论自律管理承诺书”](#)

法律信息

- 版权声明
- 用户协议
- 付费内容订阅协议
- 隐私政策

违法和不良信息

举报电话: 021-60675200 (周一到周五9:30-11:30, 13:00-18:30)
举报邮箱: contact@wallstreetcn.com
网站举报: [点击这里](#)



华尔街见闻APP



华尔街见闻公众号



微博@华尔街见闻



中央网信办
违法和不良信息举报中心

上海市互联网
违法和不良信息举报信息

[违法和不良信息举报受理和处置管理办法](#)

[清朗·财经违规内容专项整治公告](#)



举报中心

友情链接

[腾讯财经](#) | [财经网](#) | [澎湃新闻](#) | [界面新闻](#) | [全景财经](#) | [陆家嘴金融网](#) | [富途牛牛](#) | [网易财经](#) | [凤凰网财经](#) | [虎嗅](#)

© 2010 - 2022 上海阿牛信息科技有限公司 版权所有 沪ICP备13019121号  沪公网安备 31010102002334 号 增值电信业务经营许可证沪B2-20180399

