Final Assignment

"Artificial Scientists and the needed paradigm shift to get there"

Science Theory and Research Methodology DD2205

Jim Holmström 890503-7571
jimho@kth.se

December 20, 2012

1 Summary

This essay will be focusing on the current software and hardware limitations towards the development of new machine learning techniques a specially the artificial scientist. Much of the claims and discussions will be based on the article "Towards 2020 Science" from Microsoft Research.[7]

It seems that we are currently standing even closer to the brink of a new computer revolution. New machine learning techniques are starting to be developed and set off into production. Cognitive computing is taking form. At least two big companies I have been working with, Ericsson and Oracle, are starting to consider using machine learning for tasks like for example abnormality detection for inhumanly complex systems with multiple soft states, classify the different states or different automatic utilities for testing software. This among other things results in the need of getting massive amounts of computing power. This hinder is nowadays much more focused on how to distribute the computations to more computation units or sometimes use graphic cards but this is not a long time solution. We basically need a new computation paradigm handmade for machine learning to get the machines as good as human performance in problems like object recognition or planing.

Having the cognitive system work would probably yield the solution for using automatic systems to do science. At the moment computers are not only used for computer aided engineering like FEM analysis but also in computer aided mathematics (CAM) where the proofs are too long or to complicated to be found by hand so the computers are given a set of reduction rules it has and then transforms the problem into a search problem through proof-space.

A cool though would be to strap a cognitive system to a system (could be everything from a real chemical mixer to a mathematical system) to do experiments, conclusions, and hypothesises on it's own with or without a clear

goal. And by this working as an artificial scientist (AS). Hopefully there will still be science jobs in the future.

2 The dawn of new hardware

The current processing capabilities for computers has lately degraded to a sub-exponential growth which is starting to have consequences such as restricting the use of some algorithms as well as setting limits to the processable size of datasets. This sub-exponential growth in computing has been a historical indicator that a new paradigm is on it's way, from vacuum tubes to transistors to integrated circuits to microprocessors, which should be the case this time as well or at least show that there is a need for one at least for artificial intelligence and machine learning.

The issue with most current hardware systems is their heavy and naïve use of the von Neumann architecture. Basically the throughput for this types of architecture has for physical reasons started to reach the end of the line performance wise.[8]

The duck-tape solution to this has been keeping the von Neumann architecture, with it's limiting bandwidth, and just throw in more cores and/or more machines in parallel. This might work well in some big data applications where the problems is easily map-reducible but perform really badly in many other problems.[10] Also this type of scaling becomes very space and power exhaustive since you have the need for cooling of all the cores.

The only other solution is to just drop the overused von Neumann architecture and go for something which is much more throughput oriented. In the 2020 paper[7] foresighted [1] the use of graphical processing units (GPU) for general computing (GPGPU). General problems in this case is basically problems with non graphical nature like solving differential equations, FFT or matrix operations. High end GPU's nowadays have around 1500 stream processors in parallel which can perform one type of operation (kernel) on a 1500-element-chunk of the data in parallel at a time. This together with the heavy use of a read-only memory which removes some of the problems with memory locks or write-after-write and that all the buses are separated for maximum throughput, they can in fact for many suitable cases not only scale almost linearly[2] on the number of cores but also have a much more effective throughput on the data crunching compared to ordinary CPU's which almost never manages.

Both the scaling up with ordinary CPU's into multi- or manycore as well as the GPGPU parallelization requires new libraries utilizing new programming paradigms.

For the paradigm shift for software part, one possible solution instead of the old classical object oriented programming mostly used today would be to use a much more higher level and pure functional programming language such as for

[1] 1 and 2 years before the official announcement of CUDA resp. openCL.

