k均值聚类，贝叶斯，神经网络，在线学习等等，还有很多其他的算法。比如说：免疫算法，遗传算法，主成分分析，蚁群算法等等。

<http://www.ssdfans.com/?p=4585>

<http://www.bigdata.ren/portal.php?mod=view&aid=1144>

http://blog.csdn.net/my\_share/article/details/38945137

        SA算法，也叫退火算法。和梯度下降法的意义是一样的。就是一种逼近模型。其实是一种2个函数逼近算法。其中一个函数，可能是离散的点，或者是半截函数，这是目标函数。我现在就用一个函数去逼近那个函数，两个函数之间的误差，到某个程度就逼近完成了。为了保证有比较优的解，算法往往采取慢降温、多抽样、以及把“终止温度”设的比较低等方式，导致算法运行时间比较长，但是从查找到的资料来看，它可以改进神经网络，神经网络是重要的一种图像识别方法。所以有必要对这方向做详细的了解分析。

局部搜索，模拟退火，[遗传算法](http://baike.baidu.com/view/45853.htm)，禁忌搜索的形象比喻：

为了找出地球上最高的山，一群有志气的兔子们开始想办法。

1．兔子朝着比现在高的地方跳去。他们找到了不远处的最高山峰。但是这座山不一定是珠穆朗玛峰。这就是局部搜索，它不能保证局部最优值就是全局最优值。

2．兔子喝醉了。他随机地跳了很长时间。这期间，它可能走向高处，也可能踏入平地。但是，他渐渐清醒了并朝最高方向跳去。这就是模拟退火。

3．兔子们吃了失忆药片，并被发射到太空，然后随机落到了地球上的某些地方。他们不知道自己的使命是什么。但是，如果你过几年就杀死一部分海拔低的兔子，多产的兔子们自己就会找到珠穆朗玛峰。这就是[遗传算法](http://baike.baidu.com/view/45853.htm)。

4.兔子们知道一个兔的力量是渺小的。他们互相转告着，哪里的山已经找过，并且找过的每一座山他们都留下一只兔子做记号。他们制定了下一步去哪里寻找的策略。这就是[禁忌搜索](http://baike.baidu.com/view/1117099.htm" \t "_blank)。

梯度下降法，它和SA算法作用是一样的。只是有一些改进上的变化。梯度下降法，它是神经网络中的一部分嘛。其实从其他的资料显示，用退火算法来改进神经网络会更好。我大概说下梯度下降法，梯度是最快的下降角度。比如你想找一条最短的路径走到一个盆地的最底部，梯度下降法每次只从你当前所处位置选一个坡度最大的方向走一步，牛顿法在选择方向时，不仅会考虑坡度是否够大，还会考虑你走了一步之后，坡度是否会变得更大。所以，可以说牛顿法比梯度下降法看得更远一点，能更快地走到最底部。  
遗传算法,遗传算法好像比较复杂一些，从性能角度上说，它好像是可以很容易的收敛到全局最小值，而不受局部最小值影响。缺点是，遗传算法通常的效率比其他传统的优化方法低。遗传算法容易过早收敛。遗传算法对算法的精度、可行度、计算复杂性等方面，还没有有效的定量分析方法。

禁忌算法，半启发式算法，前面那些都是启发式算法。小兔子的比喻已经能够理解一部分了，我在说点，其实这几种都是一样的作用，都是逼近模型。禁忌算法里面有一个表，也就是记录兔子在哪个山峰上。当然这个表是有限的，就是通过选最高的，在选最高的来找到最高点。也叫“tabu搜索算法”。从目前了解的情况来看，好像没有找到优点和缺点。看来需要了解的更多。应该有比较新的缘故吧。

 A星算法，对于这个算法我看了看好像也没有了解什么。可是从我查到的信息来看，好像和人工智能没什么关系。好像叫弱人工智能。但是它也是启发式搜索算法里面的一种。在查找资料的时候，好像很少和人工智能有关。所以不做深入了解了。有这么一个概念就好了。

