Hinton & Nowlan's computational Baldwin effect revisit: Are we happy with it?

Brest State Technical University
Moskowskaja 267, Brest 224017 Republic of Belarus
akira@bsty.by

Abstract

In their seminal paper published in 1987, Hinton & Nowlan showed us an elegant experiment which might be called *an evolution with the Baldwin effect in computers* which searches for only one object located in a huge search space. The object was called *a-needle-in-a-haystack*. Hinton & Nowlan evolved a population of candidates of the solution in the same way as a standard evolutionary search. What made it unique was an exploitation of individual's *lifetime-learning*. Since then we have had a fair amount of proposals of how we reach the needle more efficiently. The issue, however, is still open to debate. We try to repeat their experiment and take a consideration on it.

1 Introduction

Let's start with a thought experiment. *"(i) Suppose we have N sea-shells on the table and a marble is hidden under one of those N shells. Then, how much query on average will be necessary to locate where the marble is? (ii) If no such marble is hidden at all, how many queries will be needed to know that?"* Or, we might imagine that we would try to break a N-digit PIN code (Personal Identification Number) by a *random-trial-and-error* or *exhaustive-one-by-one* search. Then what if N is very huge?

Why we search for a needle? — Computational analogue. The problem of looking for only one object hidden in a huge search space is sometimes called *a-needle-in-a-haystack* problem. It has long attracted and is still attracting many computer researchers. Let's name a few.

Crammer & Chechik (2004) defined the problem as, *"The problem of finding a small and coherent subset of points in a given data which sometimes referred to as one-class or set covering ..."* Joshi et al. (2001) pointed out yet another but a similar situation, writing, *"The traditional evaluation metric of accuracy is not adequate when the target-class is rare. If the class is very rare, say 0.5%, then predicting everything to be of non-target-class can also achieve very high accuracy level of 99.5%."* The technique proposed by Crammer & Chechik to solve this problem exploits two *phases* called *positive & negative*. Hence they call it *PN-rule*. Weiss (2004) summarized this method as, *"This approach identifies regions likely to contain needles in the first phase and then learns to discard the strands of hay within these regions in the second phase."* Weiss also discusses about, *"the role that rare classes and rare cases play in data mining,"* citing an interesting example of *"a machine learning technique to detect oil spills from satellite images (Kubat et al. 1998)."*

Sabhnani et al. (2003) applied nine different machine learning techniques to the KDD-cup-1999 dataset (Stolfo et al. 1999) to know how these techniques detect network intrusions. In the dataset, data for four categories of intrusions are given, together with data for normal transactions. It was shown that the two out of four categories were all immune to any of the nine methods more or less, while the other two were universally easy to be detected. We hypothesized that this is due to these two categories of attacks are like needles in a haystack of a normal and other categories of attacks (Imada, 2006).

Among others, besides a specific data in a huge database, the most popular issue on this topic these days is probably a searching for needles in a huge hay of world-wide-web resources, and designing such search-engines. See, for example, (Makris et al. 2006). In software engineering community, this issue is also explored. See, for example, (Whitaker et al. 2004).

Hinnton & Nowlan's model – Computational Baldwin Effect. In their seminal paper, Hinton & Nowlan (1987) proposed a needle defining it as just a unique configuration of 20-bit binary string, with all other configurations being a haystack. Then they tried an evolutionary search starting with a population of random candidates of the needle. A smart trick in their evolution is an inclusion of *flexible genes* in its *genotype*. Each of these flexible genes is replaced with either 1 or 0 in its *phenotype*. Then each of these phenotypes (again a binary 20-bit string) is checked if it matches the needle. Hence, this is called a *lifetime-learning*. Each genotype can try learning during its lifetime. Note that successful flexible genes in phenotype are not re-mapped into phenotype. This is a computational analogue of biological model of evolution called the *Baldwin effect* (Baldwin, 1896). Although we have had lots of reports concerning this approach, such as (Mills & Watson, 2006), the topic still includes open issues.

Note that if we apply a standard genetic algorithm starting with a population of chromosomes with their genes being either 0 or 1 at random, then the fitness of each individual is always zero unless the individual is ultra lucky to be coincidentally identical to the needle. Hence the fitness landscape is everywhere flat land of altitude zero except for the only one point. See Figure 1. How could we hillclimb if not a hill to climb! Impossible to evolve.

