神经网络处理

1. 神经网络算法

人工智能相关研究的发展在最近的几年呈现出异常火热的趋势。随着互联网时代数据的爆发式增长以及计算机计算能力的提升，人工智能再次向前发展。人工智能(Artificial Intelligence)，即试图理解智能的实质，并制造出能以人类智能相似的方式做出反应的智能机器。过去的很多努力还是基于某些预设规则的快速搜索和推理，离真正的智能还有相当的距离，或者说距离创造像人类一样具有抽象学习能力的机器还很遥远。而以神经网络算法为基础的能力为机器带来了抽象认知的能力。通过神经网络算法，机器也开始能够感知和观察我们所在的世界。

以图像识别为例，图像的原始输入是像素，相邻像素组成线条，多个线条组成纹理，进一步形成图案，图案构成了物体的局部，直至整个物体的样子。不难发现，可以找到原始输入和浅层特征之间的联系，再通过中层特征，一步一步获得和高层特征的联系。神经网络算法通过训练发现逐层特征之间的关系，学会挖掘出特征背后抽象的意义。

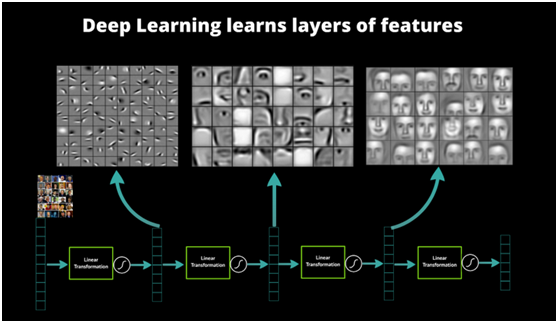


图 1 深度学习逐层学习特征[1]

(1)神经网络算法的仿生学依据

人工神经网络本身就是对人类神经系统的模拟，这种模拟具有仿生学的依据。人类的视觉系统包含了不同的视觉神经元，这些神经元与瞳孔所受的刺激（系统输入）之间存在着某种对应关系（神经元之间的连接参数），即受到某种刺激后（对于给定的输入），某些神经元就会活跃（被激活）。这证实了人类神经系统和大脑的工作其实是不断将低级抽象传导为高级抽象的过程，高层特征是低层特征的组合，越到高层特征就越抽象。

(2)特征的层次可表示性。



表1 层次表示

如上表所示，在真实世界不同的任务领域内其实都存在着逐层递进的认知过程。以中文文本举例，我们学习偏旁部首，组合拼接而成得到了汉字。汉字按照语义组合得到了单词。单词拼接组合得到短语。短语连接起来获得句子。多个句子拼接起来得到段落。而段落则组成了一篇篇文章。

在阅读相关的文献[1][2][3]之后，我了解到，无论是在文本、语音还是图像领域，神经网络算法的输入都是以向量或者矩阵的形式表达的。神经网络算法在对输入文本、语音或者图像的逐层前向传播求解与后向传播训练的算法中，均包含着大量的矩阵运算[4][5]，这明显地表现出神经网络算法的训练及应用可以通过并行计算的方式大大加快计算速度，提高其效率。

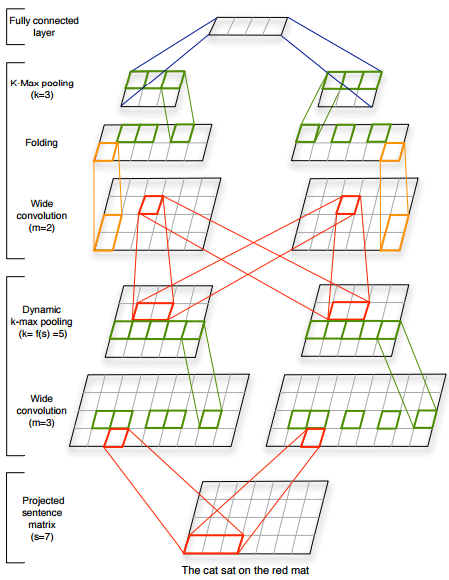


图 3 卷积神经网络对句子建模[5]

二、神经网络并行处理

神经网络算法是典型的并行算法，它的硬件实现均采用并行处理技术来提高处理器的并行度，进而达到缩短运行时间的目的。开发ANN算法并行性的等级按照并行粒度的大小分为微任务级、神经元级、任务级等，在任务级开发神经网络算法的并行性，其通用性较好。[6]

