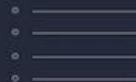
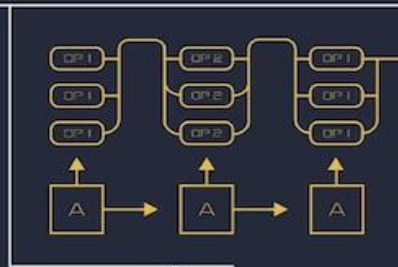


TensorFlow

从0到1之AI 路

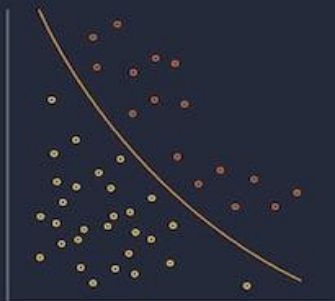
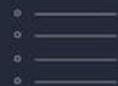


内容提要

1.人工智能落地应用

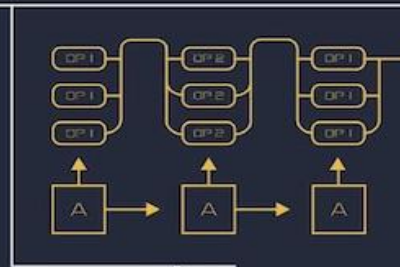
2.机器学习基础

3.使用TensorFlow构建第一个AI



TensorFlow

人工智能实践



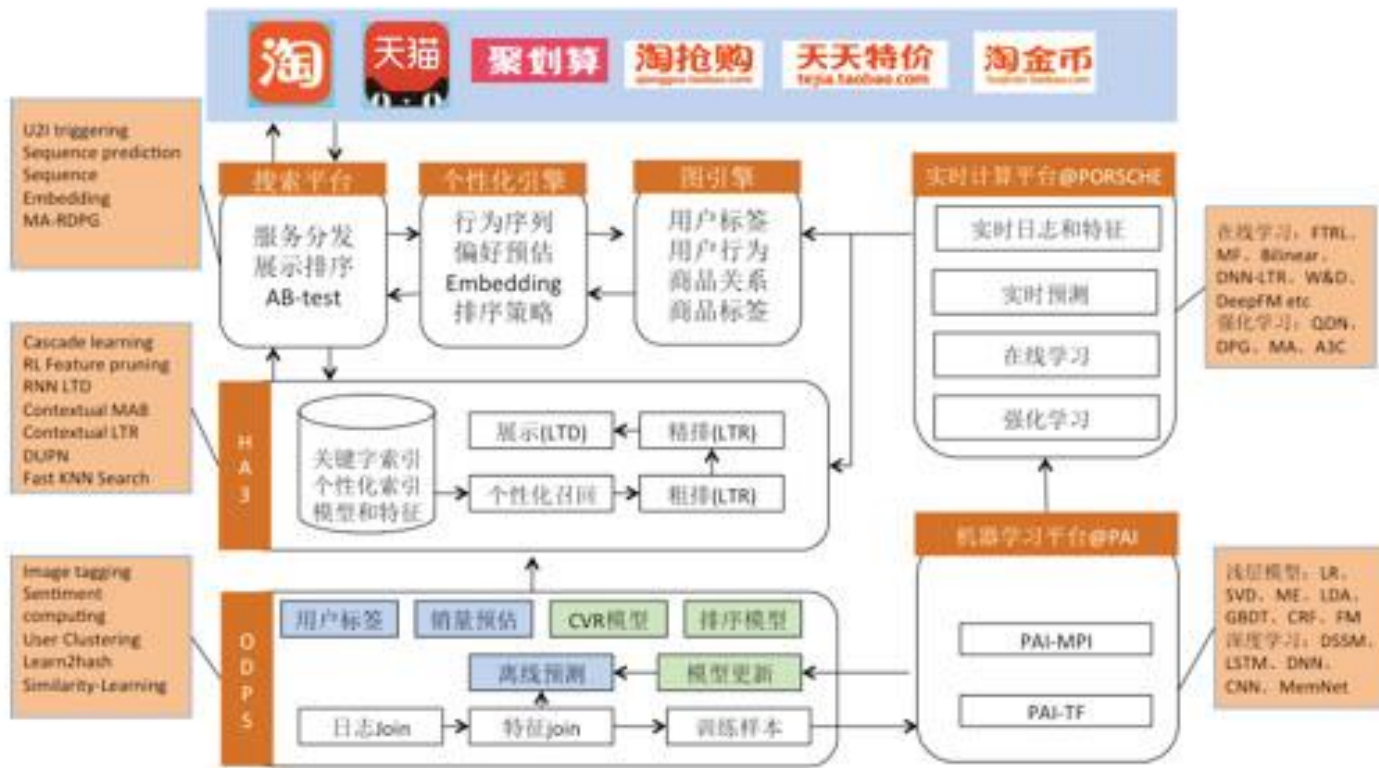
人机大战



人工智能在百度



人工智能在阿里



人工智能在腾讯



今日头条



科大讯飞



讯飞语音云3.0

Reliable

更加靠谱

Multiple

更多交互模式

Business

更好商业化支持

更多...

