The Hardware Lottery

Sara Hooker

Google Research, Brain Team
shooker@google.com

Abstract

Hardware, systems and algorithms research communities have historically had different incentive structures and fluctuating motivation to engage with each other explicitly. This historical treatment is odd given that hardware and software have frequently determined which research ideas succeed (and fail). This essay introduces the term hardware lottery to describe when a research idea wins because it is suited to the available software and hardware and *not* because the idea is superior to alternative research directions. Examples from early computer science history illustrate how hardware lotteries can delay research progress by casting successful ideas as failures. These lessons are particularly salient given the advent of domain specialized hardware which make it increasingly costly to stray off of the beaten path of research ideas. This essay posits that the gains from progress in computing are likely to become even more uneven, with certain research directions moving into the fast-lane while progress on others is further obstructed.

1 Introduction

History tells us that scientific progress is imperfect. Intellectual traditions and available tooling can prejudice scientists against certain ideas and towards others (Kuhn, 1962). This adds noise to the marketplace of ideas, and often means there is inertia in recognizing promising directions of research. In the field of artificial intelligence research, this essay posits that it is our tooling which has played a disproportionate role in deciding what ideas succeed (and which fail).

What follows is part position paper and part historical review. This essay introduces the term *hardware lottery* to describe when a research idea wins because it is compatible with available software and hardware and not because the idea is superior to alternative research directions. We argue that choices about software and hardware have often played a decisive role in deciding the winners and losers in early computer science history.

These lessons are particularly salient as we move into a new era of closer collaboration between hardware, software and machine learning research communities. After decades of treating hardware, software and algorithms as separate choices, the catalysts for closer collaboration include changing hardware economics (Hennessy, 2019), a "bigger is better" race in the size of deep learning architectures (Amodei et al., 2018; Thompson et al., 2020b) and the dizzying requirements of deploying machine learning to edge devices (Warden & Situnayake, 2019).

Closer collaboration has centered on a wave of new generation hardware that is "domain specific" to optimize for commercial use cases of deep neural networks (Jouppi et al., 2017; Gupta & Tan, 2019; ARM, 2020; Lee & Wang, 2018). While domain specialization creates important efficiency gains for mainstream research focused on deep neural networks, it arguably makes it more even more costly to stray off of the beaten path of research ideas. An increasingly fragmented hardware landscape means that the gains from progress in computing will be increasingly uneven. While deep neural networks have clear commercial use cases, there are early warning signs that the path to the next

硬件彩票

萨拉·胡克

谷歌研究，脑团队 shoker@
google.com

摘要

硬件、系统和算法研究社区在历史上有着不同的激励结构和波动的动机来明确地相互交流。考虑到硬件和软件经常决定哪些研究想法成功（或失败），这种历史处理显得奇怪。本文引入了“硬件彩票”这一术语，用以描述当一个研究想法之所以获胜是因为它适合现有的软件和硬件，而不是因为这个想法优于其他研究方向的情况。早期计算机科学历史中的例子说明了硬件彩票如何通过将成功的想法视为失败来延迟研究进展。考虑到领域专用硬件的出现，这些教训尤为重要，因为这使得偏离研究想法的主流路径变得越来越昂贵。本文认为，计算进步带来的收益可能会变得更加不均衡，某些研究方向将进入快车道，而其他方向的进展则进一步受到阻碍。

1 引言

历史告诉我们，科学进步是不完美的。知识传统和可用工具可能使科学家对某些想法产生偏见，而倾向于其他想法（库恩，1962）。这给思想市场增加了噪音，往往意味着在识别有前景的研究方向上存在惯性。在人工智能研究领域，这篇文章认为，是我们的工具在决定哪些想法成功（以及哪些失败）方面发挥了不成比例的作用。

接下来是部分立场文件和部分历史回顾。本文引入了“硬件彩票”这一术语，用以描述当一个研究想法获胜是因为它与现有的软件和硬件兼容，而不是因为这个想法优于其他研究方向。我们认为，关于软件和硬件的选择在早期计算机科学历史上往往在决定赢家和输家方面发挥了决定性作用。