Figure 1: A fictitious sketch of fitness landscape of *a needle in a haystack*. The haystack here is drawn as a 2-dimensional flat plane of fitness zero.

However, their choice of 20-bit is a good one. We tried a random search with a similar condition of Hinton & Nowlan's experiment. The number of trials needed until the needle was found is plotted as a function of number of bits. Although the number explodes exponentially, as shown in Figure 2, 20-bits is just before the explosion. But in fact we never observed a needle longer than, say, 25 bits was reached in a reasonable time.

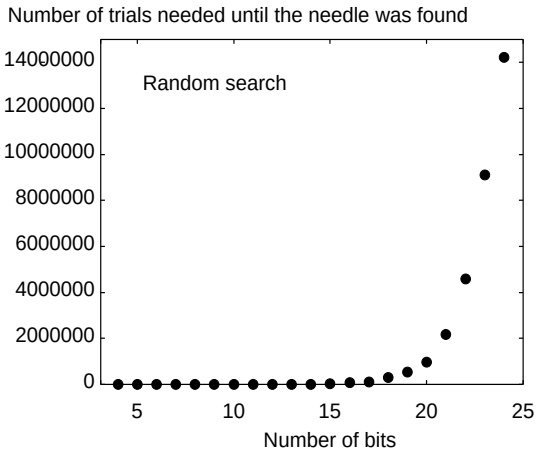

Figure 2: Assuming only one configulation of n-bit binary strings is a needle, the number of random trials needed until the needle was found is plotted against n. The result is an average during 100 runs.

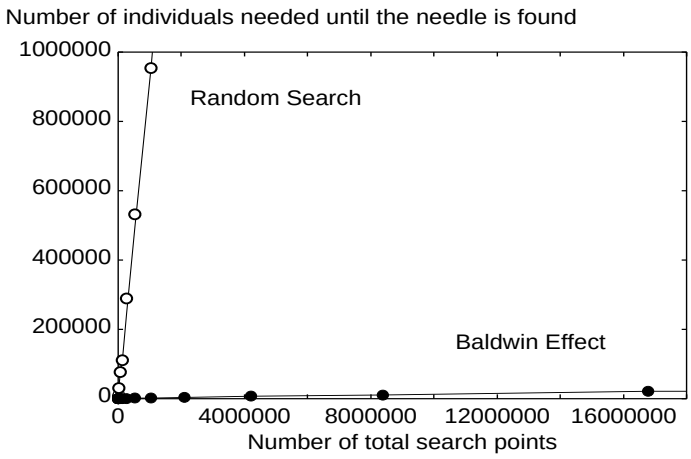

Figure 3: Number of individuals necessary for the needle to be reached for the first time vs. the number of total search points. A result of random walks (blank circles) and search by 1000 learnings during individual's lifetime (filled circles). The data are averaged after 100 runs. Complexity is both linear, but what a dramatic difference!

2 Experiment, Results, and Discussion

Following Hinton & Nowlan's experiment and a more clear specification of the experimental condition suggested by Mills & Watson (2005), we tried to repeat their experiment. We create a population of 1024 chromosomes of 20-bit whose genes are either 0, 1, or 9. Genes are determined at random with a probability of 0.25 for 0's and 1's and 0.5 for 9's. A needle here is the 20-bit string of all 1, without a loss of generality. Each chromosome is given a chance of 1000 trials each time with its 9 being randomly replaced either with 0 or 1, and check if the result of replacement matches the needle or not, if it matches the needle at n-th trial, the fitness is given as $(1000-n)$. Then we evolve this population from one generation to the next by *fitness-proportionate-selection*, *one-point-crossover* and *mutation* with the probability of 1/20.

Can we observe such an elegant result? In Figure 3 we show the number of individuals needed to find the needle in two cases: (i) by simple random search (the same data as in Figure 2) and (ii) by search with lifetime learning. The plots here are against total search points N instead of number of bits. Although the number is $O(N)$ in both cases, what a tremendously dramatic enhancement in efficiency it looks like! So far so good.