蚁群算法，蚁群算法具有如下一些优点：①通用性较强，能够解决很多可以转换为连通图结构的路径优化问题；②同时具有正负反馈的特点，通过正反馈特点利用局部解构造全局解，通过负反馈特点也就是信息素的挥发来避免算法陷入局部最优；③有间接通讯和自组织的特点，蚂蚁之间并没有直接联系，而是通过路径上的信息素来进行间接的信息传递，自组织性使得群体的力量能够解决问题。但是，基本蚁群算法也存在一些缺点：①从蚁群算法的复杂度来看，该算法与其他算法相比，所需要的搜索时间较长；②该算法在搜索进行到一定程度以后，容易出现所有蚂蚁所发现的解完全一致这种“停滞现象”，使得搜索空间受到限制。从文字里面了解，好像蚁群算法还是挺复杂的。比较新，性能好像也可以，值得深入了解。所以我准备花时间去了解蚁群算法。<http://blog.163.com/ykn_2010/blog/static/1420333362012111411258466/>。原来蚁群算法就可以构成一本书。看来内容比较多呀。

 粒子群算法，PSO 算法属于[进化算法](http://baike.baidu.com/view/1798674.htm" \t "_blank)的一种，和[模拟退火](http://baike.baidu.com/view/476038.htm)算法相似，它也是从随机解出发，通过迭代寻找[最优解](http://baike.baidu.com/view/1009692.htm)，它也是通过[适应度](http://baike.baidu.com/view/2167684.htm)来评价解的品质，但它比遗传算法规则更为简单，。这种算法以其实现容易、精度高、收敛快等优点引起了学术界的重视，并且在解决实际问题中展示了其优越性。粒子群算法是一种并行算法。百度百科里面是这么描述的。粒子群算法也有专门的教程。10年出版。从整体描述看来，它好像是更新的，也是前面退火算法的升级版。根据华中科技大学硕士论文[http://www.doc88.com/p-371141950562.html](http://www.doc88.com/p-371141950562.html" \t "_blank)对粒子群算法的介绍，用于函数优化，神经网络训练，模糊系统控制等。看来，也是神经网络中的一部分。可能需要深入了解一下。

贪心算法，（开始）所以需要说明的是，贪心算法可以与[随机化算法](http://baike.baidu.com/view/1071553.htm" \t "_blank)一起使用，具体的例子就不再多举了。其实很多的智能算法（也叫启发式算法），本质上就是贪心算法和随机化算法结合——这样的算法结果虽然也是局部最优解，但是比单纯的贪心算法更靠近了最优解。例如遗传算法，模拟退火算法（结束）（百度百科里面的）。这句话说明遗传算法，退火算法是一种贪心算法和随机化算法的结合。（开始）值得注意的是，贪心算法并不是完全不可以使用，贪心策略一旦经过证明成立后，它就是一种高效的算法。贪心算法还是很常见的算法之一，这是由于它简单易行，构造贪心策略不是很困难（结束）（百度百科里面的）。从整体了解来看，贪心算法缺点很多，好像和人工智能机器学习没有多少关系。贪心算法中，仅在当前状态下作出最好选择，即局部最优选择。然后在局部最优，怎么感觉和分类决策树差不多呢？？但是它不是分类过程。pass。

文化算法，2002年David提出基于GP的双文化算法框架。此外他们还将文化算法用于图像分割、动态优化问题、数据挖掘等。[http://wenku.baidu.com/view/136a56b369dc5022aaea0096.html](http://wenku.baidu.com/view/136a56b369dc5022aaea0096.html" \t "_blank)这个文章里面有写到，1995年，利用文化算法，求解全局优化问题，并取得良好结果。双文化算法框架，用于图像分割和数据挖掘。[http://wenku.baidu.com/view/03f03dd249649b6648d7477d.html](http://wenku.baidu.com/view/03f03dd249649b6648d7477d.html" \t "_blank)我查看了很多资料，最终也没有了解出，文化算法的优缺点，用在什么地方等信息。但是从整体结构来看，好像比较特别。可能会在关注一下。

KMP算法，是一种模式匹配算法，没有了解到它和机器学习的联系。不过匹配功能好像是需要用到。考虑到KMP算法没有连用资料，所以暂时不考虑深入了解。

凭借出色的性能和功耗指标，赛灵思 FPGA 成为设计人员构建卷积神经网络的首选。新的软件工具可简化实现工作。

人工智能正在经历一场变革，这要得益于机器学习的快速进步。在机器学习领域，人们正对一类名为“深度学习”算法产生浓厚的兴趣，因为这类算法具有出色的大数据集性能。在深度学习中，机器可以在监督或不受监督的方式下从大量数据中学习一项任务。大规模监督式学习已经在图像识别和语音识别等任务中取得巨大成功。