神经网络的并行处理通常应该指的是在神经网络算法的训练过程中使用并行处理技术提高其训练效率，减少训练时间。通常有两种解决方案：

(1)数据并行

数据并行是指对训练数据做切分，同时采用多个模型实例，对多个分片的数据并行训练。

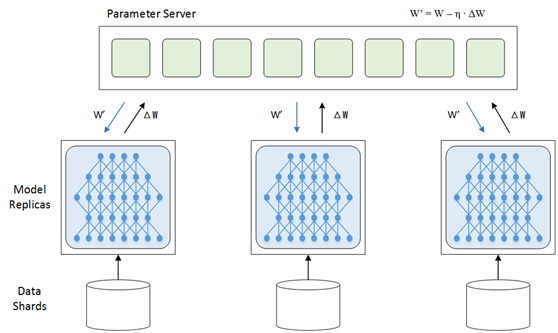


图4 数据并行的基本架构[8]

要完成数据并行需要做参数交换，通常由一个参数服务器（Parameter Server）来帮助完成。在训练的过程中，多个训练过程相互独立，训练的结果，即模型的变化量ΔW需要汇报给参数服务器，由参数服务器负责更新为最新的模型W’ = W – η ∙ ΔW，然后再将最新的模型W’分发给训练程序，以便从新的起点开始训练。

数据并行有同步模式和异步模式之分。同步模式中，所有训练程序同时训练一个批次的训练数据，完成后经过同步，再同时交换参数。参数交换完成后所有的训练程序就有了共同的新模型作为起点，再训练下一个批次。而异步模式中，训练程序完成一个批次的训练数据，立即和参数服务器交换参数，不考虑其他训练程序的状态。异步模式中一个训练程序的最新结果不会立刻体现在其他训练程序中，直到他们进行下次参数交换。

参数服务器只是一个逻辑上的概念，不一定部署为独立的一台服务器。有时候它会附属在某一个训练程序上，有时也会将参数服务器按照模型划分为不同的分片，分别部署。

(2)模型并行

模型并行将模型拆分成几个分片，由几个训练单元分别持有，共同协作完成训练。当一个神经元的输入来自另一个训练单元上的神经元的输出时，产生通信开销。

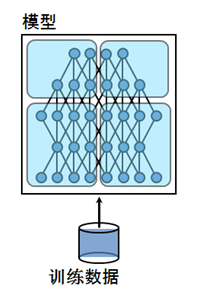


图5 模型并行的基本架构 [8]

多数情况下，模型并行带来的通信开销和同步消耗超过数据并行，因此加速比也不及数据并行。但对于单机内存无法容纳的大模型来说，模型并行是一个很好的选择。令人遗憾的是，数据并行和模型并行都不能无限扩展。数据并行的训练程序太多时，不得不减小学习率，以保证训练过程的平稳；模型并行的分片太多时，神经元输出值的交换量会急剧增加，效率大幅下降。因此，同时进行模型并行和数据并行也是一种常见的方案。如下图所示，4个GPU分为两组，GPU0，1为一组模型并行，GPU2，3为另一组，每组模型并行在计算过程中交换输出值和残差。两组GPU之间形成数据并行，Mini-batch结束后交换模型权重，考虑到模型的蓝色部分由GPU0和GPU2持有，而黄色部分由GPU1和GPU3持有，因此只有同色的GPU之间需要交换权重。

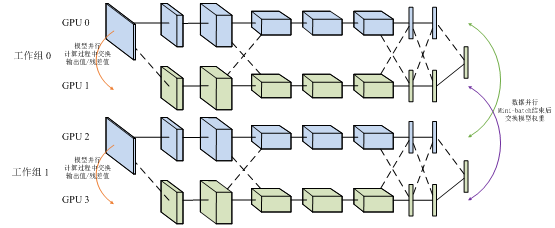


图6 4GPU卡的数据并行和模型并行混合架构[8]