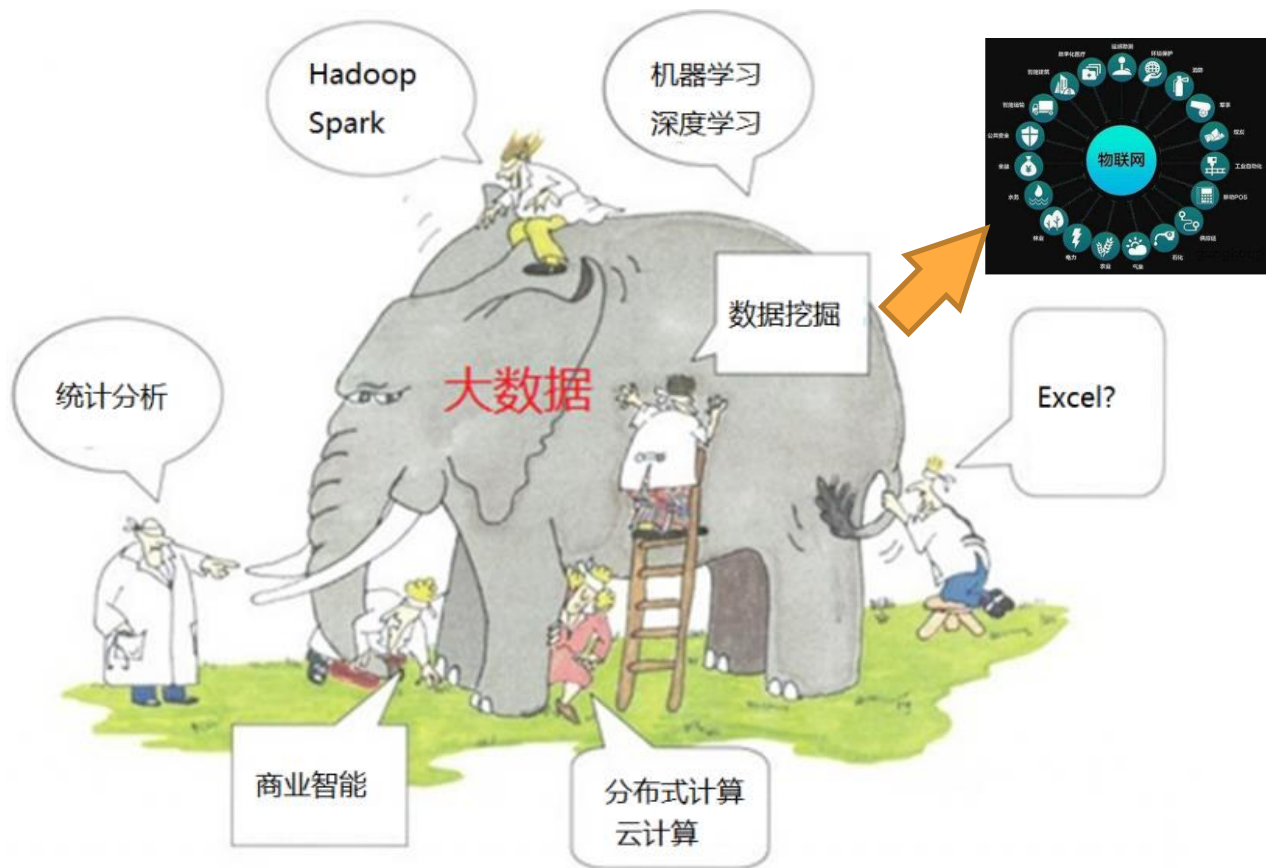


人工智能与谁相关？

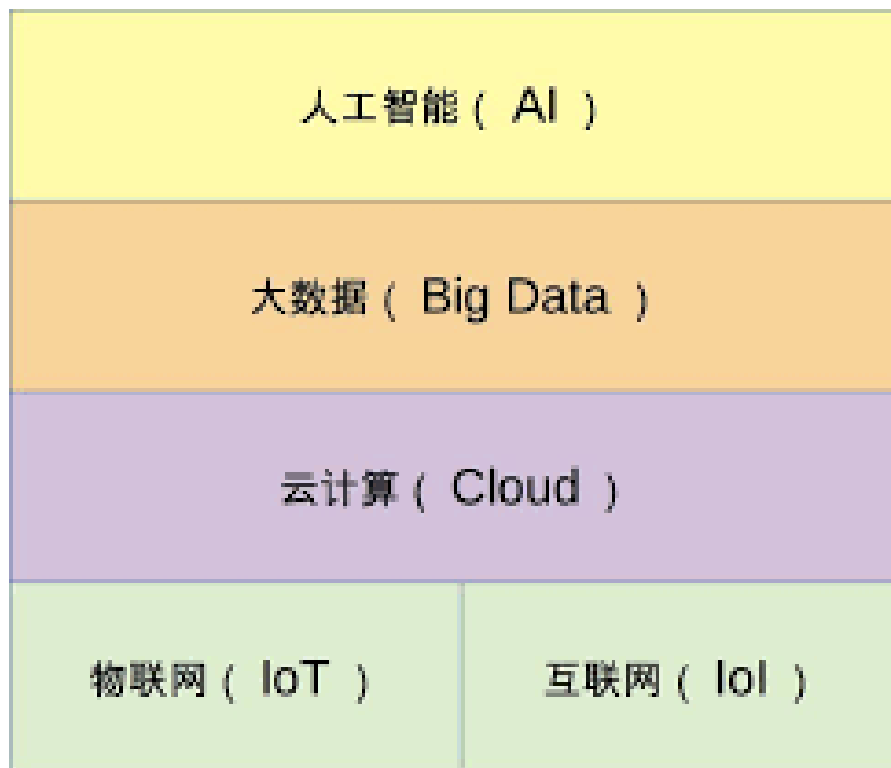
从Android到TensorFlow



“云物大智”



“云物大智”







我们从哪里开始？

✓ 人工智能我该怎么学呢？

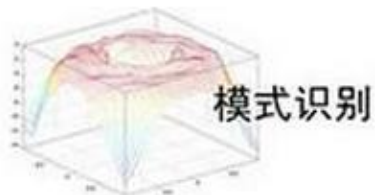
 人工智能是一个很大的圈子，但是基础必然是**机器学习**

 什么是机器学习呢？说白了就是你告诉机器你想做什么？并且给它一堆数据让它去模仿着做！
(比如，咱们上高中，老师会告诉我们一个目标就是考高分，然后给我们一堆练习册和答案，我们的目的就是让我们做的题的解和答案一致)

 机器学习需要什么？算法，数据，程序，评估，应用

✓ 机器学习能做什么？

✎ 机器学习在数据挖掘，图像识别，语音和自然语言处理中有着广泛应用



资料推荐

1. Google机器学习速成课程

<https://developers.google.cn/machine-learning/crash-course/>

2. Google机器学习讲座

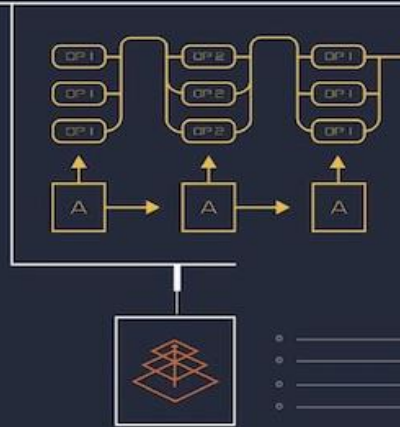
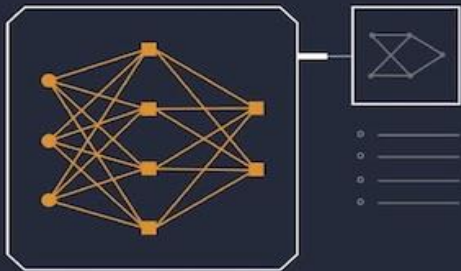
<https://www.youtube.com/watch?v=cKxRvEZd3Mw&index=1&list=PLT6eIRN3Aer7ncFlaCz8Zz-4B5cnsrOMt>