深度学习技术使用大量已知数据找到一组权重和偏差值，以匹配预期结果。这个过程被称为培训，并会产生大型模式。这激励工程师倾向于利用专用硬件（例如 GPU）进行培训和分类。

随着数据量的进一步增加，机器学习将转移到云。大型机器学习模式实现在云端的 CPU 上。尽管 GPU 对深度学习算法而言在性能方面是一种更好的选择，但功耗要求之高使其只能用于高性能计算集群。因此，亟需一种能够加速算法又不会显著增加功耗的处理平台。在这样的背景下，FPGA 似乎是一种理想的选择，其固有特性有助于在低功耗条件下轻松启动众多并行过程。

让我们来详细了解一下如何在赛灵思 FPGA 上实现卷积神经网络 (CNN)。CNN 是一类深度神经网络，在处理大规模图像识别任务以及与机器学习类似的其他问题方面已大获成功。在当前案例中,针对在 FPGA 上实现 CNN 做一个可行性研究，看一下 FPGA 是否适用于解决大规模机器学习问题。

卷积神经网络是一种深度神经网络 (DNN)，工程师最近开始将该技术用于各种识别任务。图像识别、语音识别和自然语言处理是 CNN 比较常见的几大应用。

**什么是卷积神经网络？**

卷积神经网络是一种深度神经网络 (DNN)，工程师最近开始将该技术用于各种识别任务。图像识别、语音识别和自然语言处理是 CNN 比较常见的几大应用。

2012 年,Alex Krishevsky 与来自多伦多大学 (University of Toronto) 的其他研究人员 [1] 提出了一种基于 CNN 的深度架构，赢得了当年的“Imagenet 大规模视觉识别挑战”奖。他们的模型与竞争对手以及之前几年的模型相比在识别性能方面取得了实质性的提升。自此，AlexNet 成为了所有图像识别任务中的对比基准。

AlexNet 有五个卷积层和三个致密层（图 1）。每个卷积层将一组输入特征图与一组权值滤波器进行卷积，得到一组输出特征图。致密层是完全相连的一层，其中的每个输出均为所有输入的函数。

卷积层

AlexNet 中的卷积层负责三大任务，如图 2 所示：3D 卷积；使用校正线性单元 (ReLu) 实现激活函数；子采样（最大池化）。3D 卷积可用以下公式表示：

其中Y(m,x,y) 是输出特征图 m位置 (x,y) 处的卷积输出，S是 (x,y) 周围的局部邻域，W是卷积滤波器组，X(n,x,y)是从输入特征图 n上的像素位置 (x,y) 获得的卷积运算的输入。

所用的激活函数是一个校正线性单元，可执行函数 Max(x,0)。激活函数会在网络的传递函数中引入非线性。最大池化是 AlexNet 中使用的子采样技术。使用该技术，只需选择像素局部邻域最大值传播到下一层。

定义致密层

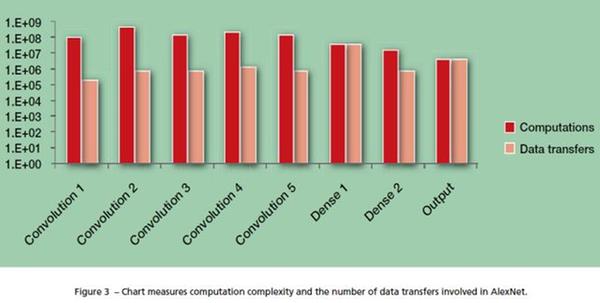
AlexNet 中的致密层相当于完全连接的层，其中每个输入节点与每个输出节点相连。AlexNet 中的第一个致密层有 9,216 个输入节点。将这个向量乘以权值矩阵，以在 4,096 个输出节点中产生输出。在下一个致密层中，将这个 4,096 节点向量与另一个权值矩阵相乘得到 4,096 个输出。最后，使用 4,096 个输出通过 softmax regression 为 1,000 个类创建概率。

