caiyongji
公众号: caiyongji 公众号: 蔡永吉

机器学习(一)：5分钟理解机器学习并上手实践

引言

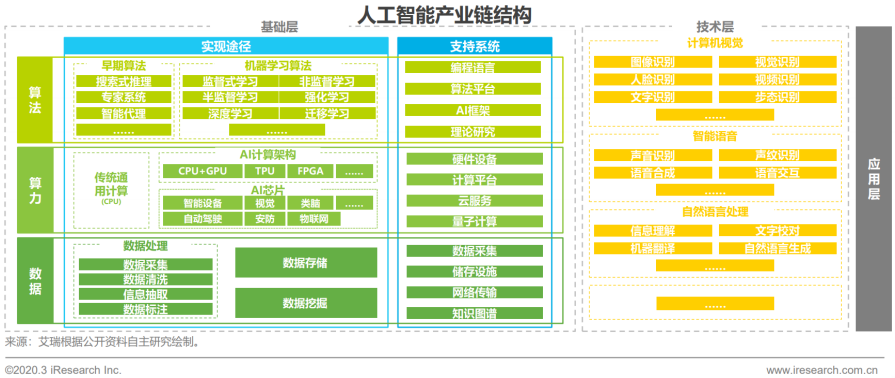
现在市面上的机器学习教程大多先学习数学基础，然后学机器学习的数学算法，再建立机器学习的数学模型，再学习深度学习，再学习工程化，再考虑落地。这其中每个环节都在快速发展，唯独落地特别困难。我们花费大量时间成本去学习以上内容，成本无疑是特别昂贵的。所以我们不如先“盲人摸象”、“不求甚解”地探索下机器学习，浅尝辄止。如果想到自己的应用场景，再学以致用，深入探索。这无疑是使沉没成本最低的决策。
本教程适合兴趣广泛的人士增加自己知识的广度，从应用的角度谨“使用”机器学习这款工具，是典型的黑盒思维。这非常契合笔者的思维方式，当然也是我个人的格局局限。
本教程会浅显易懂，让你走的很快。但如果你想走的更远还请学习数学。当然我们也只是暂时放下数学，先构建自己的知识体系。

先抬头看路，找准适合自己的方向，再埋头赶路，或深耕下去……

把视角拉高

从手工到工业化再到人工智能，这是把人类从生产活动中逐渐解放的过程。用机器来帮助人们工作，一直是人类的美好愿望。让机器智能化，以此来代替人力做更智能问题，这可以作为人工智能的简单解释。
很多教程或者书籍把人工智能、机器学习、深度学习的关系解释为从属关系，人工智能 > 机器学习 > 深度学习。这种解释不错，但却无法表示他们之间的更深层次的关系。
机器学习是通过数学方法在数据中寻找解释，以此来实现人工智能的一种手段。而深度学习是参照神经系统在机器学习基础上发展出的一种高级技巧。 它们之间是存在一定的依托关系、进化趋势的。
狭义地讲，传统的机器学习是通过数学模型不断求导来找出数据规律的过程。这其中数学模型的选择尤为重要。随着GPU、TPU等算力的发展，算法技术的进步，甚至出现了自动选模型、自动调参的技术。我们可以构建复杂的神经网络结构，只要有足够的算力支持，足够的时间我们可以用深度学习处理非常复杂的任务。所以在代码操作上，深度学习甚至比传统的机器学习对程序员更友好、更易理解。我们先学习传统机器学习而非直接学习深度学习的好处是，我们可以通过对“黑盒”的拆箱来理解机器学习过程，掌握机器学习的概念，我会对其中应用的数学模型进行解释。

我们先来看一下人工智能产业链的结构，如下图：



我们可以看到，机器学习的三大基石---算力、算法与数据。机器学习的发展离不开算法数学的进步，同样离不开算力的发展。
在技术层面，机器学习在计算机视觉（CV, Computer Vision）和自然语言处理（NLP, Nature Language Processing）取得了关键的发展和应用。
算法分类上，机器学习分为监督学习、非监督学习、半监督学习、强化学习等。

- **监督学习**：数据样本有标签。
- **非监督学习**：数据样本无标签。

昵称： CaiYongji
园龄： 6年10个月
粉丝： 43
关注： 2
+加关注

< 2021年8月 >

日	一	二	三	四	五	六
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

搜索
找找看
谷歌搜索

常用链接
我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签
机器学习(16)
深度学习(14)
神经网络(5)
程序员(4)
程序人生(3)
面试技巧(2)
面试(2)
人工智能(2)
Tensorflow(2)
chrome(2)
更多