神经网络是一种受生物学视觉皮层细胞启发的多层神经网络模型，它在结构中植入了二维模式的拓扑结构，可以从原始输入中学习到高阶不变性的特征，泛化能力强，在模式识别等诸多领域有广泛的应用。但是，训练一个串行实现的卷积神经网络往往需要花费数十天甚至几个月的时间。随着信息技术和存储技术的快速发展，在生产生活领域积累了TB级、PB级甚至更多的数据，并且随着卷积神经网络面临的问题越来越复杂，就需要网络具有更强的学习能力，而网络学习能力的增强需要更多的层和更多的可训练参数，这无疑加重了网络训练过程中的计算量，导致训练时间较长。较长的训练时间成为研究卷积神经网络，以及卷积神经网络在实际应用中的一个严重制约问题。针对这一问题，本文提出首先在云计算平台上并行实现卷积神经网络，增强网络的伸缩性，然后在平台的每个结点中采用GPU通用计算技术进一步并行化网络的前向传播和反向传播过程，加速网络的运算，缩短训练时间。

三、神经网络算法中的几种可并行性

通过对神经网络结构和训练过程的分析与学习，可以如下描述网络的

训练过程：

For数据集中各训练样本

For每个层(前向和反向)

For每层各特征图

For特征图上各神经元

For神经元各权值

对于每个For循环都意味着一种潜在可并行性，根据以上分析，卷积神经网络的训练算法至少存在五种并行方式，即训练样本的并行性、前反向层间的并行性、同层特征图的并行性，特征图中神经元的并行性和神经元权值的并行性。

(1)训练样本的并行性

将大量的训练样本映射到不同的处理单元进行处理，每个处理单元只处理整个训练集中的部分样本。各处理单元并行处理训练样本，处理结束之后交换结果，做一次批量更新。

(2)前反向层间并行性

样本以流水线的方式在网络各层进行计算，这就意味着在同一时刻，存在多个样本在网络中并行计算。

(3)特征图、神经元和权值并行性

同层特征图、神经元及神经元权值相互独立，可以并行计算。例如，在前向传播中，同层各特征图中神经元可以并行计算各自输出；在反向传播中，同层神经元的输出误差、各个权值和偏置的局部误差及权值的更新都可以并行计算而不相互影响。[12]

以上是从神经网络算法的流程方面挖掘并行性，下面再从三种主流的神经网络算法的结构、运行机制、并 行处理方面的特点进行总结。 a. 前馈反传型NN ( Feedforward with Back propagation ANN,简称 FFBP ANN)

①结构特点: 由输入层、若干中间层(隐层)、输出层构成的阶层型人工神经网络。

②运行机制特点--有监督学习。网络从输入层开始，层层计算每个神经元的响应值，根据计算出的实际输出值与理想输出值进行比较,得到误差(一般采用累计误差计算法)并调整网络连接权与学习率参数,接着开始下一模式的训练,直至误差减小到期望的范围内。

③并行处理特点:

1)同一层上每个神经元在同一时刻具有相同的输入向量,可以并行地通过矩阵运算得到各自的响应值。

2) 若使用累积误差计算法修正连接权矩阵和学习率,则具有不同输 入向量的不同模式,从输入层开始最终得到实际输出值的训练过程只要允许一个时间差的存在,也是可以并行实现。

b. HopField ANN

①结构特点: 全互连型网络 。

②运行机制特点--无监督学习。网络运行时每个神经元在同一时刻具有相同长度的输入向量,其状态(响应值)根据一定的工作规则得到更 新,最终当网络稳定,即每个神经元的响应值不变或在允许的范围内变化时,网络停止运行。

③ 并行处理特点: 同一时刻,每个神经元接收其余所有神经元的响应值作为其输入向量,通过矩阵运算更新其自身的响应值。网络中各个神经元更新一次响应值的处理可以并行实现。

c. 竞争型ANN

①结构特点: 由输入层、竞争层构成的两层网络。

②运行机制特点--无监督学习。通过竞争层神经元对输入模式的竞争响应,最后得到获胜神经元并将与之有关的连接权值朝着更有利于它竞 争的方向调整,并对邻近区域的神经元值进行抑制,再开始下一模式的训练。为了获得竞争层获胜神经元,存在一个求最大(小)值的logN复杂度的问题(设参与竞争的神经元个数为N)。

③并行处理特点: l)竞争层每个神经元由矩阵运算求得响应值的过程可以并行实现。2)求获胜神经元时,也可以用成对并行比较方法处理。

在实际实现的过程,利用MapReduce 等框架实现训练样本的并行化，即将整个训练集分为一个一个小块，分布式存储在并行计算平台的每个节点上。每个节点都存储一个相同的完整的神经网络， 各个结点使用该结点存储的数据对网络进行训练。对于小块中的每一个样本， 18 卷积神经网络的并行化设计与实现 结点都执行一次前向传播和反向传播计算，得出各个权值和偏置的局部改变量， 接着汇总每个权值和偏置的局部改变量得到全局改变量，多次用全局改变量更 新权值之后，获得最终网络。在这种并行方式中，训练集中所有的样本经过网 络计算之后进行一次通信，因此这是一种“粗粒度”的并行方式，可以有效的减少 平台中各个结点间的通信开销，有利于在分布式网络环境下实现。