这些课程在我们进入一个更紧密合作的新纪元时尤为重要。硬件、软件和机器学习研究社区之间的关系。经过数十年将硬件、软件和算法视为独立选择，促进更紧密合作的催化剂包括硬件经济的变化（Hennessy, 2019）、深度学习架构规模的“越大越好”竞赛（Amodei et al., 2018; Thompson et al., 2020b）以及将机器学习部署到边缘设备的令人眼花缭乱的要求（Warden & Situnayake, 2019）。

更紧密的合作集中在一波“特定领域”的新一代硬件上，以优化深度神经网络的商业用例（Jouppi et al., 2017; Gupta & Tan, 2019; ARM, 2020; Lee & Wang, 2018）。虽然领域专业化为专注于深度神经网络的主流研究带来了重要的效率提升，但可以说，这使得偏离研究思路的主流路径变得更加昂贵。日益碎片化的硬件格局意味着计算进步带来的收益将越来越不均衡。尽管深度神经网络有明确的商业用例，但有早期警告迹象表明，通往下一个{v*}的道路并不平坦。

Figure 1: Early computers such as the Mark I were single use and were not expected to be re-purposed. While Mark I could be programmed to compute different calculations, it was essentially a very powerful calculator and could not run the variety of programs that we expect of our modern day machines.

breakthrough in AI may require an entirely different combination of algorithm, hardware and software.

This essay begins by acknowledging a crucial paradox: machine learning researchers mostly ignore hardware despite the role it plays in determining what ideas succeed. In Section 2 we ask what has incentivized the development of software, hardware and machine learning research in isolation? Section 3 considers the ramifications of this siloed evolution with examples of early hardware and software lotteries. Today the hardware landscape is increasingly heterogeneous. This essay posits that the hardware lottery has not gone away, and the gap between the winners and losers will grow increasingly larger. Sections 4-5 unpack these arguments and Section 6 concludes with some thoughts on what it will take to avoid future hardware lotteries.

2 Separate Tribes

> It is not a bad description of man to describe him as a tool making animal.
>
> *Charles Babbage, 1851*

For the creators of the first computers the program was the machine. Early machines were single use and were not expected to be re-purposed for a new task because of both the cost of the electronics and a lack of cross-purpose software. Charles Babbage's difference machine was intended solely to compute polynomial functions (1817)(Collier, 1991). Mark I was a programmable calculator (1944)(Isaacson, 2014). Rosenblatt's perceptron machine computed a step-wise single layer network (1958)(Van Der Malsburg, 1986). Even the Jacquard loom, which is often thought of as one of the first programmable machines, in practice was so expensive to re-thread that it was typically threaded once to support a pre-fixed set of input fields (1804)(Posselt, 1888).

The specialization of these early computers was out of necessity and not because computer architects thought one-off customized hardware was intrinsically better. However, it is worth pointing out that our own intelligence is both algorithm and machine. We do not inhabit multiple brains over the course of our lifetime. Instead, the notion of human intelligence is intrinsically associated with the physical 1400g of brain tissue and the patterns of connectivity between an estimated 85 billion neurons in your head (Sainani, 2017). When we talk about human intelligence, the prototypical image that probably surfaces as you read this is of a pink ridged cartoon blob. It is impossible to think of our cognitive intelligence without summoning up an image of the hardware it runs on.

Today, in contrast to the necessary specialization in the very early days of computing, machine learning researchers tend to think of hardware, software and algorithm as three separate choices. This is largely due to a pe-

图1：早期计算机如马克一号是一次性使用的，不被期望重新利用。虽然马克一号可以被编程来计算不同的计算，但它本质上是一个非常强大的计算器，无法运行我们现代机器所期望的各种程序。

在人工智能的突破中，可能需要一种完全不同的算法、硬件和软件组合。

这篇文章首先承认一个关键的悖论：机器学习研究人员大多忽视硬件，尽管它在决定哪些想法成功中扮演着重要角色。在第二节中，我们询问是什么激励了软件、硬件和机器学习研究的孤立发展？第三节考虑了这种孤立演变的影响，并举例说明早期硬件和软件彩票。如今，硬件环境日益异质化。本文认为，硬件彩票并没有消失，赢家和输家之间的差距将越来越大。第四至第五节详细阐述了这些论点，第六节则总结了一些关于如何避免未来硬件彩票的思考。

