在Spark上构建分布式神经网络

王奕恒

Intel, Big Data Technology

Agenda

- 背景和概要介绍
- 通用计算平台上的性能优化

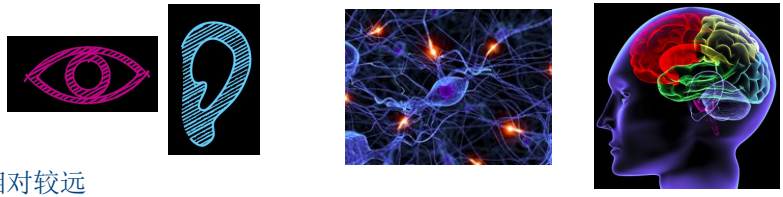
4

Agenda

- 背景和概要介绍
- 通用计算平台上的性能优化

神经网络 Artificial Neural Network

外部输入　　神经元　　认知



神经网络与神经科学的关系

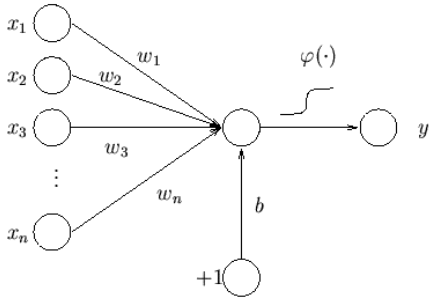
- Biological Inspiration
- 最初的研究动机： 解决问题/研究人脑
- 现代的“深度学习”和神经科学的关系已相对较远

Perceptron：最基础的神经元模型

- 本质上就是线性模型叠加非线性activation

向量内积

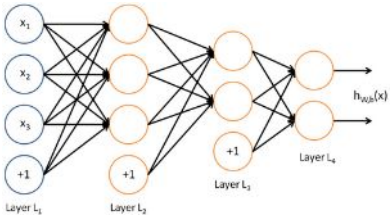
$$y = \varphi\left(\sum_{i=1}^{n} w_i x_i + b\right) = \varphi(\mathbf{w}^T\mathbf{x} + b)$$



MLP – Multilayer Perceptron

多个Perceptron组合成网络 (无环，feed-forward)

向量矩阵相乘



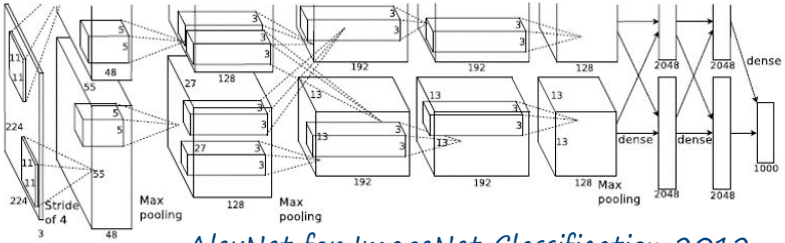
卷积神经网络 Convolutional Neural Network

一种特殊的神经网络形式——可以处理已知的、具有网格状拓扑结构的数据（根据卷积操作的特性)

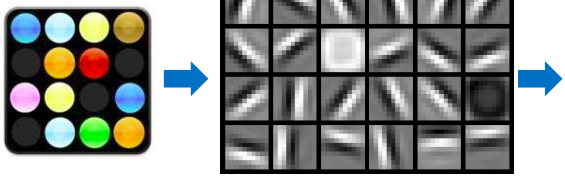
CNN在实践中已经有非常成功的应用，特别是在图像识别领域。

ImageNet image classification result

	Year	Error Rate
...	2011	>26%
AlexNet	2012	15.3%
GoogLeNet	2014	6.6%
State-of-art	2015	<5%