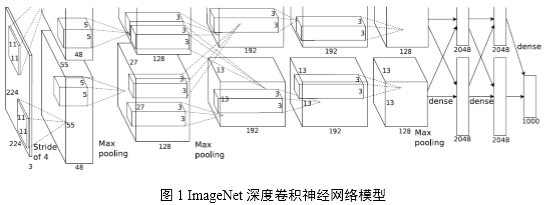
四、 多 GPU 卷积神经网络并行处理框架介绍

在这一节中，文献[8]中介绍了一种卷积神经网络的多 GPU 并行处理框架。

1. **CNNs 模型并行导论**

**1.1 典型应用分析：图像识别**

图像识别是深度卷积神经网络获得成功的一个典型应用范例。下图揭示了一个具有5个卷积层和3个全连接层的深度卷积神经网络，该模型可应用于图像分类。 

[[](http://cms.csdnimg.cn/article/201408/20/53f45bf969d24.jpg)](http://cms.csdnimg.cn/article/201408/20/53f45bf969d24.jpg)

**1.2 现有系统的问题**

在将CNN应用于图像相关领域的算法研究以及CNN训练平台搭建的实践过程中，受限于单个GPU上的显存大小，在尝试调整参数和网络规模的实验中，往往难以存储下更大规模的深度卷积神经网络模型，使得包含较多参数的网络不能在单GPU上训练，需要通过多GPU模型并行技术，拆分模型到多个GPU上存储和训练来解决。

随着训练数据集扩充、模型复杂度增加，即使采用GPU加速，在实验过程中也存在着严重的性能不足，往往需要十余天时间才能达到模型的收敛，不能满足对于训练大规模网络、开展更多试验的需求

考虑到上述问题，腾讯公司在其深度学习平台的Deep CNNs多GPU并行训练框架中，通过设计模型拆分方法、模型并行执行引擎和优化访存性能的Transfer Layer，并吸收在数据并行方面的设计经验，实现了多GPU加速的模型并行和数据并行版本。

在此描述多GPU加速深度卷积神经网络训练系统的模型并行和数据并行实现方法及其性能优化，以多GPU的强大协同并行计算能力为基础，结合目标Deep CNNs模型在训练中的并行特点，实现快速高效的深度卷积神经网络训练。

**1.3 框架设计目标**

多GPU模型并行+数据并行期望达到下述目标：充分利用Deep CNNs模型的可并行特点，结合SGD（Stochastic Gradient Descent，随机梯度下降）训练的数据并行特性，加速模型训练过程；突破显存大小限制，使得训练超过单GPU显存的模型成为可能，并预期通过训练更复杂的网络来获得更好的模型效果。

上述目标完成后，系统可以更快地训练目标Deep CNNs模型。模型拆分到不同GPU上可减少对单GPU显存占用，适用于训练更深层次、更多参数的卷积神经网络。

**1.4挑战**

在图像识别应用中，深度卷积神经网络模型的卷积层计算量大，全连接层参数多。因此，如何划分计算资源，通过模型并行和数据并行两个数据/计算组织层次上来加速训练是框架设计首要解决的问题。

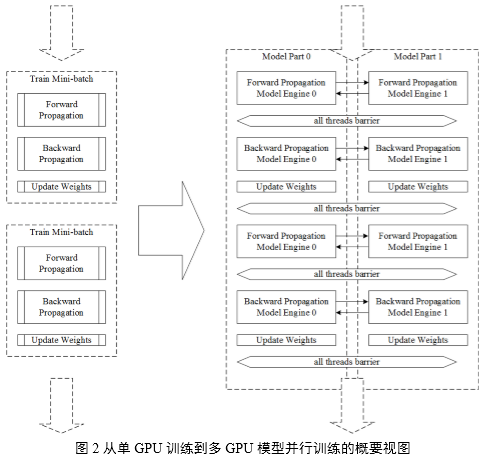
图像作为输入数据，其数据量庞大，且需要预处理过程，因此在Batch训练时磁盘I/O、数据预处理工作也要消耗一定时间。经典的用计算时间掩盖I/O时间的方法是引入流水线，因此如何设计一套有效的流水线方法来掩盖I/O时间和CPU处理时间，以使得整体耗时只取决于实际GPU训练时间，是一个重要问题。