3. 莫烦的一系列讲座

<https://morvanzhou.github.io/tutorials/machine-learning/tensorflow/>

4. 优达的机器学习课程

<https://www.youtube.com/watch?v=ICKBWIkfeJ8&index=1&list=PLAwXTw4SYaPkQXg8TkVdlvYv4HfLG7SiH>



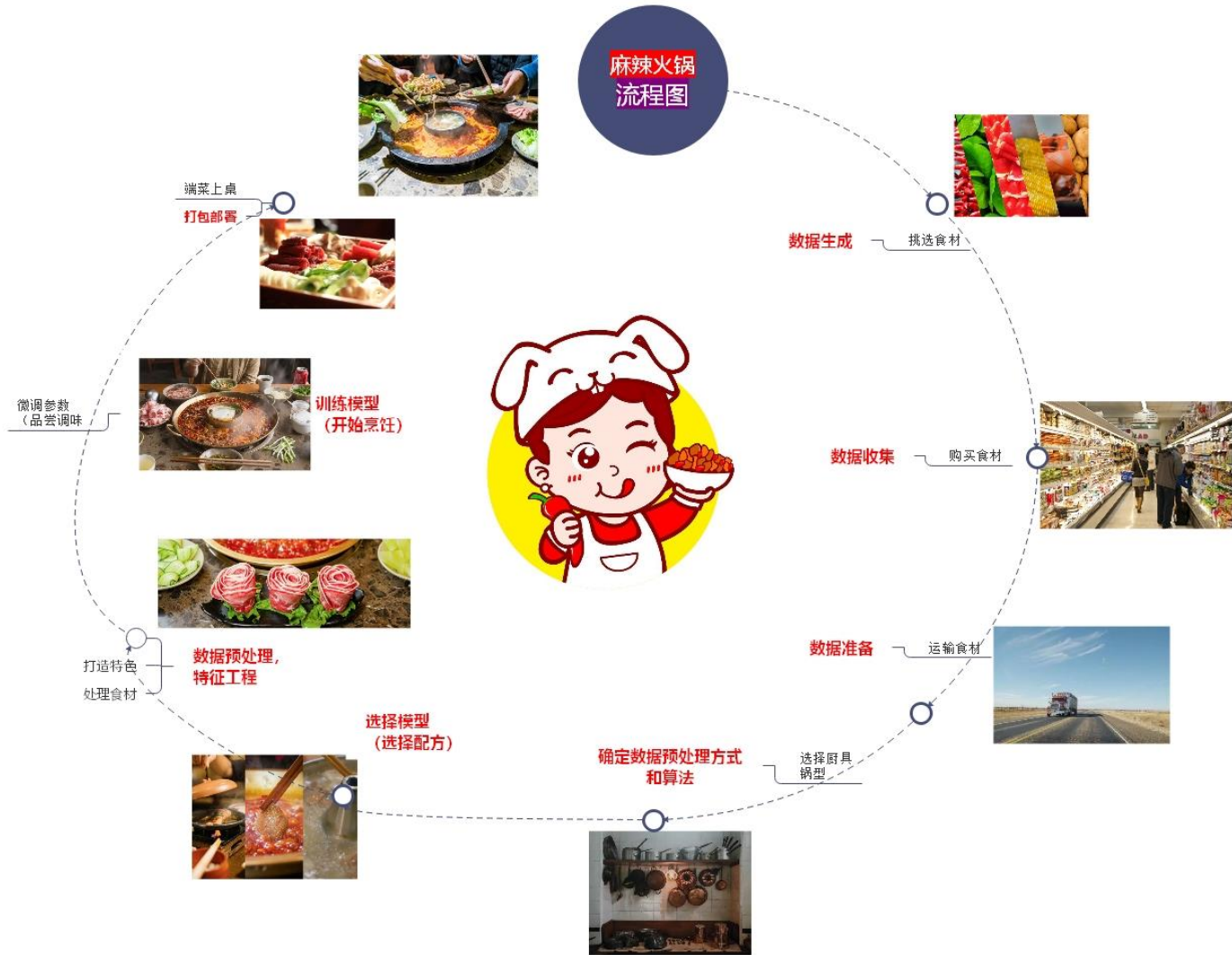
TensorFlow

机器学习原理与基础

机器学习的原理

火锅里的机器学习之道

图片来自：
Daniel Godoy
<https://towardsdatascience.com>



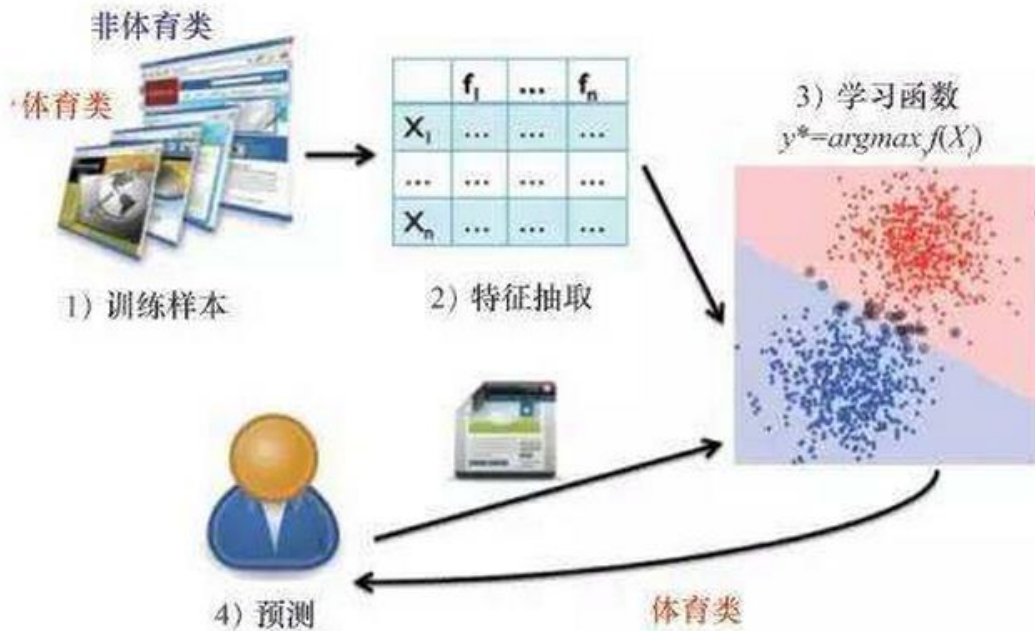
机器学习流程?