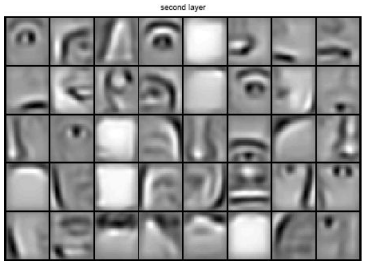


AlexNet for ImageNet Classification 2012

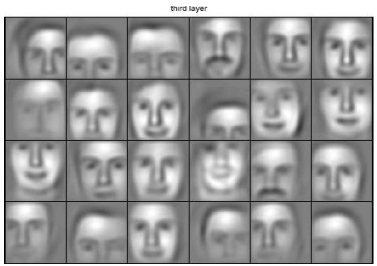


Pixels

Edges



Object Parts



Object models

Illustrations from Andrew Ng. CS229

Spark上神经网络的支持

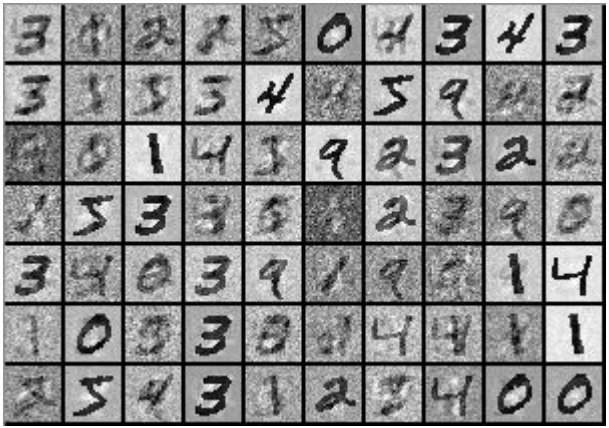
Spark 1.5开始支持

- MLP的实现在spark.ml（ spark.ml.MultilayerPerceptronClassifier ）
- 数据接口基于spark的DataFrame
- Layer, error, activation等基本功能在ml.ann.Layer
- Optimizer和Evaluation主要实现在mllib，和其他算法共享
- Some docs: http://spark.apache.org/docs/latest/ml-ann.html

MNIST 手写识别数据集

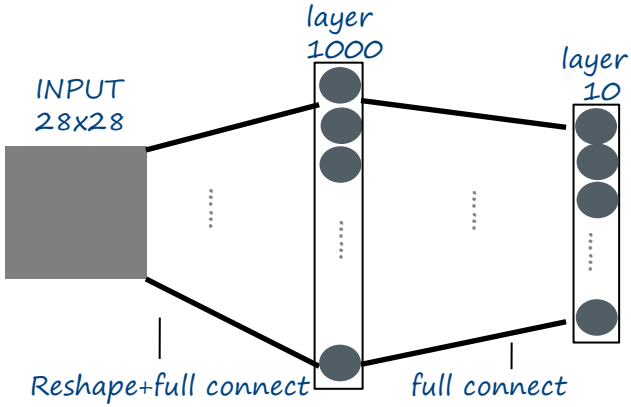
http://yann.lecun.com/exdb/mnist/

A training set of 60,000 examples, and a test set of 10,000 examples. The digits have been size-normalized and centered in a fixed-size image.

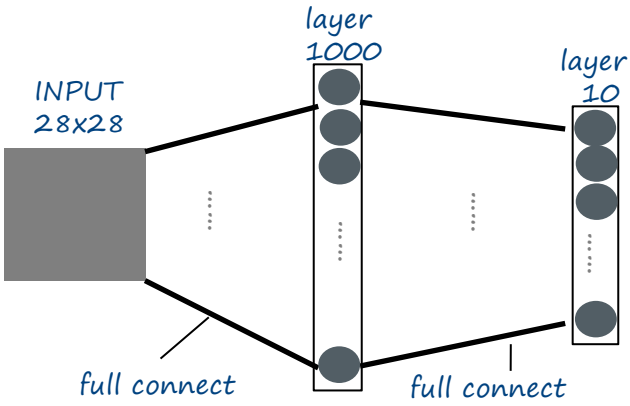


MLP用于MNIST

Vanilla MLP



Spark上神经网络的支持



Spark MLlib对应的代码实现

```
val topology = FeedForwardTopology.multiLayerPerceptron(Array(784, 1000, 10), false)
val initialWeights = FeedForwardModel(topology, 23124).weights()
val trainer = new FeedForwardTrainer(topology, 784, 10)
trainer.setWeights(initialWeights)
trainer.LBFGSOptimizer.setNumIterations(100)
val model = trainer.train(trainDataSet)
val output = model.predict(input)
```

我们的工作

和社区一起改进Spark上现有NN的实现

- CNN的支持
  - listed in Spark 1.6 roadmap (stretch)
  - Discussion on SPARK-5575, code @ https://github.com/hhbyyh/mCNN
- 完善NN的功能（更多的Layer和activation）
- 性能上的改进（硬件加速，训练/优化算法）
- 更多应用和benchmark（ImageNet等）