模型并行是将一个完整Deep CNNs网络的计算拆分到多个GPU上来执行而采取的并行手段，结合并行资源对模型各并行部分进行合理调度以达到模型并行加速效果是实现模型并行的关键步骤。

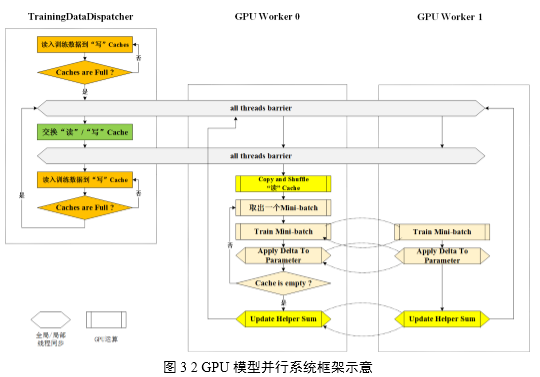
多GPU系统通过UVA（Unified Virtual Address，统一虚拟地址）技术，允许一颗GPU在kernel计算时访问其他GPU的设备内存（即显存），但由于远程设备存储访问速度远远低于本地存储访问速度，实际性能不佳。因此在跨GPU的邻接层数据访问时，需要关注如何高效利用设备间数据拷贝，使所有计算数据本地化。

**2.系统概述**

如图2所示，揭示了从单GPU训练到多GPU模型并行训练的相异之处，主要在于：在使用单GPU训练的场景下，模型不进行拆分，GPU显存上存储整个模型；模型并行的场景下，将模型拆分到多个GPU上存储，因此在训练过程中每个GPU上实际只负责训练模型的一部分，通过执行引擎的调度在一个WorkerGroup内完成对整个模型的训练。

[](http://cms.csdnimg.cn/article/201408/20/53f45c27ec48e.jpg)

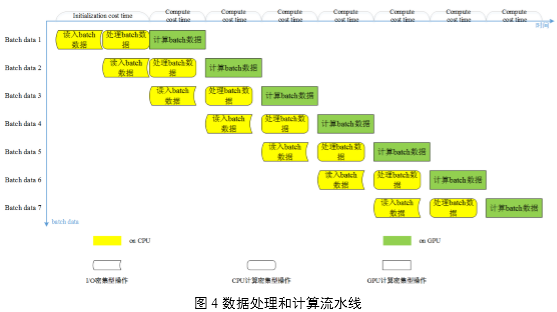
多GPU并行系统从功能上划分为用于读取和分发数据的Training Data Dispatcher和用于做模型并行训练的GPU Worker，如图3所示。训练数据从磁盘文件读取到CPU主存再拷贝到GPU显存，故此设计在各Worker计算每batch数据时，由Training Data Dispatcher从文件中读取并分发下一batch数据，以达到用计算时间掩盖I/O时间的设计目标。

[[](http://cms.csdnimg.cn/article/201408/20/53f45c44a834b.jpg)](http://cms.csdnimg.cn/article/201408/20/53f45c44a834b.jpg)

**3.训练数据处理的并行加速**

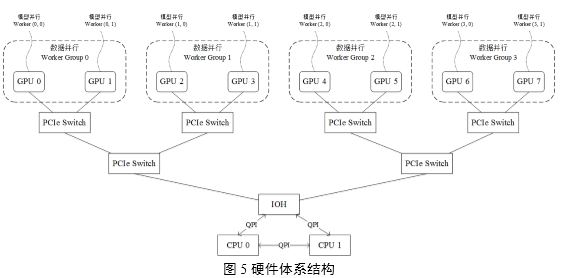
基于mini-batch的训练，现有技术方案在训练深度卷积神经网络时，每次从数据文件中读入和处理1个batch数据，在GPU计算某一batch时由CPU预读取和预处理下一batch。但是随着训练集图片像素数增大，读取和处理时间随之增加，由于采用多GPU技术加速了单个batch计算时间，数据处理的性能问题随之而来，需要减少数据处理的用时，以使最终加速效果取决于计算用时。

