什么是 TensorFlow

1. 是一个编程系统（建立在其他高级语言之上）
2. 是一个科学计算工具
3. 是一个 Python 库（也有 C/C++支持，但是 Python 是最友好的）
4. 特别适合深度学习（但是不限于）
5. 由 Google 开源

TF 基础概念

1. 用图（Graph）来表示计算任务
2. 在会话（Session）的上下文（Context）中执行图
3. 使用 tensor 表示数据
4. 通过变量（Variable）维护状态信息
5. Feed 和 fetch 为操作赋值或从中获取数据

写 TF 代码的步骤

1. 脑子里有一个计算图
2. 把你设想的图告诉计算机，当然需要用计算机听的懂的语言（coding: give instructions to your machine）
3. Let it run
4. 投喂数据，取出数据

从两个简单的例子开始

Hello World!

仪式性的惯例

初识 Session

矩阵乘法

A * B：

A = ？

B = ?

C = A * B

Given Matrix (A, B, D), C = A*B, E = C*D，output E

如果使用传统编程语言，如何解决这个问题？

Input A， B， D

C = A * B

E = C * D

Output E

使用 TensorFlow：构造图、执行计算（图）的过程严格区分

代码

https://github.com/AlexSun1995/TensorFlow_test/blob/master/matrix_mult.py

回到 TF 的重要概念

1. TensorFlow 用图来表示计算任务，图中的节点被称之为 operation，缩写成 op
2. 一个节点获得 0 个或者多个张量 tensor，执行计算，产生 0 个或多个张量
3. 图必须在会话(Session)里被启动，会话(Session)将图的 op 分发到 CPU 或 GPU 之类的设备上，同时提供执行 op 的方法，这些方法执行后，将产生的张量(tensor)返回

TF 的优势

第一，基于 Python，写的很快并且具有可读性。

第二，在多 GPU 系统上的运行更为顺畅。

第三，代码编译效率较高。

第四，社区发展的非常迅速并且活跃。

第五，能够生成显示网络拓扑结构和性能的可视化图。

TF 的工作原理

TensorFlow 是用数据流图(data flow graphs)技术来进行数值计算的。

数据流图是描述有向图中的数值计算过程。

有向图中，节点通常代表数学运算，边表示节点之间的某种联系，它负责传输多维数据(Tensors)。

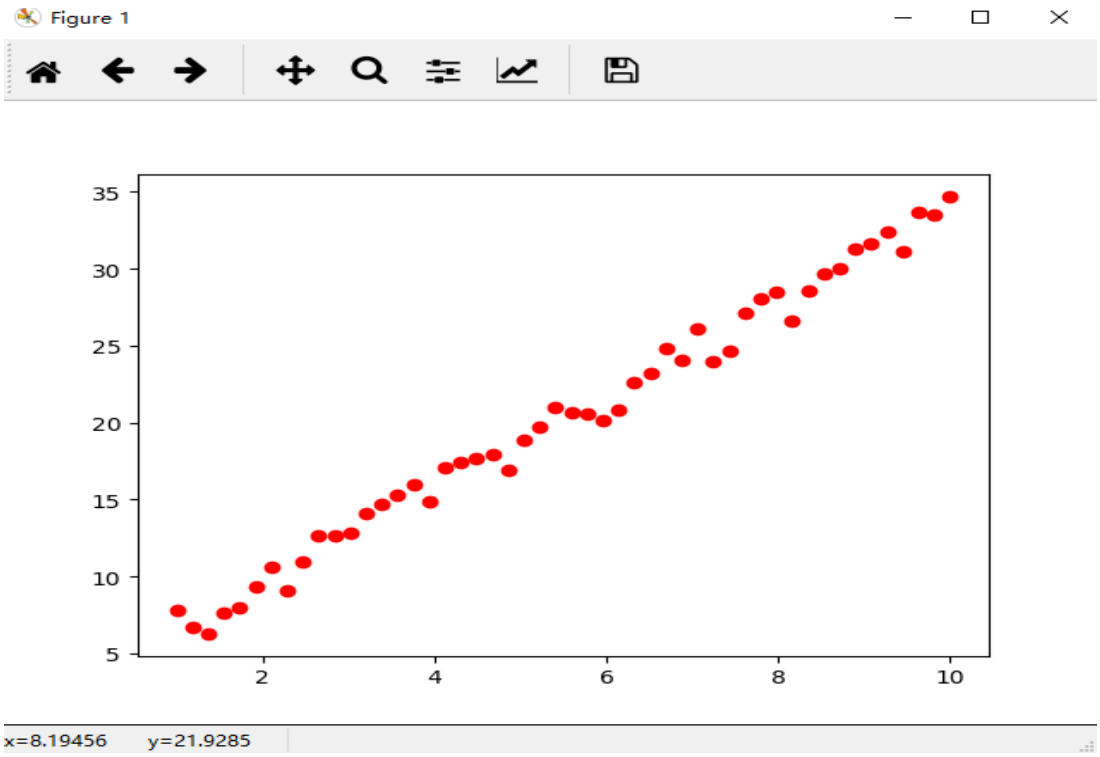
节点可以被分配到多个计算设备上，可以异步和并行地执行操作。因为是有向图，所以只有等到之前的入度节点们的计算状态完成后，当前节点才能执行操作。