Agenda

- 背景和概要介绍
- 通用计算平台上的性能优化

一些和性能相关的数据

https://gist.github.com/hellerbarde/2843375

	时间	可感知的时间长度	类比
访问L1缓存	0.5 ns	0.5秒	一次心跳
分支预测失败	5 ns	5秒	打一个哈欠
访问L2缓存	7 ns	7秒	打一个长的哈欠
操作互斥锁	25 ns	25秒	准备一杯咖啡
访问主存	100 ns	100秒	刷牙
压缩1K数据	3 µs	50分钟	一集电视剧
通过千兆网卡传输2K数据	20 µs	5.5小时	半个工作日
随机访问SSD	150 µs	1.7天	一个周末
从内存中顺序读1M数据	250 µs	2.9天	元旦放假
同一机房内的数据传输	0.5 ms	5.8天	差不多一个国庆长假
从SSD顺序读1M数据	1 ms	11.6天	跨国邮寄
磁盘寻道	10 ms	16.5个星期	一个学期
从磁盘顺序读1M数据	20 ms	7.8个月	几乎可以生个小孩

知己知彼

- 面向瓶颈的性能优化
  1. 测试分析（Profiling）
  2. 测试分析（Profiling）
  3. 测试分析（Profiling）

重要的话说三遍！

- Spark上的分析工具
  - 系统层级：http://www.brendangregg.com/Perf/linux_observability_tools.png, Ganglia
  - 中间件：JVM工具(-Xprof, GC log…), Ganglia
  - 分布式框架：Spark GUI
  - 应用层级：打Log