如图4所示，总体看来，在深度卷积神经网络训练过程中始终是在执行一条三阶段并行的流水线：计算本次batch数据——处理下次batch数据——读入再下次batch数据。

[[](http://cms.csdnimg.cn/article/201408/20/53f45c61a9548.jpg)](http://cms.csdnimg.cn/article/201408/20/53f45c61a9548.jpg)

**4.GPU Worker: 模型并行的承载体**

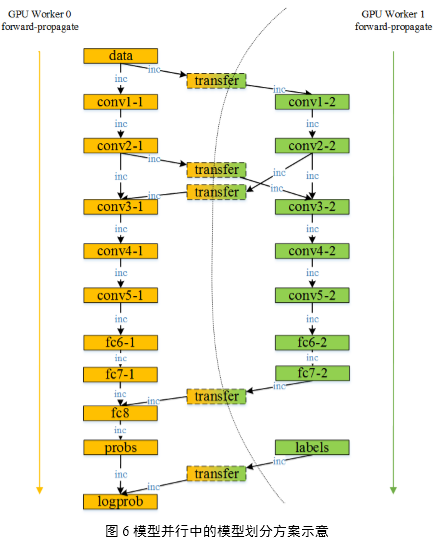
数据并行以划分Worker Group为基本组织形式，模型并行以在Worker Group内划分Worker为基本组织形式，并行训练的调度资源来源于CPU线程，计算资源来源于GPU卡。由于GPU卡通常意义上被看成是一种加速卡或协处理器卡，必须在基于CPU的主机上下文中被调用来做计算，因此遵循1个CPU线程绑定1张GPU卡能够发挥多GPU共同参与计算时的并行性效能。

[](http://cms.csdnimg.cn/article/201408/20/53f4616b5ada8.jpg)

**5.CNNs网络的模型划分**

**5.1.基本模型划分方法**

模型并行的来源是Deep CNNs网络只在特定层（如输入层、全连接层）与其他层有全面的连接，而其他较为独立的直线连接关系即可作为模型的可并行部分。将模型的可并行部分拆分到多个GPU上，同时利用多个GPU的计算能力各执行子模型的计算，可以大大加快模型的单次前向-后向训练时间。 

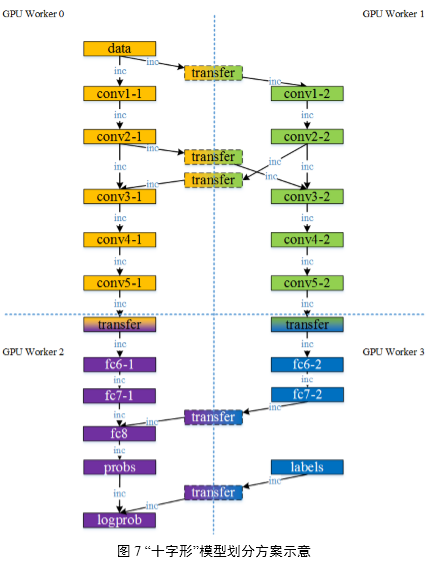
[[](http://cms.csdnimg.cn/article/201408/20/53f461877c497.jpg)](http://cms.csdnimg.cn/article/201408/20/53f461877c497.jpg)

DeepCNNs网络的层次模型实际上是一张有向无环图（DAG图），分配到每个模型并行Worker上的层集合，是有向无环图的拓扑排序子集，所有子集组成整个网络的1组模型。

**5.2“十字形”模型划分方法**

考虑极端情景：需要训练超大规模Deep CNNs模型，或者使用计算能力相对较强、显存较小（一般在1GB~3GB）的桌面级GeForce系列GPU，则利用模型本身的并行性这种基本的模型划分方法将不再适用。需要将模型再做拆分以保证单个GPU都能存储下对应的子模型。

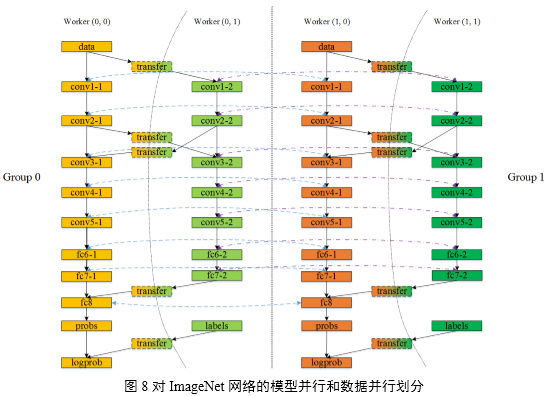
如图7所示，描述了将模型按“十字形”划分到4 Worker上训练的情景，不仅拆分了模型的可并行部分，也虽然这样的划分在Worker 0和Worker2之间，Worker 1和Worker 3之间达到并行加速效果，却能使得整个模型得以存储在4 GPU上。这种模型划分方法能够适用于训练超大规模网络等特殊模型的需求。