TF 使用进阶 1： 线性回归

问题描述

现有一组数据(xi, yi) 预测 $y_i \approx wx_i + b$

使用数据集，训练参数 w, b 的值



思考

定义一个代价函数 *cost*

$$cost = \frac{1}{2n}\sum (yi - wxi - b)^2$$

代价函数说明了预测模型的偏差，代价函数的值越小，则说明参数的值越准确。w, b 是 cost 函数的变量，求 cost 最小值的过程，就是求w, b 最优解的过程。

编码

https://github.com/AlexSun1995/TensorFlow_test/blob/master/linerar_regression.py

踩坑

X, Y 是占位符

W, b 是变量

优化 cost 的过程自动优化变量的值（cost 定义了依赖关系）

含有变量的图，一定要将变量初始化，TensorFlow 变量的初始化不同于 C 语言，仍然要用 Session

投喂的数据和占位符（X,Y）一致，投喂数据的形状（张量的形状）必须和占位符定义的形状一致

养成显示定义数据类型的习惯，否则可能会造成未知错误

小结

TensorFlow 是数据驱动的，你只需要定义好数据的依赖关系，不需要考虑 TensorFlow 是如何优化的，

不需要写一句求偏导的代码，你只需要让优化器（Optimizer）跑起来，一切都是自动完成的

TF 使用进阶 2 训练神经网络

数据集介绍

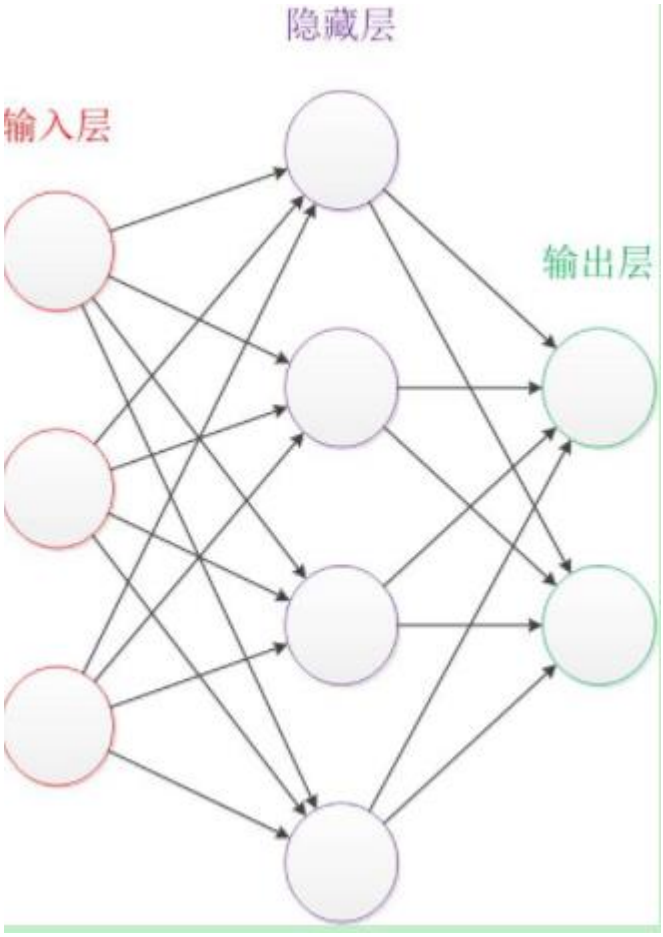
Iris 数据集是常用的分类实验数据集，由 Fisher, 1936 收集整理。Iris 也称鸢尾花卉数据集，是一类多重变量分析的数据集。数据集包含 150 个数据集，分为 3 类，每类 50 个数据，每个数据包含 4 个属性。可通过花萼长度，花萼宽度，花瓣长度，花瓣宽度 4 个属性预测鸢尾花卉属于（Setosa，Versicolour，Virginica）三个种类中的哪一类。