一个机器学习的常规套路

1. 数据收集与预处理

2. 特征选择与模型构建

3. 评估与预测





机器学习怎么动手去做？



只有实际应用啦，才觉得没白学，那么去哪里找案例呢？
最好的资源：Github，kaggle，各大资源分享点



案例的积累作用很大，其实我们干活是什么样的呢？主要就是在模仿，我们并不是科学家，能做事才能有用的，既然人家是这么做的，并且做的不错，那我们去模仿做出来的就是我们自己的！

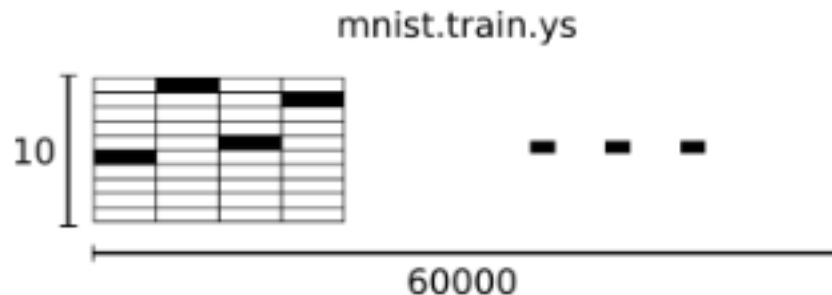
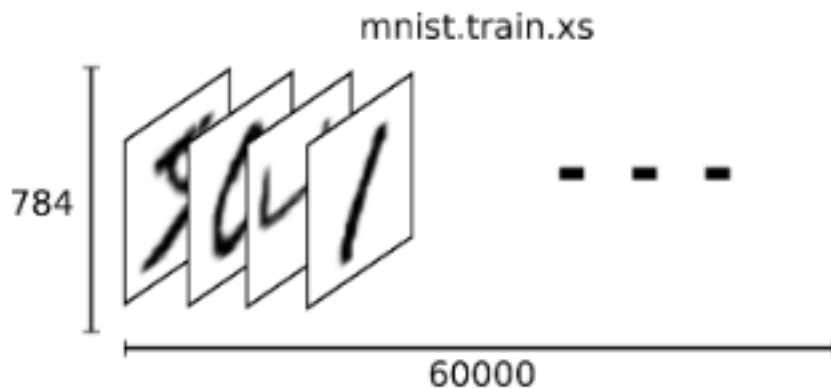
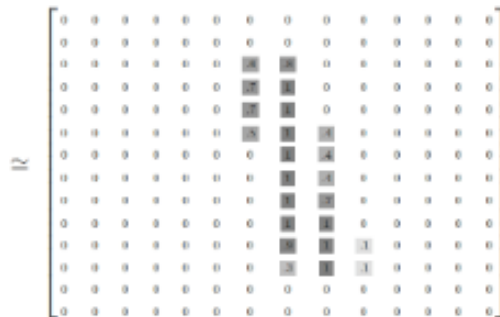
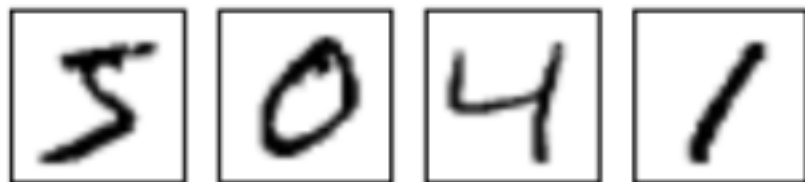


很少从头开始去写一个项目，通常都是按照之前的某种套路照搬过来，实际上大部分公司都这么做，建议大家先学会模仿，再去创作吧！

MNIST数据集

MNIST数据集

http://www.tensorfly.cn/tfdoc/tutorials/mnist_beginners.html





MNIST数据集

-----数据的收集与整理



内容提要

-----数据收集与整理


-
- 环境搭建
 - **Python**入门
 - 数据收集与整理




为嘛Python这么火?



 难度：有任何语言基础学Python你会觉得简单到家了

 实用性：能用一行代码，何必用十行呢？
(我们的定位通常都是工程师，而非科学家，能干活更重要)

 Python工具库：这些简直太多了，基本上你能想到的现在都有了
(我们常用的：Numpy, Pandas, Matplotlib, Scikit-Learn, tensorflow)

 大家都在用，各大公司开源工具库都有Python接口，并且都是主流，我们实际干活很大程度上都是使用这些库帮助我们完成任务



环境搭建



✓ Python和这些库怎么配置?


✎ 常规套路：1.安装 2.环境变量 3.用什么库去看配置文档自己安
(这个是程序员的常规套路了，但是刚转行的同学会弄个大半天)


✎ 大礼包：🍏 **ANACONDA**，傻瓜式安装，解决上述问题
(还给我们赠送了notebook写代码环境，库安装工具等)

✓ 如何学Python呢?