Why should we continue when the lifetime learning already find the needle? This is what we wondered when we read the original Hinton & Nowlan's paper. A possible answer was given by Mills & Watson (2005) in which they argued, *"This model is not intended to show any engineering advantage but a biological interest.* Then they went on, *"To remove the assumption of learning phenotypes, we evaluate fitness as the mean fitness of the lifetime phenotypes, rather than the number of trials remaining after the phenotypic solution is first found. This means that the organisms do not have to recognize their own success (as is implicit in Hinton & Nolan's model.)"* We were not fully satisfied with this answer but let's accept it here. Then the next question arises.

The higher the fitness of a genotype, the better it performs? Hinton & Nowlan's assumption is that the closer the *genotype* to the needle, the faster the learning of *phenotype*, which makes the needle-like-peak smoother. To study this, we compared their lifetime results of two genotypes: one reached the needle quickly (just 2 steps) and the other reached the needle but almost at the last chance (985 steps). We counted how many successes out of 1000 learnings of 1024 phenotypes. The histograms are shown in Figure 4. Although genotypes which reached the needle almost at the last chance sometimes reached the needle quickly, the higher fitness genotype by and large seems to perform better.

What is a fate of those flexible genes? Then questions are, "How many flexible genes are optimal?" or "Is the number of flexible genes decreasing as evolution proceeds?" Let's see what happened our evolution of these

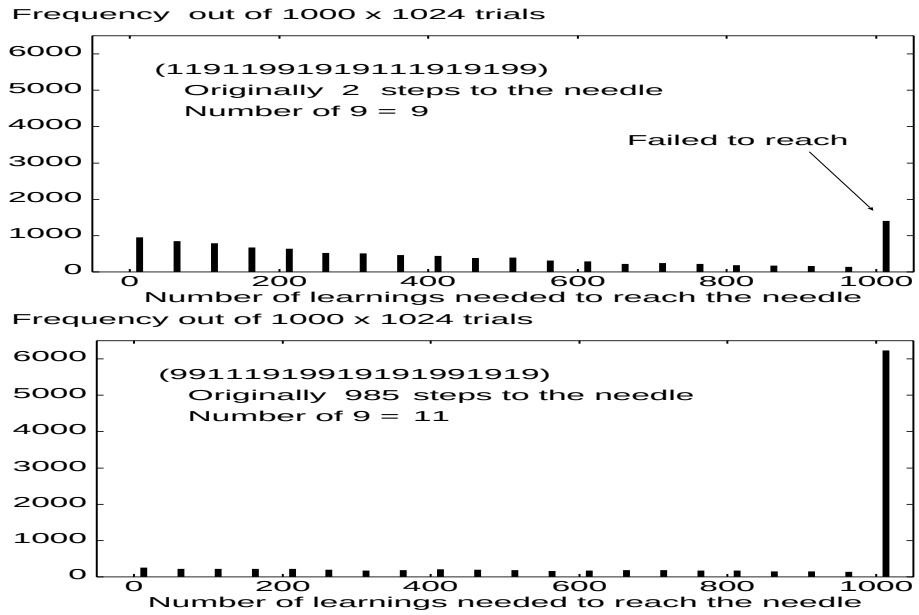

Figure 4: Examples of histogram of two genotypes of how many successes out of 1000 learnings of 1024 phenotypes. Genotype which reached the needle with 2 steps (top) and the other reached with 985 steps (bottom).

genotypes. We observed 100 runs with 100 different random number seeds, and what we saw in the 1st generation were only 4 successful genotypes at the luckiest case. The other 99 runs have only two, one or even zero successful genotypes. And in this luckiest case, the number of successful genes increased in an early stage of evolution to 300 – 400, but all of a sudden at some point of evolution, an extinction occurred. That is, the number of survival genotypes fell down to one or zero. Not possible for reproduction any more. (Figure to show this phenomenon is not in this paper.) Then, why don't we try another similar biological evolution model such as *Lamarckian inheritance* rather than sticking to the Baldwin Effect.