影响性能的因素

- 分布式机器学习的特点
  - 不同节点的差异
  - 网络传输

- 其它要考虑的
  - GC
  - Cache
  - IO
  - 计算能力（SIMD, Xeon Phi,显卡，FPGA）
  - 多线程 

分布式神经网络算法的性能优化

- 训练算法层面优化
  - 减少网络传输
  - 减少同步

- 运算层面优化
  - 更高效的计算原语实现
    - MKL…
  - 额外的计算资源
    - Xeon Phi, FPGA, 显卡…

具体的例子

Vanilla MLP for 金融风控（真实数据）

- 训练集：300万
- 测试集：100万

集群配置

- Spark：1 master，4 worker
- 节点：Xeon E5，128G内存

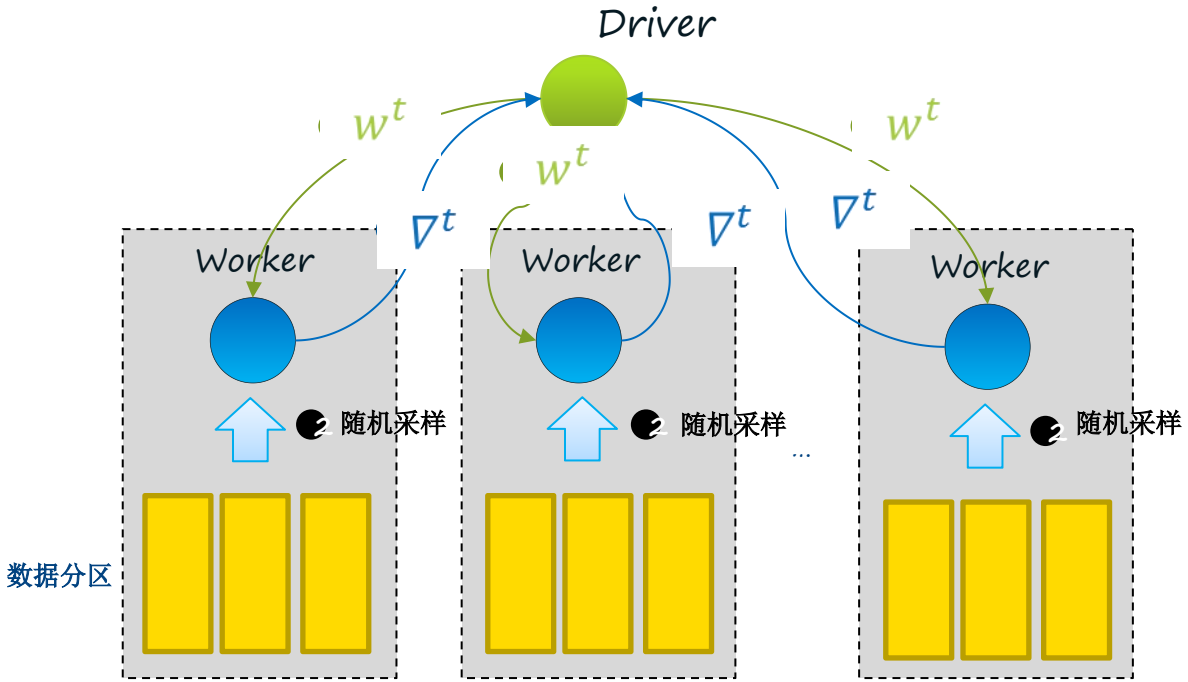
具体的例子 – 分布式训练算法

- Spark Mini-batch SGD
- Spark LBFGS
- Distributed SGD

具体的例子 – 分布式训练算法

- Spark Mini-batch SGD

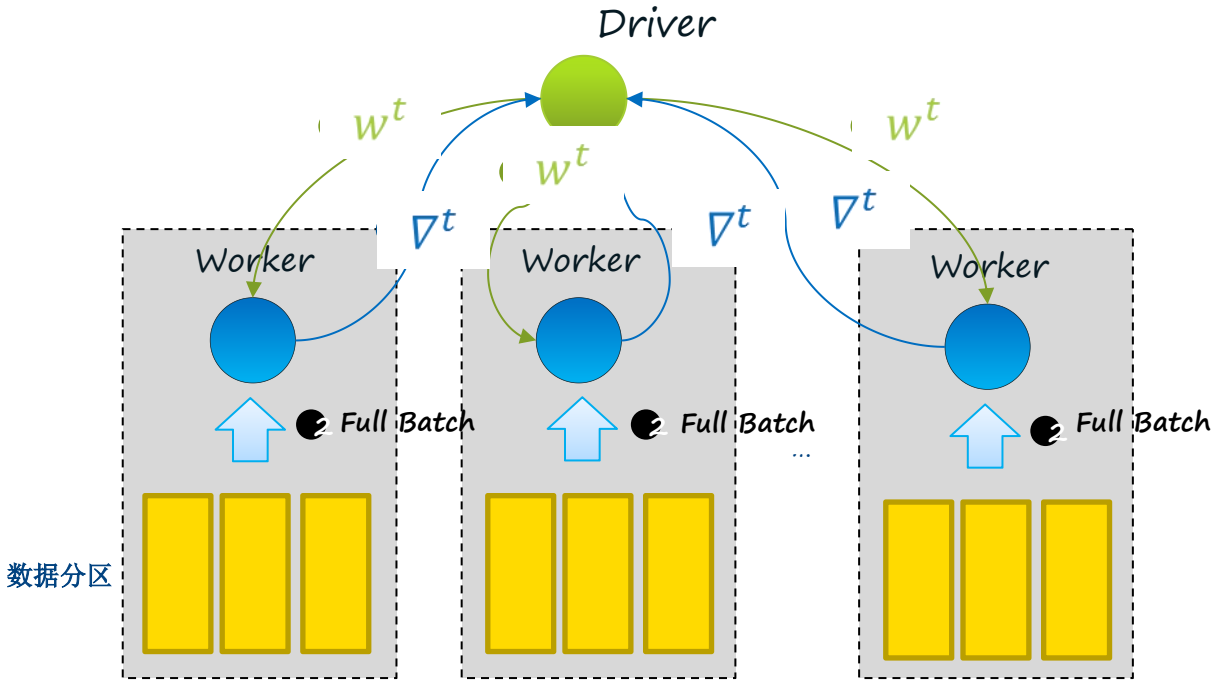
$$w^{t+1} = w^t - \alpha \cdot \nabla^t$$



