



山东大学  
SHANDONG UNIVERSITY

# 信息基础II 机器学习、模式识别和深度学习

## 实验一

陈雷

山东大学信息科学与工程学院

[lei.chen@sdu.edu.cn](mailto:lei.chen@sdu.edu.cn)

## 实验内容

实验一：常规神经网络函数逼近实验

实验二：LeNet-5

实验三：ResNet-18

实验四：Selective search

实验五：Yolo

小作业&期末项目



# 深度学习软件框架

1. TensorFlow

2. Caffe

3. MXNet

4. Torch

5. Theano

6. CNTK

7. Keras

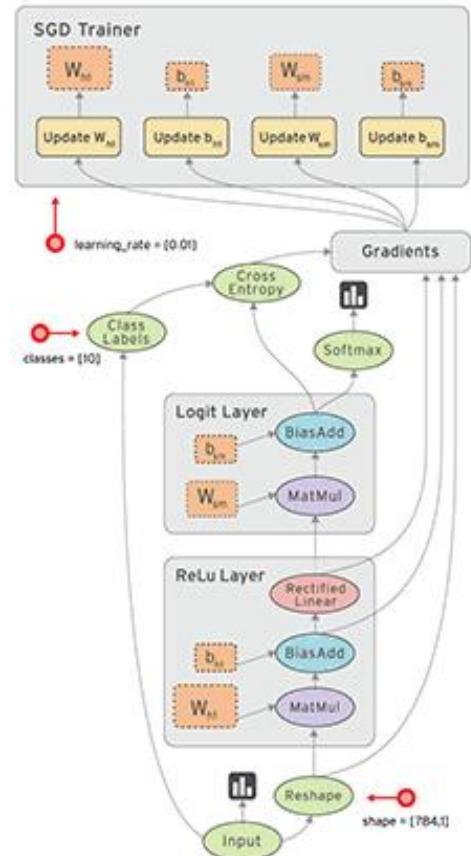


# TensorFlow

TensorFlow是一个采用数据流图，用于数值计算的开源软件库。

TensorFlow最初由Google大脑小组的研究员和工程师们开发出来，用于机器学习和深度神经网络方面的研究。

数据流图用“节点”（nodes）和“线”（edges）的有向图来描述数学计算。“节点”一般用来表示施加的数学操作，但也可以表示数据输入（feed in）的起点/输出（push out）的终点。“线”表示“节点”之间的输入/输出关系。这些数据“线”可以输运“size可动态调整”的多维数据数组，即“张量”（tensor）。张量从图中流过的直观图像是这个工具取名为“TensorFlow”的原因。一旦输入端的所有张量准备好，节点将被分配到各种计算设备完成异步并行地执行运算。



# TensorFlow的特征

## 高度的灵活性

TensorFlow 不是一个严格的“神经网络”库。只要你可以将你的计算表示为一个数据流图，你就可以使用Tensorflow。你来构建图，描写驱动计算的内部循环。我们提供了有用的工具来帮助你组装“子图”（常用于神经网络），当然用户也可以自己在Tensorflow基础上写自己的“上层库”。定义顺手好用的新复合操作和写一个python函数一样容易，而且也不用担心性能损耗。当然万一你发现找不到想要的底层数据操作，你也可以自己写一点c++代码来丰富底层的操作。

## 自动求微分

基于梯度的机器学习算法会受益于Tensorflow自动求微分的能力。作为Tensorflow用户，你只需要定义预测模型的结构，将这个结构和目标函数（objective function）结合在一起，并添加数据，Tensorflow将自动为你计算相关的微分导数。计算某个变量相对于其他变量的导数仅仅是通过扩展你的图来完成的，所以你能一直清楚看到究竟在发生什么。

## 真正的可移植性 (Portability)

Tensorflow 在CPU和GPU上运行，比如说可以运行在台式机、服务器、手机移动设备等等。想要在没有特殊硬件的前提下，在你的笔记本上跑一下机器学习的新想法？Tensorflow可以办到这点。准备将你的训练模型在多个CPU上规模化运算，又不想修改代码？Tensorflow可以办到这点。想要将你的训练好的模型作为产品的一部分用到手机app里？Tensorflow可以办到这点。你改变主意了，想要将你的模型作为云端服务运行在自己的服务器上，或者运行在Docker容器里？Tensorflow也能办到。Tensorflow就是这么拽：)