Is the Lamarckian inheritance computationally plausible? Turney (1996) wrote, *"Lamarckism requires an inverse mapping from phenotype and environment to genotype. This inverse mapping is biologically implausible."* And further assumed by describing, *"Perhaps Lamarckian evolution is superior to the Baldwin effect, when we are attempting to solve problems by evolutionary computation."* and the went on to write *"... (but) we believe that computing this mapping is intractable in general,"*

Why not? Let's try! This reverse mapping is really easy in Hinton & Nowlan's model. All we need is remap some of the successful 0's and 1's in the phenotype to the corresponding flexible genes in the genotype with a certain probability. Look at the typical success in its evolution shown in Figure 5. Though we still don't know this is *biologically* meaningful, this hypothesis once tried to be used to explain, "Why giraffe has a long neck?"

Are we really happy with these models to find needle? As already shown in Figure 2, exploiting the Baldwin effect seemed to tremendously enhance the efficiency of searching for the needle. But this is only under a comparison with the number of *individuals who tried to reach the needle*. If we compared with the number of *points which are visited*, we have almost similar result in those models. (Figure to show this is not included here.) Then how can we find a needle longer than, say, 25 bits?

In addition, we have to admit that even if we found a really efficient needle-detector, the steps we need would be still $O(N)$ like in Figure 3.

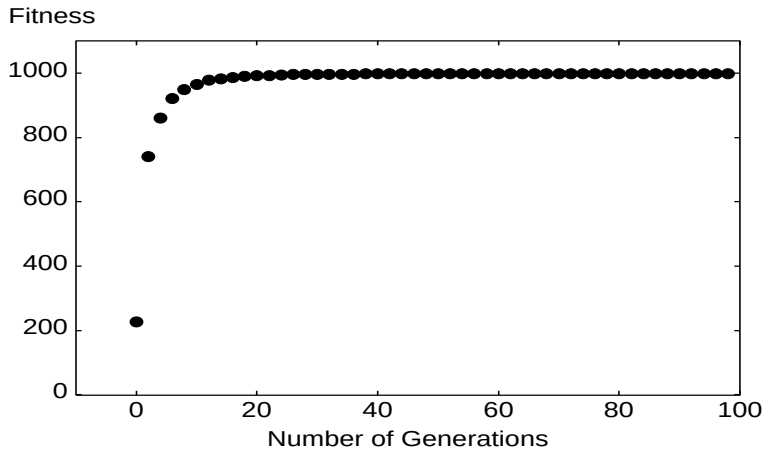

Figure 5: An evolution of genotypes including flexible genes by Lamarkian inheritance.

3 To Conclude

We have tried a bird's eye view on the topic on *a-needle-in-a-haystack problem*, or equivalently, a *computational model of the Baldwin effect*. The topic was initially proposed by Hinton & Nowlan two decades ago, but still this is a very important problem. Though we have had lots of proposals to approach to this problem, essentially we still have not find a truly efficient way of finding a needle in a really huge hay. Most techniques reported are still not sufficient when they are applied to a more scaled-up circumstance than when it was designed. A success in a small scale experiment is not necessarily a *royal road* to a success in a large scale situation in the real world.

Let me take an example. We now assume a robot in a $N \times N$ grid. A needle is hidden somewhere in the grid. Then the task of the robot is to look for the needle starting from somewhere in the grid. The robot has no idea of where is the needle. This is a two-dimensional version of a-needle-in-a-haystack problem. If N is small enough, the robot can eventually reach the needle even with random walks. And a learning, whatever it might be, can reduce the number of steps to the needle. We have so many successful reports of experiments of robot navigation in a much more complicated simulated world than the above mentioned simple grid world. The fact is, however, if the grid size explodes then we actually don't know how we make robot navigate as we like, even in a very simple world without no constraint such as corridors, walls or obstacles.

Yet another point we wanted to emphasize in this paper is that we have to avoid an effect of *like-to-hear-what-we-want-to-hear*. We had an interesting discussion between two papers: Yu & Miller's (2002) *"Finding needles in haystacks is not hard with neutrality"* vs. Collins' (2005) *"Finding needles in haystacks is harder with neutrality."*

Anyway, as already mentioned, all approaches described in this paper require $O(N)$ steps to the needle. Let me conclude this paper with the claim of a real speed up from $O(N)$ to $O(\sqrt{N})$ by Grover (1997) by his quantum search, which was already mathematically proved. The problem in his context is to find the needle from no-structured huge database. That is, *"Find x such that $P(x) = 1$ when only x from N data fulfills $P(x) = 1$ while all others do not."* This is owing to a strange path of quantum computation. Namely, when a particle goes from point A to point B, it takes all possible paths from A to B at the same time. In reality, however, no one so far has seen a real practical implementation of quantum computation.