 有其他语言基础：简单过一遍语法，直接上手应该没什么问题

 第一门语言：初级内容边学边写，高级内容暂时不需要
(我觉得语言只是一门工具，不需要先都学彻底了才能干活，边做事边学习应该会更加深理解，忌光看书不练习)

 Python这么多库，这么多函数，我需要都记下来吗？
(其实这些依旧是工具，记它干嘛？我觉得重点在于知道每个库能做什么，大概用哪个函数，等实际用的时候还是需要查一查API的，这么多参数都记下来不太切合实际，动手查的能力也很重要)



Python入门

<http://www.runoob.com/python/python-tutorial.html>



- 行和缩进
- 编码的多行显示
- 注释

Python 基础教程



```
class MyFirstDemo:
```

```
#自定义一个输出时间的方法
```

```
def showTime():
```

```
#创建一个变量获取系统当前日期
```

```
current_date = time.strftime('%Y-%m-%d', time.localtime())
```

```
#屏幕输出
```

```
print u'当前日期: ' + current_date
```

```
#创建一个主启动函数
```

```
if __name__ == '__main__':
```

```
#调用showTime方法
```

```
showTime()
```

```
pass
```

```
print u'程序结束'
```

```
if True:
```

```
print u'第一行输出'
```

```
print u'第二行输出'
```

```
else:
```

```
print u'第三行输出'
```

```
print '第四行输出'
```

```
1  #-*- coding: utf-8 -*-
2  __author__ = "中软国际教育科技.CTO办公室"
3
4  #多行代码使用演示
5  num1 = 10
6  num2 = 20
7  num3 = 30
8  #源代码:
9  #total = num1 + num2 + num3
10 total = num1 + \
11         num2 + \
12         num3
13 print u'总和:', total
```



类和方法的 多行注释

```

1  #-*- coding: utf-8 -*-
2  #文件名: ch02-demo04.py
3
4  __author__ = "中软国际教育科技-CTO办公室"
5
6  #导入时间模块库
7  import time
8
9  '''
10  名称: DateTools
11  作用: 实现输出系统当前日期
12  参数: 暂无
13  '''
14  class DateTools(object):
15
16      """docstring for DateTools"""
17      def __init__(self, arg):
18          #调动父类初始化方法
19          super(DateTools, self).__init__()
20          self.arg = arg #获取类初始化参数并赋值 self.args 类成员变量
21
22      '''
23      方法名称: getDate
24      方法参数: 暂无
25      返回值: 格式化日期字符串
26      '''
27      def getSysDate():
28          #创建一个变量获取系统当前日期
29          current_date = time.strftime('%Y-%m-%d', time.localtime())
30          #返回系统日期
31          return current_date
    
```

代码中的 单行注释

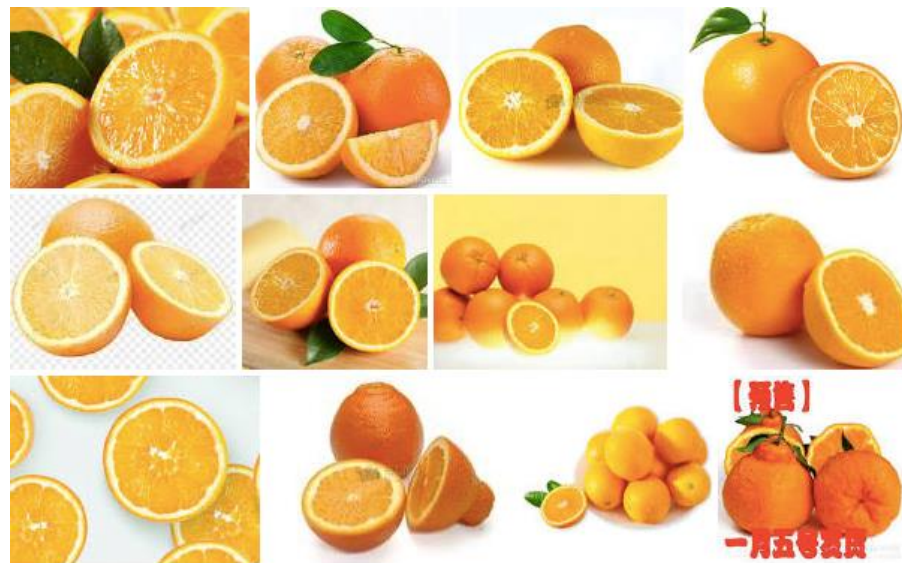
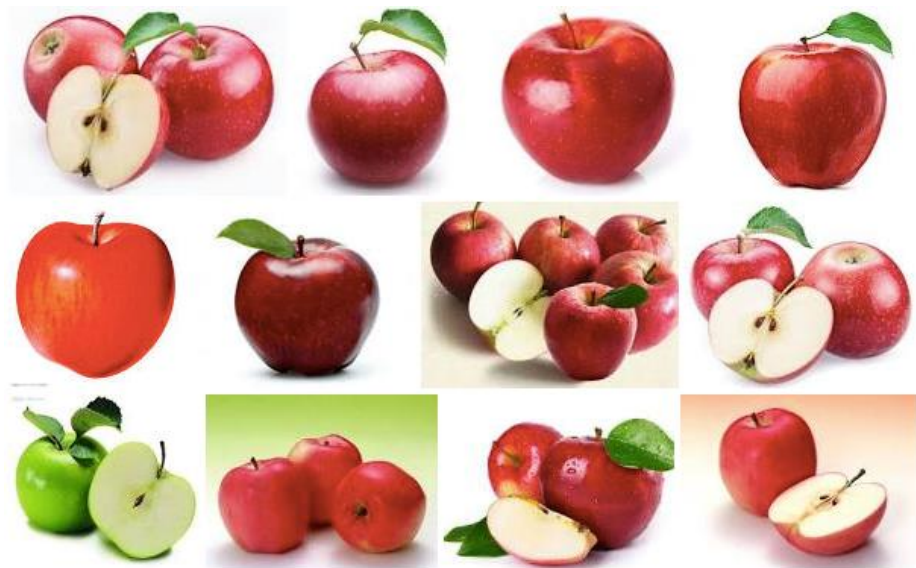

```
1 import numpy as np
2 def print_array_info(arr):
3     '''
4     打印传入的数组各项属性
5     '''
6     if isinstance(arr,np.ndarray):
7         print("\n-----")
8         print("数据类型",type(arr))
9         print("数组元素数据类型: ",arr.dtype)
10        print("数组元素的总数: ",arr.size)
11        print("数组的形状: ",arr.shape)
12        print("数组的维度: ",arr.ndim)
13        print("-----\n")
14    else:
15        print("不是numpy.ndarray")
16    return
```

https://blog.csdn.net/weixin_41770169/article/details/80565326

特征选择

模型构建、训练

特征



特征

{“苹果”, “橙子”}

{1,0}

{0.95,0.05}

{0.3,0.7}

模型训练中的过拟合和欠拟合

- 过拟合 (overfitting) : 当某个模型在经过训练后, 过度的学习了训练集的噪音和干扰项, 从而导致在测试集上表现很差, 但是在训练集上表现很好, 就称为过拟合。
- 欠拟合 (underfitting) : 值模型无论在测试集还是训练集上表现都不好, 也就是说训练样本不够。