[2] Up to the GPU reaches the memory or instruction throughput max for the device which is much higher then for a CPU in the same price class.

example Haskell. Some data driven companies has already conformed to this and are using it as their main language. [3] but in my opinion more companies needs to follow the functional train and instead focus to have libraries and the language itself run much faster. The inherited properties that makes functional programming languages preferable is for example the lack of state which makes it lack execution order which in turn makes it trivial to convert it to run it in parallel in a multi-/manycore cluster or on a GPU, since it already run in parallel on a semantical level it can be implicitly parallelized.[9] [4][5] Both also the extremely high level of the language with compositions and higher order functions built-in.[1]

But I still think that an entirely new architecture needs to be developed even if the GPU solves many of the throughput bottlenecks in machine learning.

".. it is comparatively easy to make computers exhibit adult level performance on intelligence tests or playing checkers, and difficult or impossible to give them the skills of a one-year-old when it comes to perception and mobility."[6] which basically shows the main issue namely Moravec's paradox. This still holds and shows that there is a need for cognitive systems, not only for the great applications but also for understanding the processes in the brain that makes us process information, reason on it and react accordingly. A first step for understanding is to have models that can perform in the same way.

The probably the easiest way in trying to imitate neural behavior, which in many of the hardest cases outperforms computers, is to actually imitate neural processing. But a question still remains how closely should we imitate neural processing, do we need spiking networks with simulated chemical behavior or could one just stay with special purpose circuits for convolution networks?

For example having spiking instead of non-spiking would result in a more general network with finer temporal structure in contrast to the non-spiking which is a approximation where the spiking gets accumulated over a time. But there could of course be more to this difference that would have a vital effect on for example automatic correlation analysis for the network.

A new paradigm that recently has emerged is neuromorphic computing which basically tries to mimic neural functions with electronics.

The freshest and most promising according to me is the neural processing unit (NPU) [6] which both IBM[14] and Intel[13] are starting to develop. Both the approaches avoids being just a monstrous special purpose VHDL-logic chip, which we already have seen a few of. Those would still be more efficient though but it's not what we want coming out in the other end. These attempts from two giant corporations can thus be thought of as a new paradigm in itself going one step further then simple neuromorphic computing but still being one.

[3] Jane Street, Ericsson, Campanja,

[4] The statelessness in functional programming also makes it easier to debug as well as much harder to introduce bugs, but that is another story.

[5] The implicit parallelization is not the general case for Haskell but it could be theoretically easily be performed.

[6] Both the term "neural processing unit" and the abbreviation is not consistently used in relating communities, but personally I think it fits here.

Just as for GPGPU's the usage of a NPU's requires a new programming paradigm to operate, not much information on the NPU's nor their development tools have been released at the moment, but I can speculate that it would require some variant of logical- or/and functional programming. Also the algorithms that fits into the NPU's needs to be greatly rewritten.

Another rather new and interesting neumorphic device is the dynamic vision sensor (DVS) which basically has asynchronous event based readout by spiking on color changes.[5] This in contrast to ordinary cameras which only readout entire image sample for each time step. The DVS works much more like the human retina does and these retina properties makes it easier to do for example 3D reconstruction or tracking[12] as well as easily being really fast without using large bandwidth nor much power. A results of the extreme sparsity of data one can get well over 50000fps [7] over such a comparably low bandwidth interface as USB and draw just a few milliwatts of energy.[5]

In addition to all this some more radical improvements in learning algorithms has to be made as well. A very recent shift towards deep learning algorithms in the machine learning community that probably can be described as a paradigm shift. One cannot say for sure until after the breakthrough and this issue is much discussed in the machine learning community. For example it was foresighted that kernel-methods was going to solve everything and before that neural networks.

Much of the recent success in deep learning is due to the new found computing power from GPU's [8] as I mentioned before as well as a new look on deep learning. A great example on the later is the new usage of the dropout algorithm on deep networks, more specifically a convolution network, which not only outperforms the rest on the podium by xx% compared to yy% on the ImageNet dataset.[3] But does it without using any type of ad hoc methods except for log scaling some of the features compare to the competing algorithms that had a long list of state of the art ad hoc methods.[2] This shows some real promise for deep learning networks. The deep learning without ad hoc on images is really pushing the limit for what is reasonable with todays computing power, put it is just scratching the surface of what deep learning really could be with a paradigm shift in computing.