We hope this article would be a good *prelude* to the re-opening this issue.

Acknowledgment

This topic was motivated by a comment from the floor when I talked in a seminar held in Vilnius in 2003 about evolutionary computations in general which included the topic of the computational Baldwin Effect. The comment insisted, *"Such a search will never be plausible,"* mentioning the Baldwin Effect. Though still I could say, "But it smoothes the fitness landscape, anyway," what he pointed out was correct in a sense written in this article. I now thank you for his comment.

Reference

Baldwin, J. M. (1896) "A New Factor in Evolution." American Naturalist, Vol. 30, pp. 441-457, 536-554.

Collins, M. (2005) "Finding Needles in Haystacks is Harder with Neutrality." In Proceedings of Genetic and Evolutionary Computation Conference, pp. 1613–1618.

Crammer, K., and G. Chechik (2004) "A Needle in a Haystack: Local One-Class Optimization." In Proceedings of International Conference on Machine Learning. http://citeseer.ist.psu.edu/crammer04needle.html

Grover, L. (1997) "Quantum Mechanics Helps in Searching for a Needle in a Haystack." Physical Review Letter, Vol. 79, pp. 325–328.

Hinton, G. E., and S. J. Nowlan (1987) "How Learning can Guide Evolution: Adaptive Individuals in Evolving Populations." Models and Algorithms, Addison-Wesley Longman, pp. 447–454.

Imada, A. (2006) "How Many Parachutists will be Needed to Find a Needle in a Pastoral?" In Proceedings of the International Conference on Neural Networks and Artificial Intelligence, pp. 53–60.

Joshi, M. V., R. C. Agarwal, and V. Kumar (2001) "Mining Needles in a Haystack: Classifying Rare Classes via Two-Phase Rule Induction." ACM SIGMOD (Special Interest Group on Management of Data) Record, Vol. 30 (2), pp. 91–102.

Kubat, M., R. C. Holte, and S. Matwin (1998) "Machine Learning for the Detection of Oil Spills in Satellite Radar Images." Journal of Machine Learning Vol. 30, pp. 195–215.

Makris, C., Y. Panagis, E. Sakkopoulos, and A. K. Tsakalidis (2006) "Efficient and Adaptive Discovery Techniques of Web Services Handling Large Data Sets." Journal of Systems and Software, Vol. 79(4), pp. 480–495.

Mills, R, and Watson R. A. (2005) "Genetic Assimilation and Canalization in the Baldwin Effect." In Proceedings of the European Conference on Artificial Intelligence, pp. 353-362.

Mills, R., and R. A. Watson (2006) "On Crossing Fitness Valleys with the Baldwin Effect." In Proceedings of Artificial Life X, pp. 493–499.

Sabhnani, M., and G. Serpen (2003) "Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset within Misuse Detection Context." In Proceedings of the International Conference on Machine Learning: Models, Technologies and Applications, pp. 209–215.

Stolfo, S. J., F. Wei, W. Lee, A. Prodromidis, and P. K. Chan (1999) "KDD Cup Knowledge Discovery and Data Mining Competition." http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

Turney, P. (1996) "Myths and Legends of the Baldwin Effect" In Proceedings of International Conference on Machine Learning, pp. 135–142.

Weiss, G. M. (2004) "Mining with Rarity: A Unifying Framework." ACM SIGKDD (Special Interest Group on Knowledge Discovery and Data Mining) Explorations, Vol. 6(1), pp. 7–19.

Whitaker, A., R. S. Cox, and S. D. Gribble (2004) "Configuration Debugging as Search: Finding the Needle in the Haystack." Journal of Operating Systems Design and Implementation 2004, pp. 77–90.

Yu, T., and J. Miller (2002) "Finding Needles in Haystacks is not Hard with Neutrality." In Proceedings of EuroGP 2002, Lecture Notes in Computer Science Vol. 2278, Springer, pp. 13–25.