模型构建与训练

Figure 1

[0. 0. 0. 0. 0. 0. 1. 0. 0. 0.]



```
15 # 搭建模型
16 x = tf.placeholder(tf.float32,[None,784],name="input_x")
17 W = tf.Variable(tf.zeros([784,10]),name="weight")
18 b = tf.Variable(tf.zeros([10]),name="bias")
19
20 y = tf.nn.softmax(tf.matmul(x,W)+b,name="softmax")
21 y_ = tf.placeholder(tf.float32,[None,10],name="input_y")
22
23 ## 计算交叉熵
24 cross_entropy = tf.reduce_mean(-tf.reduce_sum(y_ * tf.log(y),reduction_indices=[1]),name="cross_entropy")
25
26 # 用梯度下降算法以0.01的学习速率最小化交叉熵
27 train_step = tf.train.GradientDescentOptimizer(0.01).minimize(cross_entropy)
28
29 ## 初始化一个session
30 sess = tf.InteractiveSession()
31
32 ##启动模型
33 tf.global_variables_initializer().run()
34
35 # 开始训练模型 让模型循环训练1000次
36 for i in range(1000):
37     batch_xs,batch_ys = mnist.train.next_batch(100)
38
39     sess.run(train_step,feed_dict={x:batch_xs,y_:batch_ys})
40
41 print("训练完成...")
```

Run: mnist_train_test

-0.09776217 0.64133996 -1.4413497 -0.25590855]

测试结果:
正确率为91.91 %
计算模型结果成功!
预测结果:6

模型构建测试与保存

File Edit View Navigate Code Refactor Run Tools VCS Window Help

TJGDGdemo mnist_train_test.py

Project TJGDGdemo E:\Google\TJGDG

- MNIST_data
- mnist_web_xc
- model_web
- saved_model
 - variables
 - saved_model.pb
- venv
 - Include
 - Lib
 - Scripts
 - pyenv.cfg
 - test_tensorflow.py
- mnist_explore.py
- mnist_test.py
- mnist_train_test.py
- web API.docx

External Libraries

Scratches and Consoles

```
48 print("正确率为%.2f"%(sess.run(accuracy, feed_dict={x: mnist.test.images, y_: mnist.test.labels})*100),'  
49  
50 # 测试  
51 # 从mnist数据集取出一个测试图片  
52 idx=np.random.randint(0,6000)  
53 img = mnist.test.images[idx]  
54  
55 img = img.reshape(1,784)  
56 ret = sess.run(y, feed_dict={x: img})  
57 print("计算模型结果成功!")  
58 # 显示测试结果  
59 print("预测结果:%d"%(ret.argmax()))  
60  
61 # 可视化测试集中的图片  
62 import matplotlib.pyplot as plt  
63 # % matplotlib inline  
64 img = img.reshape(28,28)*255  
65 label = mnist.test.labels[idx]  
66 plt.title(label)  
67 plt.imshow(img,cmap=plt.get_cmap('gray_r'))  
68 plt.show()  
69  
70 # 保存模型  
71 saver = tf.train.Saver()  
72  
73 model_dir = "model_web"  
74 model_name = "savedModel"
```

Figure 1

[0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]



Run: mnist_train_test.py

测试结果:
正确率为91.83 %
计算模型结果成功!
预测结果:7

模型保存

```
61 # 可视化测试集中的图片
62 import matplotlib.pyplot as plt
63 # # % matplotlib inline
64 img = img.reshape(28,28)*255
65 label = mnist.test.labels[idx]
66 plt.title(label)
67 plt.imshow(img,cmap=plt.get_cmap('gray_r'))
68 plt.show()
69
70 # 保存模型
71 saver = tf.train.Saver()
72
73 model_dir = "model_web"
74 model_name = "savedModel"
75 # model_name = "ckp1021"
76 if not os.path.exists(model_dir):
77     os.mkdir(model_dir)
78 # 保存js模型
79 tf.saved_model.simple_save(sess, os.path.join(model_dir, model_name), inputs={"x": x, },
80                               outputs={"softmax": y, })
81 print("模型保存成功")
82
```

File Explorer (Left):

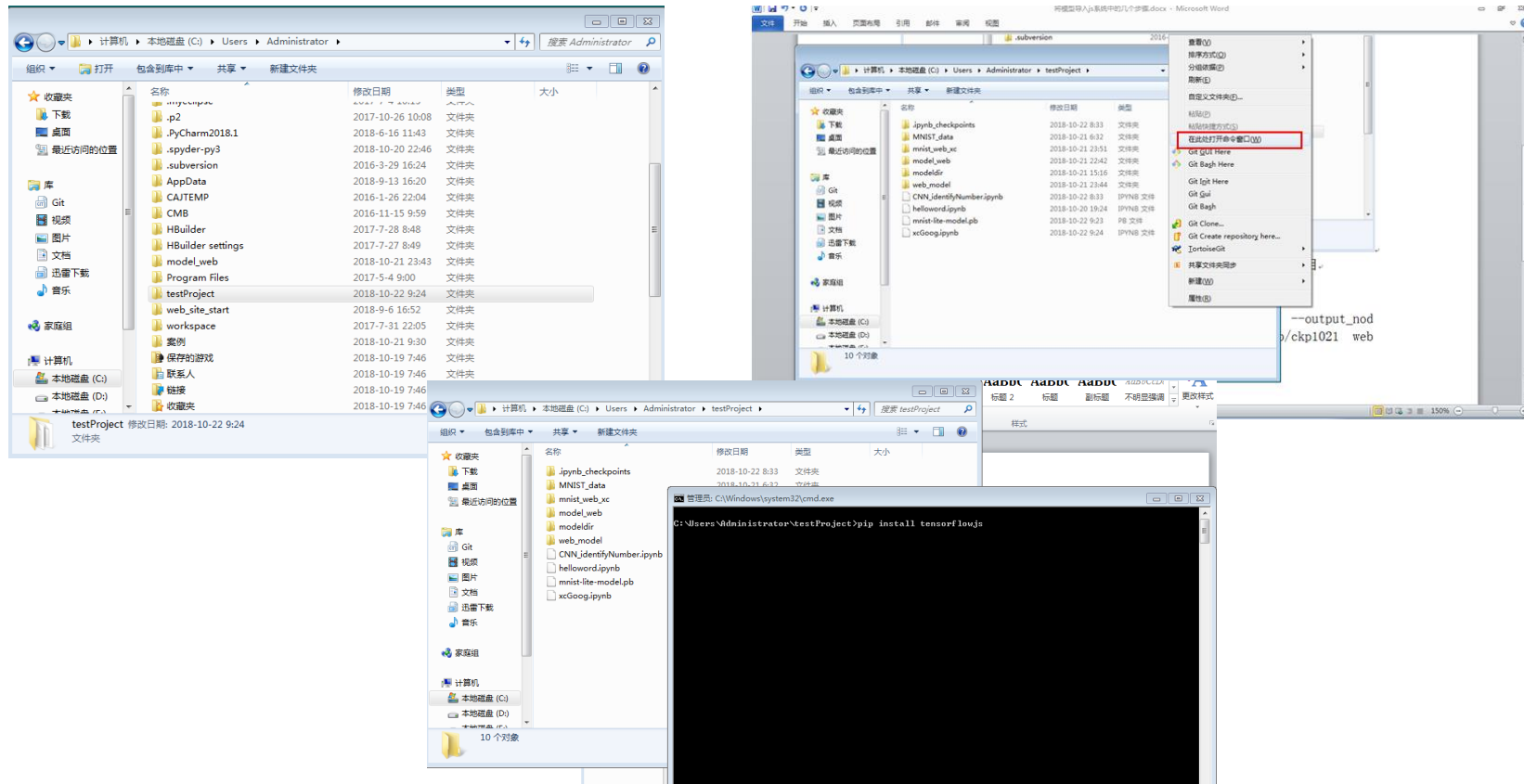
- TJGDGdemo E:\Google\TJG
 - MNIST_data
 - mnist_web_xc
 - model_web
 - saved_model
 - variables
 - saved_model.pb
 - venv
 - Include
 - Lib
 - Scripts
 - pyvenv.cfg
 - test_tensorflow.py
 - mnist_explore.py
 - mnist_test.py
 - mnist_train_test.py
 - web API.docx
 - External Libraries
 - Scratches and Consoles