## 将科研和产品联系在一起

过去如果要将科研中的机器学习想法用到产品中，需要大量的代码重写工作。那样的日子一去不复返了！在Google，科学家用Tensorflow尝试新的算法，产品团队则用Tensorflow来训练和使用计算模型，并直接提供给在线用户。使用Tensorflow可以让应用型研究者将想法迅速运用到产品中，也可以让学术性研究者更直接地彼此分享代码，从而提高科研产出率。

## 多语言支持

Tensorflow 有一个合理的c++使用界面，也有一个易用的python使用界面来构建和执行你的graphs。你可以直接写python/c++程序，也可以用交互式的ipython界面来用Tensorflow尝试些想法，它可以帮你将笔记、代码、可视化等有条理地归置好。当然这仅仅是个起点——我们希望能鼓励你创造自己最喜欢的语言界面，比如Go, Java, Lua, Javascript, 或者是R。

## 性能最优化

比如说你又一个32个CPU内核、4个GPU显卡的工作站，想要将你工作站的计算潜能全部发挥出来？由于Tensorflow 给予了线程、队列、异步操作等以最佳的支持，Tensorflow让你可以将你手边硬件的计算潜能全部发挥出来。你可以自由地将Tensorflow图中的计算元素分配到不同设备上，Tensorflow可以帮你管理好这些不同副本。



## Torch

Torch是一个开源的机器学习框架，采用统一计算设备架构进行GPU交互。

Torch一直聚焦于大规模的机器学习应用，尤其是图像或者视频应用等领域。

Torch在使用方式非常简单的基础上最大化地保证算法的灵活性和速度。Torch的核心是流行的神经网络和简单易用的优化库，使用Torch能在实现复杂的神经网络拓扑结构的时候保持最大的灵活性。



## PyTorch

PyTorch的前身是Torch。由于Torch由Lua语言编写，开发者把Torch移植到Python语言中，形成了PyTorch。

PyTorch由Adam Paszke, Sam Gross, Soumith Chintala等人牵头开发，其成员来自Facebook FAIR和其他多家实验室。2017年1月，Facebook开源了PyTorch。

PyTorch支持卷积神经网络(CNN)、循环神经网络(RNN)以及长短期记忆网络(LSTM)，同时提供了大量的图片数据方便用户进行实验。



## TensorFlow

<https://tensorflow.google.cn/>

<http://doc.codingdict.com/tensorflow/index-2.html>

## PyTorch

<https://pytorch.org/>

<https://www.ptorch.com/>

<https://www.pytorchtutorial.com/>



# 实验一：常规神经网络函数逼近实验

## 问题描述：

函数逼近问题是指在不给出函数关系式的前提下，仅通过大量的函数值对应实例对神经网络进行训练，使得神经网络可以根据一个未知的自变量预测对应的应变量值。

## 可用神经网络：

反向传播（Back Propagation, BP）神经网络

径向基函数（Radial Basis Function, RBF）神经网络

反馈型Elman神经网络

小波神经网络

...



## 实验目的：

1. 了解基本的神经网络编程，基础的数据集准备和前向后向传播过程；
2. 学习神经网络训练方法；
3. 学习神经网络结构搭建方法。



## 实验要求:

1. 基本的神经网络（Basic Neural Networks）；
2. 应用三层神经网络进行函数逼近；
3. 函数包括 XOR, 和  $y=1/\sin(x) + 1/\cos(x)$ ；
4. 要求自己编程实现前向、后向网络结构；
5. 开发语言为**python**或C/C++；
6. 要求实现训练拟合，并且记录拟合过程，自己分析拟合的结果和评价情况；
7. 要求提交报告和代码。



以BP为例：

1. 编程实现BP神经网络算法；
2. 探究BP算法中学习因子收敛趋势、收敛速度之间的关系；
3. 修改训练后BP神经网络部分连接权值，分析连接权值修改前和修改后对相同测试样本结果，理解神经网络分布存储等特点。



## BP神经网络算法步骤和流程：

1. 初始化网络权值；
2. 由给定的输入输出模式对计算隐层、输出层各单元输出；
3. 计算新的连接权值；
4. 选取下一个输入模式，返回第2步反复训练直到网络输出误差达到要求结束训练。