3 A new breed of scientists

The need of new hardware becomes really apperant both in theory and practice when trying to create a general intelligence. A few really recent and in many ways successful tries has been made like for example: Google's unsupervised cat detector.[4] These tries has been ran on large clusters but would gain a significant efficiency boost by being ran on a more natural habitat namely the NPU.

[7] Under normal "real life" conditions, it cannot of course reach this extreme fps if the neurons spikes constantly from each pixel.

[8] And Google's 16'000+ core strong computer cluster of course.

All these are probably the first baby steps towards having a cognitive system capable of finding good features completely unsupervised.

Having a cognitive system is probably a must for having a general AS working with human capabilities and hopefully a computers speed.

Still a few missing vital components (think there is even more) for AS to work would be to have some form of active learning by querying the experiment space by a measure of interest. The interest measure for focusing the search is hard problem and requires the AS to have a concept of both curiosity and creativity.[11]

Personally I think very interesting things are going to happen when we start connecting a good AS to systems like for example pure mathematics or DNA applications. The moment the CAM systems are proving very specific problems like the four coloring problem, but it would be cooler if the AS could surf through math-space and perhaps invent new mathematics by itself that it finds interesting. Of course after a find it halts and signals so that experts can review it just as an normal scientist has to do.

Finally I will highlight the possibility for recursiveness for artificial scientists: imagine letting a artificial scientist improve itself, so it can improve itself, so it can improve itself, so it can improve itself, so it can improve itself ...[9]

References

[1] The haskell programming language. `http://www.haskell.org`.

[2] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012.

[3] Imagenet is an organized image database. `http://www.image-net.org/`.

[4] Quoc V. Le, Marc'Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg S. Corrado, Jeff Dean, and Andrew Y. Ng. Building high-level features using large scale unsupervised learning. jul 2012.

[5] Lichtsteiner, P., Posch, C., and T. Delbruck. A 128 $\times$ 128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *Solid-State Circuits, IEEE Journal of*, 43(2):566 –576, feb 2008.

[6] *Mind Children: The Future of Robot and Human Intelligence*, page 15. 1988.

[7] Microsoft Research. Towards 2020 Science. `research.microsoft.com/en-us/um/cambridge/projects/towards2020science/downloads/T2020S_ReportA4.pdf`, jan 2006.

[9] Probably there is a theoretical limit somewhere to this, else we will see cognitive system being super intelligent and have an out of our world understanding of things.

[8] Cognitive systems: A new era of computing. http://www.research.ibm.com/new-era-of-computing.shtml.

[9] Prakash Prabhu, Soumyadeep Ghosh, Yun Zhang, Nick P. Johnson, and David I. August. Commutative set: A language extension for implicit parallel programming. 2011.

[10] Colby Ranger, Ramanan Raghuraman, Arun Penmetsa, Gary Bradski, and Christos Kozyrakis. Evaluating mapreduce for multi-core and multiprocessor systems. In *Proceedings of the 2007 IEEE 13th International Symposium on High Performance Computer Architecture*, HPCA '07, pages 13–24, Washington, DC, USA, 2007. IEEE Computer Society.

[11] Jürgen Schmidhuber. Active exploration, artificial curiosity and what's interesting. http://www.idsia.ch/~juergen/interest.html.

[12] S. Schraml, A.N. Belbachir, N. Milosevic, and P. Schön. Dynamic stereo vision system for real-time tracking. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pages 1409 –1412, jun 2010.

[13] Mrigank Sharad, Charles Augustine, Georgios Panagopoulos, and Kaushik Roy. Proposal for neuromorphic hardware using spin devices. jun 2012.

[14] IBM Unveils Cognitive Computing Chips. http://www-03.ibm.com/press/us/en/pressrelease/35251.wss, aug 2011.