2 个独立的部落

> 将人描述为一种制造工具的动物并不是一个坏的描述。
>
> *Charles Babbage, 1851*

对于第一台计算机的创造者来说，程序就是机器。早期的机器是一次性使用的，不被期望重新用于新任务，因为电子元件的成本和缺乏跨用途软件。查尔斯·巴贝奇的差分机仅用于计算多项式函数（1817）（Collier，1991）。马克 I 是一台可编程计算器（1944）（Isaacson，2014）。罗森布拉特的感知机计算了一个逐步的单层网络（1958）（Van Der Malsburg，1986）。即使是雅卡尔织机，通常被认为是最早的可编程机器之一，实际上重新穿线的成本非常高，因此通常只穿线一次以支持一组预设的输入字段（1804）（Posselt，1888）。

这些早期计算机的专业化是出于必要，而不是因为计算机架构师认为一次性定制硬件本质上更好。然而，值得指出的是，我们自己的智能既是算法也是机器。我们在一生中并不拥有多个大脑。相反，人类智能的概念本质上与1400克的脑组织以及你头部大约850亿个神经元之间的连接模式相关联（Sainani，2017）。当我们谈论人类智能时，您在阅读时可能浮现出的原型形象是一个粉红色的有脊的卡通球体。没有召唤出它运行所依赖的硬件的形象，就不可能想到我们的认知智能。

今天，与计算机早期阶段所需的专业化相比，机器学习研究人员倾向于将硬件、软件和算法视为三个独立的选择。这在很大程度上是由于一个pe-

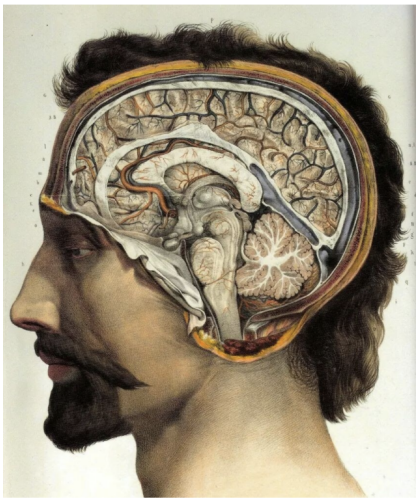

Figure 2: Our own cognitive intelligence is inextricably both hardware and algorithm. We do not inhabit multiple brains over our lifetime.

riod in computer science history that radically changed the type of hardware that was made and incentivized hardware, software and machine learning research communities to evolve in isolation.

2.1 The General Purpose Era

The general purpose computer era crystallized in 1969, when an opinion piece by a young engineer called Gordan Moore appeared in Electronics magazine with the apt title "Cramming more components onto circuit boards"(Moore, 1965). Moore predicted you could double the amount of transistors on an integrated circuit every two years. Originally, the article and subsequent follow-up was motivated by a simple desire – Moore thought it would sell more chips. However, the prediction held and motivated a remarkable decline in the cost of transforming energy into information over the next 50 years.

Moore's law combined with Dennard scaling (Dennard et al., 1974) enabled a factor of three magnitude increase in microprocessor performance between 1980-2010 (CHM, 2020). The predictable increases in compute and memory every two years meant hardware design became risk-averse. Even for tasks which demanded higher performance, the benefits of moving to specialized hardware could be quickly eclipsed by the next generation of general purpose hardware with ever growing compute.

The emphasis shifted to universal processors which could solve a myriad of different tasks. Why experiment on more specialized hardware designs for an uncertain reward when Moore's law allowed chip makers to lock in predictable profit margins? The few attempts to deviate and produce specialized supercomputers for research were financially unsustainable and short lived (Asanovic, 2018; Taubes, 1995). A few very narrow tasks like mastering chess were an exception to this rule because the prestige and visibility of beating a human adversary attracted corporate sponsorship (Moravec, 1998).

Treating the choice of hardware, software and algorithm as independent has persisted until recently. It is expensive to explore new types of hardware, both in terms of time and capital required. Producing a next generation chip typically costs $30-80 million dollars and 2-3 years to develop (Feldman, 2019). These formidable barriers to entry have produced a hardware research culture that might feel odd or perhaps even slow to the average machine learning researcher. While the number of machine learning publications has grown exponentially in the last 30 years (Dean, 2020), the number of hardware publications have maintained a fairly even cadence (Singh et al., 2015). For a hardware company, leakage of intellectual property can make or break the survival of the firm. This has led to a much more closely guarded research culture.