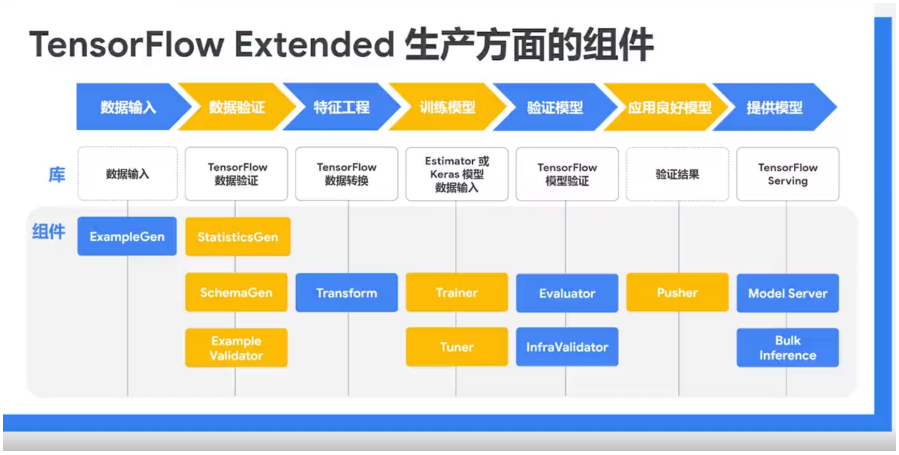
随笔档案
2021年7月(1)
2021年2月(3)
2021年1月(2)
2020年12月(5)
2020年11月(1)
2020年5月(2)
2020年4月(1)
2019年4月(5)
2018年7月(1)
2018年5月(2)
2018年4月(1)
2018年3月(1)

- **半监督学习**：数据样本有部分(少量)标签。
- **强化学习**：趋向结果则奖励，偏离结果则惩罚。

所谓Garbage in, Garbage out(垃圾进，垃圾出)。数据是机器学习的重中之重。我们需要花费大量的时间来处理数据，甚至占到整个机器学习任务的90%以上。
比如数据处理过程中的数据采集，如果我们采样的方式欠妥，就可能导致非代表性的数据集，这就导致了采样偏差。
我们的数据可能会有很多无效的数据，我们需要剔除无效的数据，就叫做数据清洗。
我们通过挖掘大量数据来发现不太明显的规律，就称作数据挖掘。

机器学习工业化流程

我们以一款工业化流水线工具TFX为例，看一下机器学习的技术流程。



流程分为数据输入、数据验证、特征工程、训练模型、验证模型、应用良好模型和提供模型六个部分：

1. 输入数据，并根据需要拆分数据集。
2. 生成训练数据和服务数据的特征统计信息。通过从训练数据中推断出类型、类别和范围来创建架构。识别训练数据和服务数据中的异常值。
3. 对数据集执行特征工程。
4. 训练模型，调整模型的超参数。
5. 对训练结果进行深入分析，并帮助验证导出的模型。检查模型是否确实可以从基础架构提供服务，并防止推送不良模型。
6. 将模型部署到服务基础架构。

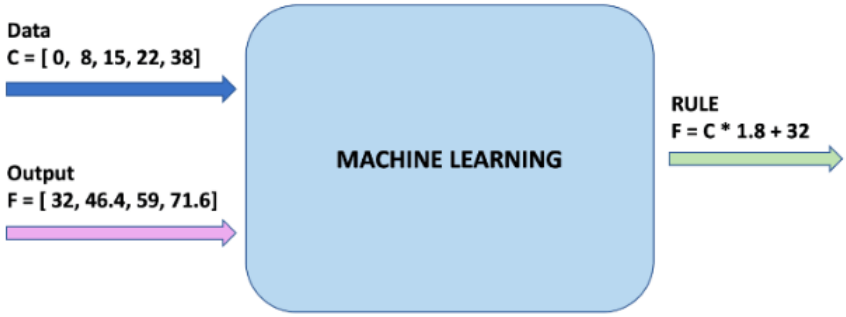
我想通过以上解释，大家应该可以对机器学习的实践方法有了一定宏观的了解。

机器是如何学习的

我们从宏观角度看了机器学习的产业结构、工业化流程，你应该对你自己在机器学习的这些环节中有哪些发挥有了一定的把握。现在我们把视角拉回到微观层面，看看机器是如何学习的。

我们以摄氏度转换华氏度为例。
传统编程中，我们要求得摄氏度和华氏度的关系，我们必须找出公式：
$$ Fahrenheit = Celsius * 1.8 + 32 $$