模型的应用

--基于web端的应
用

如何调用模型? --步骤1

 pip install tensorflowjs



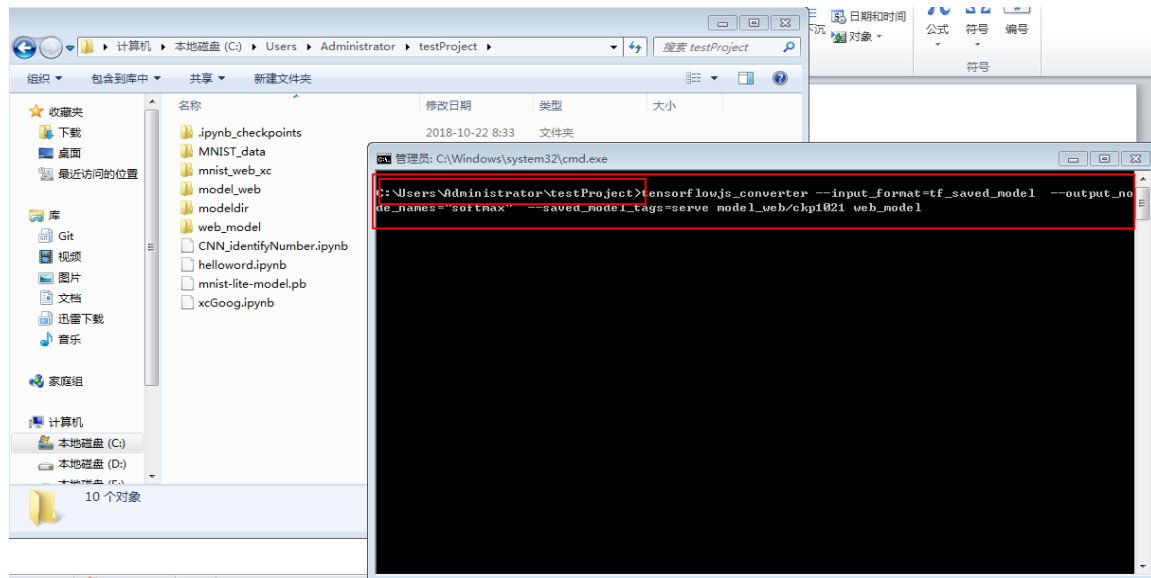
The image is a collage of four screenshots illustrating the steps to install tensorflowjs:

- Top-left:** A Windows File Explorer window showing the 'testProject' folder in the user's home directory.
- Top-right:** A Windows File Explorer window showing the contents of the 'testProject' folder. A context menu is open over the 'testProject' folder, with the option '在此处打开命令窗口(C)' (Open Command Window Here) highlighted.
- Bottom-left:** A Windows File Explorer window showing the contents of the 'testProject' folder, listing files like 'ipynb_checkpoints', 'MNIST_data', 'mnist_web_xc', 'model_web', 'modeldir', 'web_model', 'CNN_identityNumber.ipynb', 'helloworld.ipynb', 'mnist-lite-model.pb', and 'xcGoog.ipynb'.
- Bottom-right:** A Windows Command Prompt window showing the command 'C:\Users\Administrator>pip install tensorflowjs' being entered.

如何调用模型? --步骤2

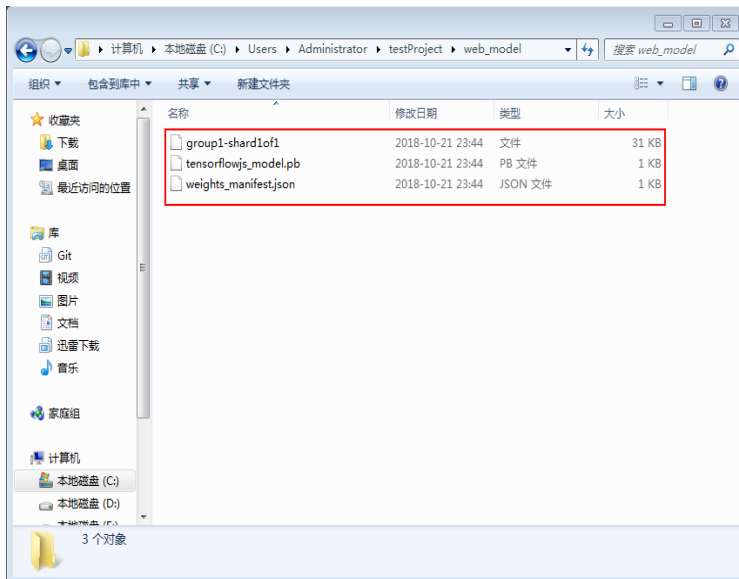
运行命令

```
tensorflowjs_converter --input_format=tf_saved_model --  
\output_node_names="softmax" --  
\saved_model_tags=serve model_web/saved_model web_model
```