具体的例子 – 分布式训练算法

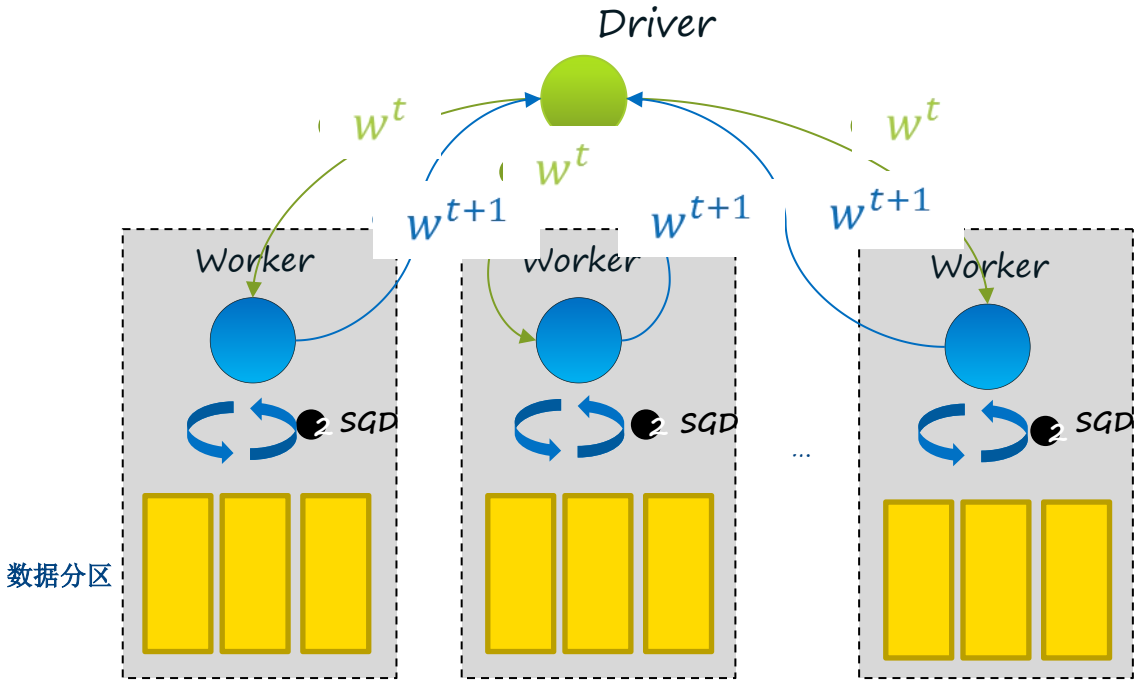
- Spark LBFGS

$$w^{t+1} = w^t - \alpha \cdot H^{-1} \cdot \nabla_t$$

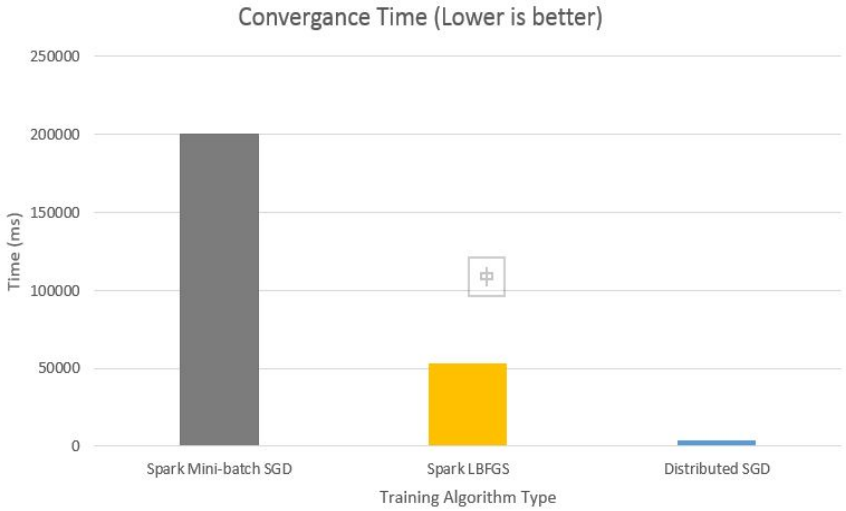


具体的例子 – 分布式训练算法

- Distributed SGD



具体的例子 – 训练算法对比