在 FPGA 上实现 CNN

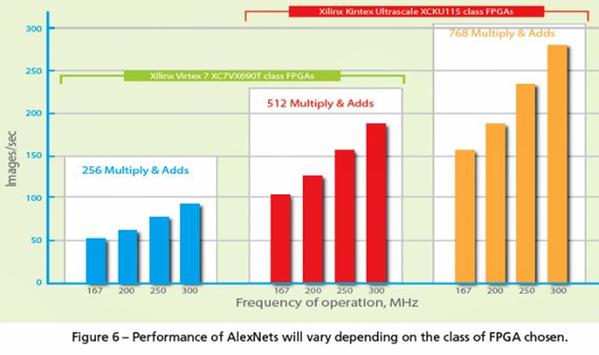
随着新型年高级设计环境的推出，软件开发人员可以更方便地将其设计移植到赛灵思 FPGA 中。软件开发人员可通过从 C/C++ 代码调用函数来充分利用 FPGA 与生俱来的架构优势。Auviz Systems 的库（例如 AuvizDNN）可为用户提供最佳函数，以便其针对各种应用创建定制 CNN。可在赛灵思 SD-Accel™ 这样的设计环境中调用这些函数，以在 FPGA 上启动内核。

最简单的方法是以顺序方式实现卷积和向量矩阵运算。考虑到所涉及计算量，因此顺序计算会产生较大时延。

顺序实现产生很大时迟的主要原因在于 CNN 所涉及的计算的绝对数量。图 3 显示了 AlexNet 中每层的计算量和数据传输情况，以说明其复杂性。

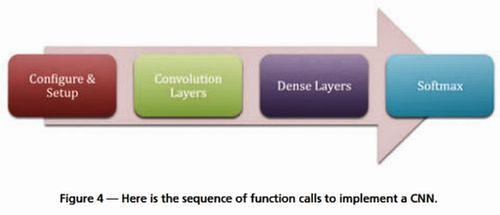


因此，很有必要采用并行计算。有很多方法可将实现过程并行化。图 6 给出了其中一种。在这里，将 11x11 的权值矩阵与一个 11x11 的输入特征图并行求卷积，以产生一个输出值。这个过程涉及 121 个并行的乘法-累加运算。根据 FPGA 的可用资源，我们可以并行对 512 抑或 768 个值求卷积。



为了进一步提升吞吐量，我们可以将实现过程进行流水线化。流水线能为需要一个周期以上才能完成的运算实现更高的吞吐量，例如浮点数乘法和加法。通过流水线处理，第一个输出的时延略有增加，但每个周期我们都可获得一个输出。

使用 AuvizDNN 在 FPGA 上实现的完整 CNN 就像从 C/C++ 程序中调用一连串函数。在建立对象和数据容器后，首先通过函数调用来创建每个卷积层，然后创建致密层，最后是创建 softmax 层，如图 4 所示。



AuvizDNN 是 Auviz Systems 公司提供的一种函数库,用于在 FPGA 上实现 CNN。该函数库提供轻松实现 CNN 所需的所有对象、类和函数。用户只需要提供所需的参数来创建不同的层。例如，图 5 中的代码片段显示了如何创建 AlexNet 中的第一层。

AuvizDNN 提供配置函数，用以创建 CNN 的任何类型和配置参数。AlexNet 仅用于演示说明。CNN 实现内容作为完整比特流载入 FPGA 并从 C/C++ 程序中调用，这使开发人员无需运行实现软件即可使用 AuvizDNN。

FPGA 具有大量的查找表 (LUT)、DSP 模块和片上存储器，因此是实现深度 CNN 的最佳选择。在数据中心，单位功耗性能比原始性能更为重要。数据中心需要高性能，但功耗要在数据中心服务器要求限值之内。

像赛灵思 Kintex® UltraScale™ 这样的 FPGA 器件可提供高于 14 张图像/秒/瓦特的性能，使其成为数据中心应用的理想选择。图 6 介绍了使用不同类型的 FPGA 所能实现的性能。

**一切始于 c/c++**

卷积神经网络备受青睐，并大规模部署用于处理图像识别、自然语言处理等众多任务。随着 CNN 从高性能计算应用 (HPC) 向数据中心迁移，需要采用高效方法来实现它们。

FPGA 可高效实现 CNN。FPGA 的具有出色的单位功耗性能，因此非常适用于数据中心。

AuvizDNN 函数库可用来在 FPGA 上实现 CNN。AuvizDNN 能降低 FPGA 的使用复杂性，并提供用户可从其 C/C++ 程序中调用的简单函数，用以在 FPGA 上实现加速。使用 AuvizDNN 时，可在 AuvizDNN 库中调用函数，因此实现 FPGA 加速与编写 C/C++ 程序没有太大区别。