而在对机器学习来说，我们有大量的数据，却需要找出关系。机器学习的过程就是不断求导，以此来找出数学模型，来解释规律的过程。



2018年1月(5)
2017年12月(2)
2017年10月(1)
更多

最新评论

1. Re:机器学习(二)：理解线性回归与梯度下降并做简单预测
就感觉把数学知识都还给老师了，但看着您写的又觉得好有道理
--yangboom
2. Re:机器学习(一)：5分钟理解机器学习并上手实践
整体意识，大局观很重要
--豕叒双又
3. Re:防卒指南：996+健身≈猝死
间接性死亡
--敲代码改变不了中国
4. Re:防卒指南：996+健身≈猝死
看博主是在01:30发的文章，心疼你2秒
--不懂01的ITer-Jack
5. Re:防卒指南：996+健身≈猝死
没错，996只是暂时的，不会一辈子，可健康是一辈子
--绝叫の白头翁

阅读排行榜

1. github emoji 表情列表(5841)
2. 你的知识死角不能否定你的技术能力(4813)
3. 程序员必备工具目录(2956)
4. 微信红包的随机算法是怎样实现的？(2689)
5. 发布 Google Chrome插件教程(2365)

评论排行榜

1. 你的知识死角不能否定你的技术能力(52)
2. 防卒指南：996+健身≈猝死(5)
3. 如何正确的提问题(2)
4. 机器学习(二)：理解线性回归与梯度下降并做简单预测(1)
5. 机器学习(一)：5分钟理解机器学习并上手实践(1)

推荐排行榜

1. 你的知识死角不能否定你的技术能力(53)
2. 防卒指南：996+健身≈猝死(8)
3. github emoji 表情列表(5)
4. 机器学习导图系列（2）：概念(3)
5. AI时代：推荐引擎正在塑造人类(3)

如图所示，我们有摄氏度数据0, 8, 15, 22, 38以及华氏度数据32, 46.4, 59, 71.6, 100.4，机器学习的过程就是找出公式的过程。
其中，摄氏度就是我们的**特征**，华氏度就是我们的**标签**，摄氏度与华氏度的关系就是**实例**。

- **特征**：我们模型的输入。 在这种情况下，只有一个值-摄氏度。
- **标签**：我们的模型预测的输出。 在这种情况下，只有一个值-华氏度。
- **实例**：训练期间使用的一对输入/输出。 在我们的例子中，是摄氏度/华氏度一对数据，例如，(0, 32), (8, 46.4)。

蓝色的部分表示我们设置好数学函数，然后通过不断的调整权重与偏差不断地**拟合**数据，最终得到可以表示规律的**模型**的过程。

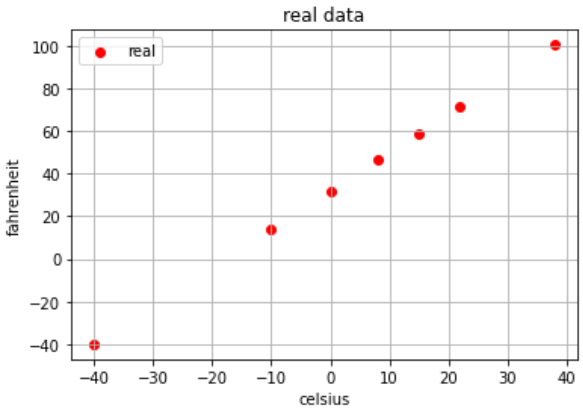
- **拟合**：通过训练数据，使模型来概括表示数据的过程。
- **模型**：图结构，包含了训练过程中的权重与偏差的数据。其中的图为由各函数组成的计算结构。

简单上手机器学习代码

在上手代码之前我默认你已经配置好了环境，掌握了Jupyter, Numpy, Pandas, Matplotlib的用法。如果你没有掌握以上技能，请参考我写的配套教程前置机器学习系列

```
import numpy as np
import matplotlib.pyplot as plt
celsius    = [[-40], [-10], [ 0], [ 8], [15], [22], [ 38]]
fahrenheit = [[-40], [ 14], [32], [46.4], [59], [71.6], [100.4]]
plt.scatter(celsius,fahrenheit, c='red', label='real')
plt.xlabel('celsius')
plt.ylabel('fahrenheit')
plt.legend()
plt.grid(True)
plt.title('real data')
plt.show()
```