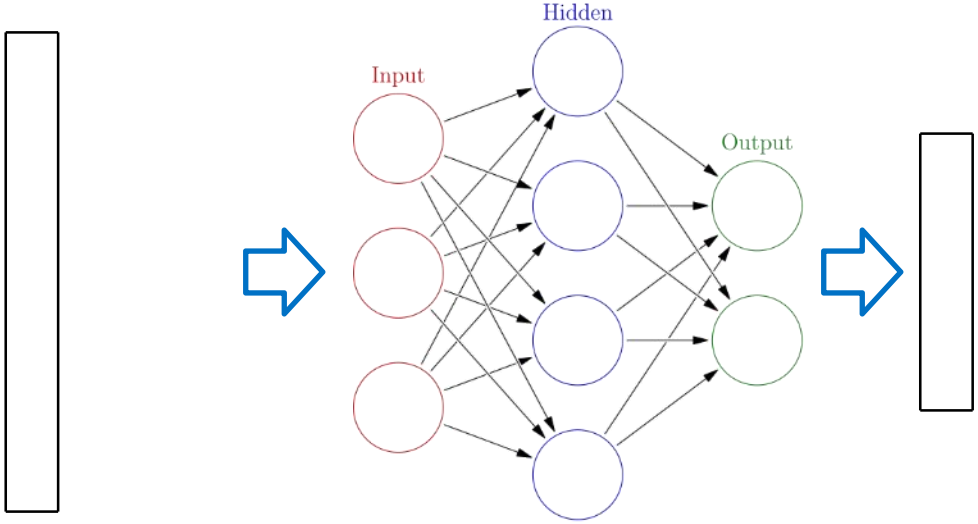


准确率比较

Spark Mini-batch SGD	Spark LBFGS	Distributed SGD
95.3%	96.6%	95.7%

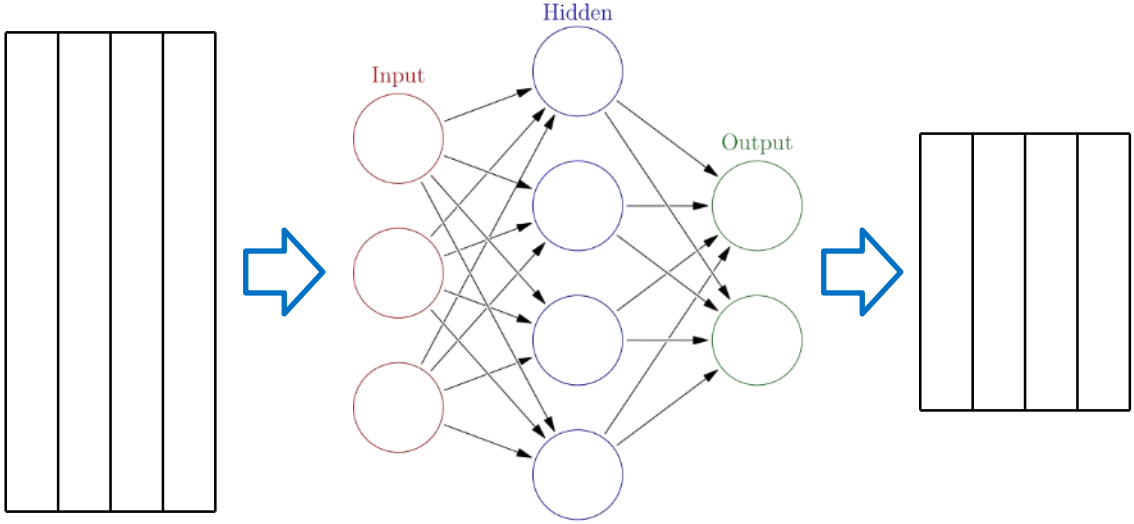
具体的例子 – 运算向量化

矩阵向量相乘（GEMV）

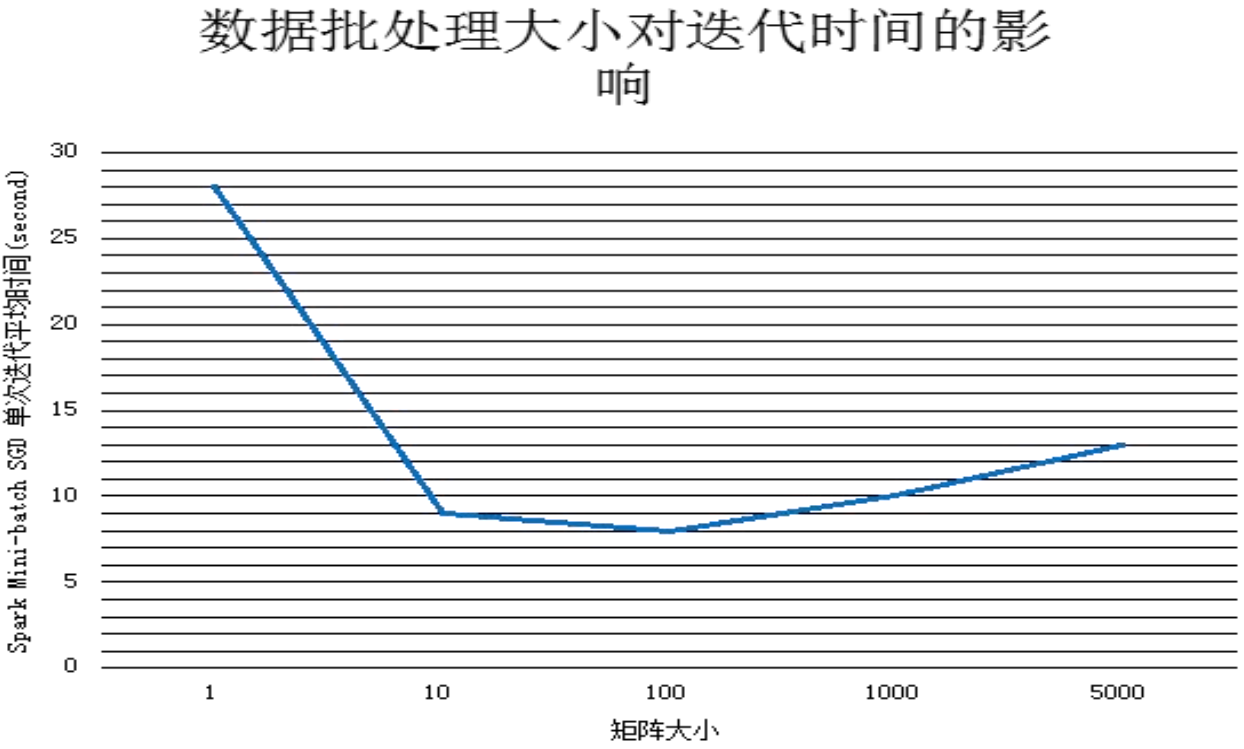