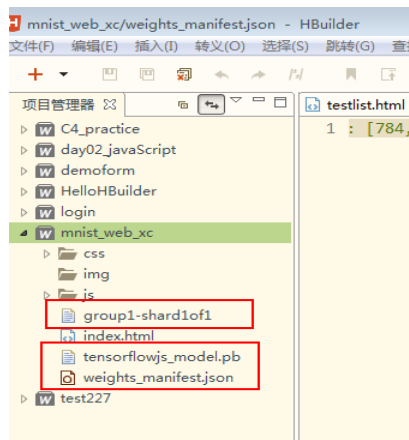
如何调用模型? --步骤2

- 运行成功后就在“testProject”目录下有一个“web_model”的文件夹生成，且里有3个文件，生成的文件如图所示：



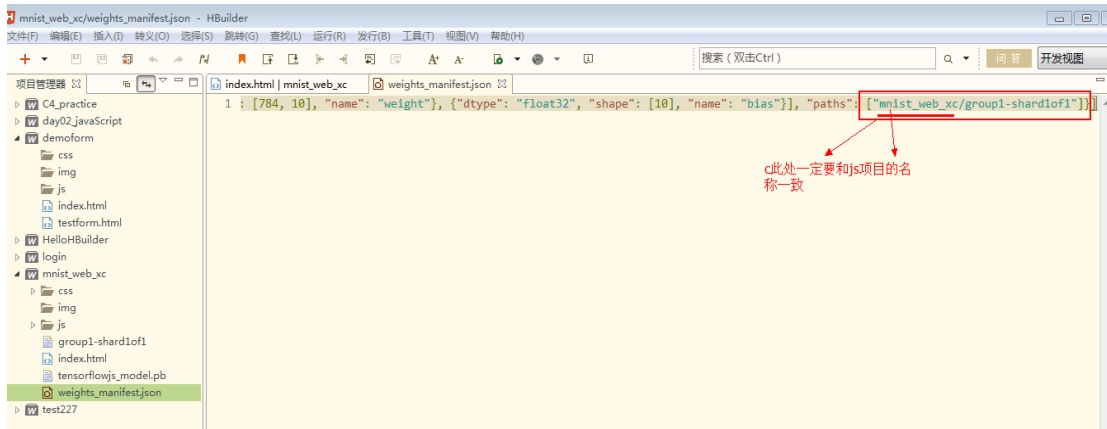
在js前端的工作--步骤3

(1) 在Hbuild中创建一个工程，如创建了一个工程“mnist_web_xc”后，然后将生成的图5中的那3个文件拷贝到mnist_web_xc”项目下的“js”文件夹中，以便于在js前端应用程序中调用保存的模型，如图所示。



在js前端的工作--步骤3

(2) 修改“weights_manifest.json”中的“path”值，一定要保证标注的地方与项目名称一致。



在js前端的工作--步骤3

(3) 自定义函数draw () 在js前端页面上实现手写字功能，同时注意图中标注的位置。

```
67 contexttemp.drawImage(draw.canvas, 0, 0, 28, 28);
68 //contexttemp.putImageData(data, 0, 0, 0, 28, 28);
69 var data1 = contexttemp.getImageData(0, 0, tempdraw.canvas.width, tempdraw.canvas.height);
70 //console.log(data1);
71 var resultarray = [];
72 for(var i = 0; i < 28; i++) {
73   var temparray = [];
74   for(var j = 1; j <= 28; j++) {
75     temparray.push(data1.data[i * 28 * 4 + j * 4 - 1] / 255.0);
76   }
77   resultarray.push(temparray);
78 }
79
80 var resultarray784 = [];
81 for(var i = 1; i <= 28 * 28; i++) {
82   resultarray784.push(data1.data[i * 4 - 1] / 255);
83 }
84
85 // 使用模型
86 const MODEL_URL = 'tensorflowjs_model.pb';
87 const WEIGHTS_URL = 'weights_manifest.json';
88
89 try {
90   const model = tf.loadFrozenModel(MODEL_URL, WEIGHTS_URL);
91   var xs = tf.tensor2d(resultarray784, [1, 784], tf.FLOAT_MAT32);
92
93   model.then(model => {
94     var output = model.execute({
95       'input_x': xs
96     });
97   });
98 }
```

项目管理器

draw.js

此处一定要定义一个常数

名字一定要与左侧的一致

```
79
80 var resultarray784 = [];
81 for(var i = 1; i <= 28 * 28; i++) {
82   resultarray784.push(data1.data[i * 4 - 1] / 255);
83 }
84
85 // 使用模型
86 const MODEL_URL = 'tensorflowjs_model.pb';
87 const WEIGHTS_URL = 'weights_manifest.json';
88
89 try {
90   const model = tf.loadFrozenModel(MODEL_URL, WEIGHTS_URL);
91   var xs = tf.tensor2d(resultarray784, [1, 784], tf.FLOAT_MAT32);
92
93   model.then(model => {
94     var output = model.execute({
95       'input_x': xs
96     });
97     var max = output.max().dataSync()[0];
98     var outputarr = output.dataSync();
99     //output.print();
100     outputarr.forEach((item, index) => {
101       var texts = document.querySelectorAll('.result+div');
102       var elements = document.querySelectorAll('.result');
103       texts[index].innerHTML = new Number(item * 100).toFixed(2) + "%";
104       if(max === item) {
105         elements[index].classList.value += " result-red";
106       }
107     });
108   });
109 }
```

项目管理器

draw.js

加载固定模型

模型使用

模型定义中的参数，一定要与模型定义中的一致

得到模型返回值的最大值

识别效果

127.0.0.1:8020/mnist_web_xc/index.html?_hbt=1553304180235



MNIST based on tensorflow

这是一个应用tensorflow demo



4

清除

提交