[[](http://cms.csdnimg.cn/article/201408/20/53f461ac30d2c.jpg)](http://cms.csdnimg.cn/article/201408/20/53f461ac30d2c.jpg)

**6.CNNs网络的模型并行工作引擎**

每个模型并行Worker上以一个模型并行执行引擎负责调度本Worker上子模型的执行过程。执行引擎控制所有Worker上的子模型完成前向和后向计算，各自对子模型完成参数更新后，到达主线程同步点，开始下一mini-batch训练。

多GPU模型并行和数据并行的Deep CNNs模型replicas及划分结构如图8所示，在使用4 GPU的场景下，划分了2组Worker Group用于数据并行；每个Worker Group内划分2个Worker用于模型并行。

[](http://cms.csdnimg.cn/article/201408/20/53f461d90edad.jpg)

五、 总结

近年来人工智能领域掀起了深度学习的浪潮，从学术界到工业界都热情高涨。深度学习尝试解决人工智能中抽象认知的难题，从理论分析和应用方面都获得了很大的成功。可以说深度学习是目前最接近人脑的智能学习方法。

通过此次读书报告的书写，我对深度学习以及神经网络算法的相关知识有了进一步的了解，同时对并行计算在神经网络处理中的应用有了进一步的认识。在并行集群上设计神经网络模型能够有效发挥其内部并行机制的优势，其训练时间与单机上的训练时间相比具有很大程度上的提高。随着并行处理技术的不断成熟，并行系统中的神经网络体现出了越来越广阔的应用前景。神经网络并行处理技术开发了计算资源的计算潜力，充分利用了神经网络算法具有的并行特性完成算法模型的训练与求解。并行计算可以利用集群或者多 GPU 的优势，加快并行处理神经网络中权值的学习，使得神经网络算法的研究与应用进一步靠近现实。

此次读书报告中我查阅了许多文献，对神经网络并行处理领域的研究有了一个初步的了解和认识，同时锻炼了我搜索、阅读文献的能力，收益颇丰。

参考文献

[1] [简单易学的机器学习算法——神经网络之BP神经网络](http://blog.csdn.net/google19890102/article/details/32723459)

[EB/OL] <http://blog.csdn.net/google19890102/article/details/32723459>

[2] 卷积神经网络在自然语言处理的应用

[EB/OL] <http://www.csdn.net/article/2015-11-11/2826192>

[3] 张晴晴, 刘勇, 王智超等;卷积神经网络在语音识别中应用 [C]

[4] Kim Y. Convolutional neural networks for sentence classification[J]. arXiv preprint arXiv:1408.5882, 2014.

[5] Kalchbrenner N, Grefenstette E, Blunsom P. A convolutional neural network for modelling sentences[J]. arXiv preprint arXiv:1404.2188, 2014.

[6] Alex Krizhevsky One weird trick for parallelizing convolutional neural networks [J]

[7] 钱 艺，王 沁，吴 巍，刘金龙；神经网络并行MIMD处理器的研究及实现[J]

[8] Dean, J., Corrado, G.S., Monga, R., et al, Ng, A. Y. Large Scale Distributed Deep Networks. In Proceedings of the Neural Information Processing Systems (NIPS’12) (Lake Tahoe, Nevada, United States, December 3–6, 2012). Curran Associates, Inc, 57 Morehouse Lane, Red Hook, NY, 2013, 1223-1232.

[9] 王裕民，顾乃杰，张孝慈；多GPU环境下的卷积神经网络并行算法 [J]

[10] 钱艺，李占才，李昂，王沁；一种神经网络并行处理器的体系结构 [J]

[11] 张代远；基于分布式并行计算的神经网络算法 [J]

[12] Baolei Fan Research on Parallelization of Convolutional Neural Networks [C]

[13] 陈云霁 从人工智能到神经网络处理器 [EB/OL]