具体的例子 – 运算向量化

矩阵矩阵相乘（GEMM）



具体的例子 – 运算向量化



Intel® Math Kernel Library（MKL）

Intel CPU上最快的常用数学运算库

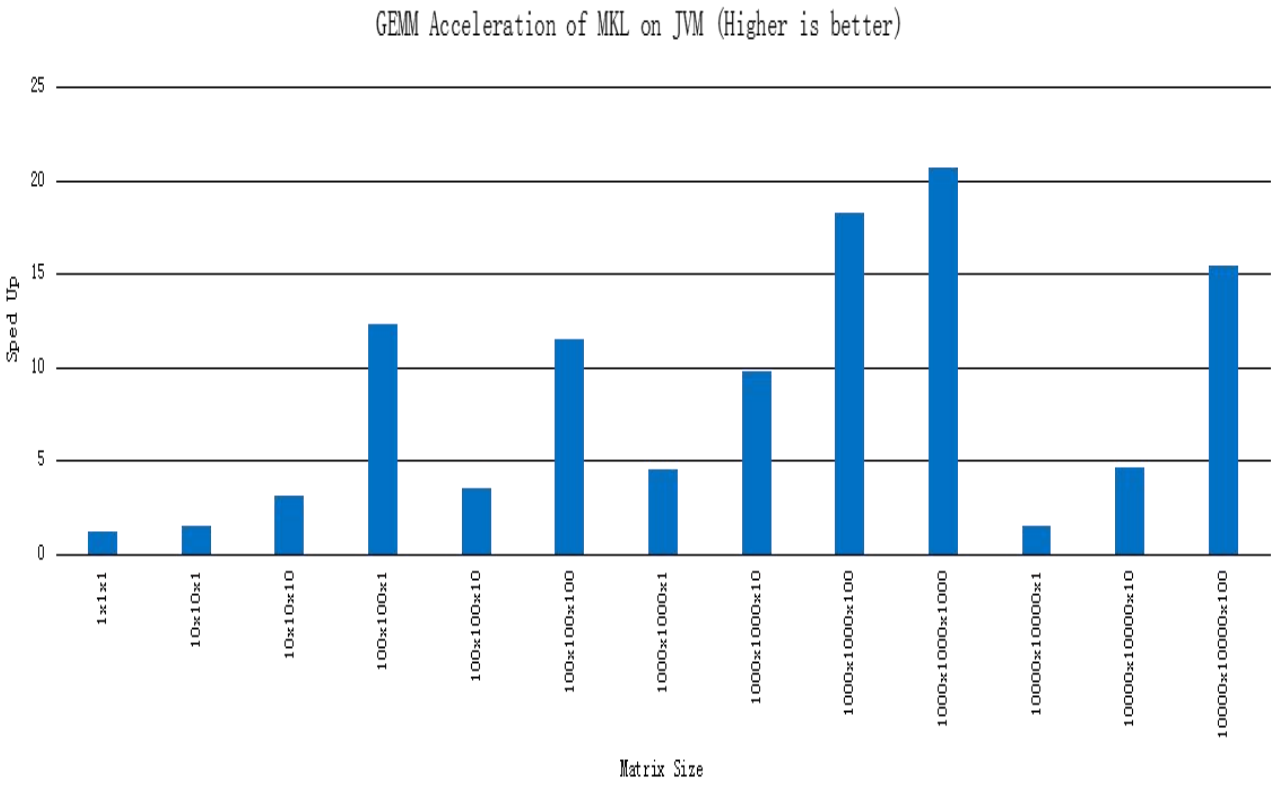
- BLAS/LAPACK/ARPACK(常用线性代数运算)
- Sparse BLAS（常用稀疏矩阵/向量运算）
- VML（常用向量化数学运算）
- FFT (快速傅里叶变换)