任务

根据四个维度的数据，构建一个简单的神经网络模型，实现一个鸢尾花卉类型分类器。

实现方法

Iris 数据集一共有 150 条数据，按照 50，50，50 的顺序，分别代表 0,1,2 三种花的数据。数据格式为：
Feature1, feature2, feature3, type
我们在 3 个种类的数据中分别各取 25 个，组成 75 条数据作为训练数据集，剩下的则作为测试数据集



一个神经网络的例子

输入层 :X, 4 个维度的特征数据 (长度为 4 的特征向量), 长度记为 I

隐藏层：（节点数可变）无实际意义，长度记为 H（神经元的个数）

输出层： Y, 长度为 3 的向量，数值表示预测值：输入的数据对应响应种类花的预测概率。

例如，如果输出结果为[0.1, 0.8, 0.1]则表明我们的算法

根据特征数据预测属于种类 0 的概率是 0.1， 属于种类 1 的概率是 0.8， 属于种类 2 的概率是 0.1。也就是说，算法预测这朵花有 80%的概率是第 1 类花。

模型解释

$$\mathrm{X} = \mathrm{matrix}[batch\_size][\mathrm{I}]$$

$$\mathrm{b}_1 = matrix([H])$$

$$\mathrm{W}_1 = \mathrm{matrix}[I][\mathrm{H}]$$

$$\mathrm{b}_2 = matrix([O])$$

$$\mathrm{W}_2 = \mathrm{matrix}[\mathrm{H}][\mathrm{O}]$$

$$\mathrm{Y}_1 = sigmoid(X * W_1 + b_1)$$

$$\mathrm{Predicted}_\mathrm{y} = sotfmax(Y_1 * W_2 + b_2)$$

Think: output shape?

如何来评价模型？ 在这个问题中，变量是

$$\mathrm{w}_1, w_2, b_1, b_2$$

如果预测出来的结果和真实的结果更接近，当然就可以说这个模型更好

举例： 如果

预测输出 0.1， 0.0， 0.9

实际结果 0， 0， 1

我们就可以说，这个预测结果和实际的结果很接近了

问题是， 如何用数学方法，定量地描述这种偏差？我们在这个问题中使用交叉熵。

$$\text{entropy} = -\frac{1}{n}\sum_{i}^{n} \log(predict_{yi}) * real_{yi}$$

这个 entropy 就可以理解是代价函数，代价函数的值越小，参数值就越合理

代码

https://github.com/AlexSun1995/TensorFlow_test/blob/master/neural_network.py

TF & DL 的一些参考资源：

1. TensorFlow 中文文档
   http://wiki.jikexueyuan.com/project/tensorflow-zh/how_tos/reading_data.html
2. Udacity 深度学习
https://classroom.udacity.com/courses/ud730/lessons

/6377263405/concepts/last-viewed

3. TensorFlow GitHub

https://github.com/tensorflow/tensorflow

4 Stanford CS224N

https://nlp.stanford.edu/courses/cs224n/2015/

5 Andrew Ng, Machine Learning

...