如上代码所示，我们准备摄氏度与华氏度的数据，然后通过matplotlib库绘制图像。

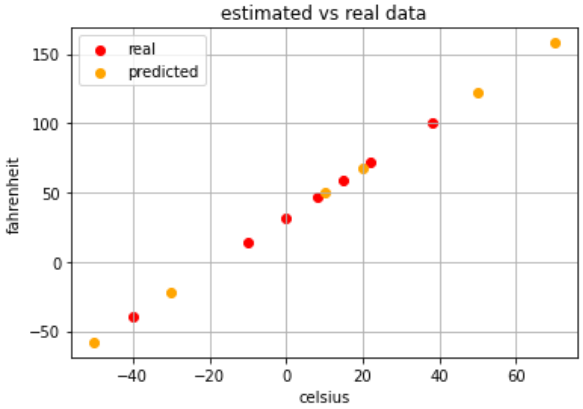



```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(celsius,fahrenheit)
```

我们通过上方仅仅3行代码就训练了数据。 `LinearRegression` 是scikit-learn包下的线性回归方法，是普通的最小二乘线性回归。而 `fit` 就是拟合的意思，以此来训练模型。

```
celsius_test = [[-50],[-30],[10],[20],[50],[70]]
fahrenheit_test = lr.predict(celsius_test)
plt.scatter(celsius,fahrenheit, c='red', label='real')
plt.scatter(celsius_test,fahrenheit_test, c='orange', label='predicted')
plt.xlabel('celsius')
plt.ylabel('fahrenheit')
plt.legend()
plt.grid(True)
plt.title('estimated vs real data')
plt.show()
```

接下来我们调用 `lr.predict(celsius_test)` 方法来进行预测，以此来检验我们的模型准确度。我们通过下方图像中黄色的点可以看出，我们的模型非常准确。



你就说这玩意简单不简单！ 咳咳，别嚣张，我们好好玩。

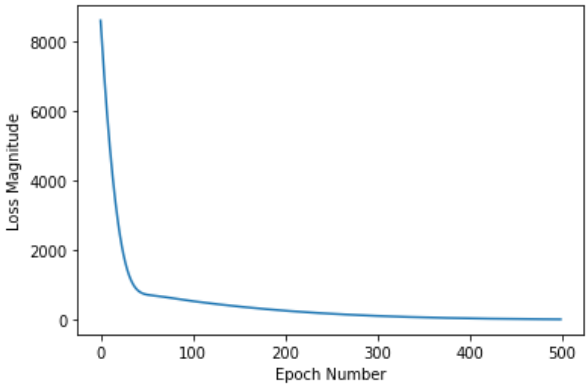
顺带一提的深度学习代码

既然都上手了，我们也试一试深度学习代码：

```
import tensorflow as tf
import numpy as np

# prepare data
celsius_q    = np.array([-40, -10,  0,  8, 15, 22,  38],  dtype=float)
fahrenheit_a = np.array([-40,  14, 32, 46.4, 59, 71.6, 100.4],  dtype=float

# fit model
model = tf.keras.Sequential([tf.keras.layers.Dense(units=1, input_shape=[1]
model.compile(loss='mean_squared_error', optimizer=tf.keras.optimizers.Adam
history = model.fit(celsius_q, fahrenheit_a, epochs=500, verbose=False)
print("Finished training the model")

# print loss
import matplotlib.pyplot as plt
plt.xlabel('Epoch Number')
plt.ylabel("Loss Magnitude")
plt.plot(history.history['loss'])
```

我们使用TensorFlow内置的Keras方法创建了1层的神经网络，选择了MSE损失函数以及Adam优化器，训练了500代。

如下图可以看到，随着代(epoch)数量的增加，损失函数的结果逐渐降低。



那么什么是损失函数呢？我们在接下来的文章中一探究竟。感谢您的关注公众号【caiyongji】与支持！

前置学习系列

- 前置机器学习（五）：30分钟掌握常用Matplotlib用法
- 前置机器学习（四）：一文掌握Pandas用法
- 前置机器学习（三）：30分钟掌握常用NumPy用法
- 前置机器学习（二）：30分钟掌握常用Jupyter Notebook用法
- 前置机器学习（一）：数学符号及希腊字母

标签：深度学习，机器学习

CaiYongji
关注 - 2
粉丝 - 43
+加关注

0 0

« 上一篇： 前置机器学习（五）：30分钟掌握常用Matplotlib用法
» 下一篇： 机器学习(二)：理解线性回归与梯度下降并做简单预测

posted @ 2021-01-16 00:21 CaiYongji 阅读(845) 评论(1) 编辑 收藏 举报