在Spark上利用MKL加速矩阵运算

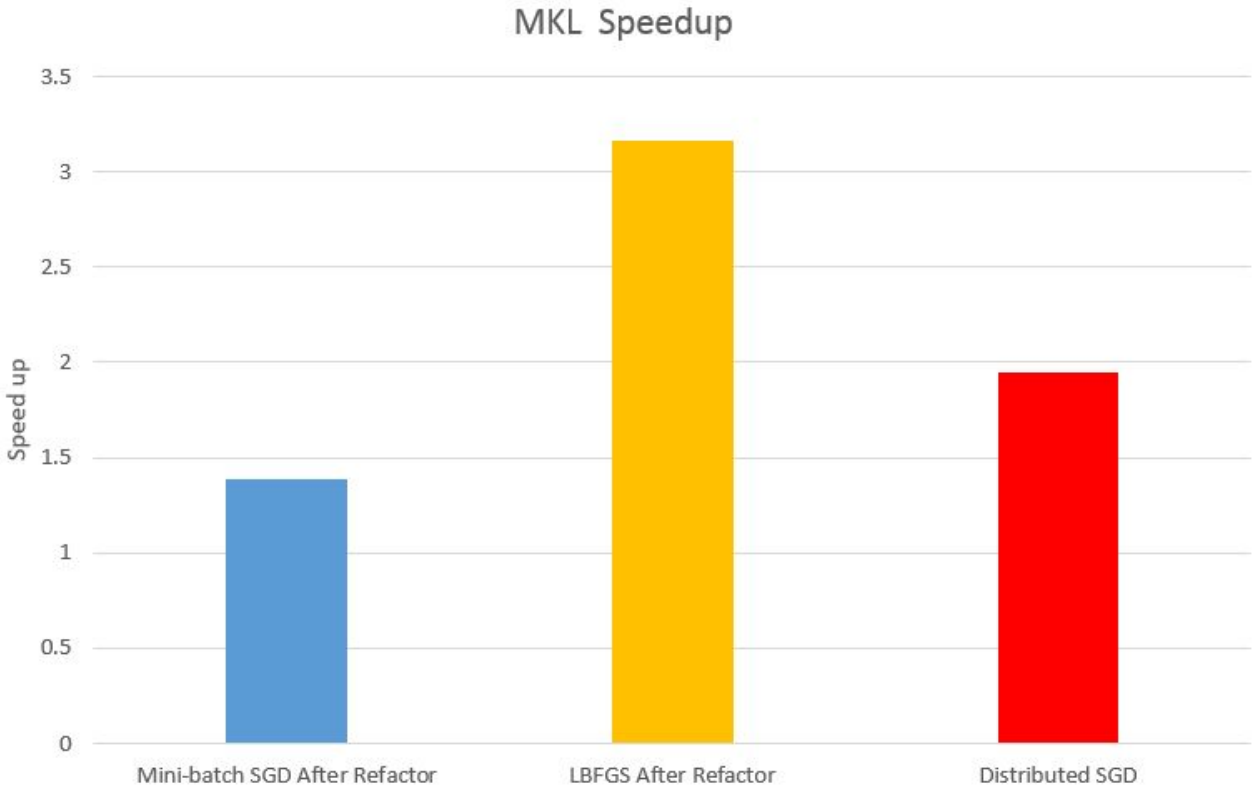
Spark上使用MKL

- 支持常用线性代数运算加速（BLAS/LAPACK/ARPACK）
- 通过JNI调用(netlib-java)
- 无需修改代码

在Spark上利用MKL加速矩阵运算



具体的例子 – 利用MKL加速ANN算法



总结

- 背景和概要介绍
  - 神经网络
  - Spark上对神经网络的支持
  - 我们的工作
- 通用计算平台上的性能优化
  - Profiling
  - 训练算法层面优化
  - 运算层面优化
  - 具体的例子

Q & A

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

Forecasts: Any forecasts of requirements for goods and services are provided for discussion purposes only. Intel will have no liability to make any purchase pursuant to forecasts. Any cost or expense you incur to respond to requests for information or in reliance on any forecast will be at your own risk and expense.

Business Forecast: Statements in this document that refer to Intel's plans and expectations for the quarter, the year, and the future, are forward-looking statements that involve a number of risks and uncertainties. A detailed discussion of the factors that could affect Intel's results and plans is included in Intel's SEC filings, including the annual report on Form 10-K.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.htm.

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.