In the absence of any lever with which to influence hardware development, machine learning researchers rationally began to treat hardware as a sunk cost to work around rather than something fluid that could be shaped. However, just because we have abstracted away hardware does not mean it has ceased to exist. Early computer science history tells us there are many hardware lotteries where the choice of hardware and software has determined which ideas succeed (and which fail).

3 The Hardware Lottery

> I suppose it is tempting, if the only tool you have is a hammer, to treat everything as if it were a nail.
>
> *Abraham Maslow*, 1966.

The first sentence of Anna Karenina by Tolstoy reads "Happy families are all alike, every

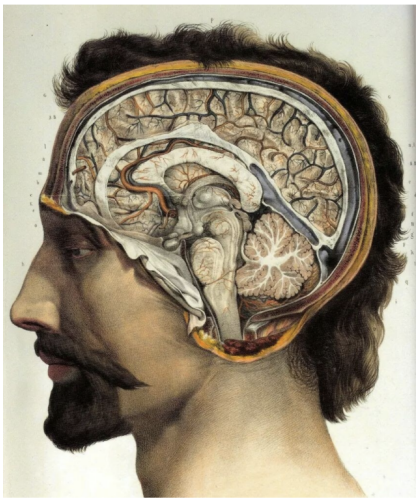

图2：我们自己的认知智能不可分割地既是硬件又是算法。我们在一生中并不拥有多个大脑。

计算机科学历史上一个彻底改变硬件类型的时期，激励了硬件、软件和机器学习研究社区在孤立中发展。

2.1 通用目的时代

通用计算机时代在1969年得以巩固，当时一位名叫戈登·摩尔的年轻工程师在《电子》杂志上发表了一篇题为“在电路板上塞入更多组件”的评论文章（摩尔，1965）。摩尔预测每两年可以将集成电路上的晶体管数量翻一番。最初，这篇文章及其后续跟进的动机很简单——摩尔认为这会卖出更多的芯片。然而，这一预测得以实现，并在接下来的50年中促成了将能量转化为信息的成本显著下降。

摩尔定律结合了丹纳德缩放（Dennard et al., 1974），使得1980年至2010年间微处理器性能提高了三倍（CHM, 2020）。每两年可预测的计算和内存增长意味着硬件设计变得规避风险。即使对于需要更高性能的任务，转向专用硬件的好处也可能很快被下一代通用硬件的不断增长的计算能力所掩盖。

重点转向了能够解决无数不同任务的通用处理器。为什么要在不确定的回报上对更专业的硬件设计进行实验，而摩尔定律允许芯片制造商锁定可预测的利润率？少数尝试偏离并为研究生产专业超级计算机的努力在财务上不可持续且寿命短暂（Asanovic, 2018；Taubes, 1995）。像掌握国际象棋这样的一些非常狭窄的任务是这个规则的例外，因为击败人类对手的声望和可见性吸引了企业赞助（Moravec, 1998）。

将硬件、软件和算法的选择视为独立的做法一直持续到最近。探索新类型的硬件既耗时又需要大量资金。生产下一代芯片通常需要3000万到8000万美元的成本和2到3年的开发时间（Feldman，2019）。这些巨大的进入壁垒产生了一种硬件研究文化，这对普通的机器学习研究人员来说可能显得奇怪甚至缓慢。在过去30年中，机器学习出版物的数量呈指数增长（Dean，2020），而硬件出版物的数量则保持了相对均匀的节奏（Singh等，2015）。对于一家硬件公司来说，知识产权的泄露可能会影响公司的生存。这导致了一种更加严密的研究文化。

在没有任何杠杆可以影响硬件发展的情况下，机器学习研究人员理性地开始将硬件视为一个沉没成本，而不是可以塑造的流动事物。然而，仅仅因为我们已经抽象掉了硬件，并不意味着它已经不存在。早期计算机科学的历史告诉我们，有许多硬件彩票，其中硬件和软件的选择决定了哪些想法成功（以及哪些失败）。

3 硬件彩票

> 我想，如果你唯一的工具是一把锤子，
> 那么把所有东西都当作钉子来对待是很
> 有诱惑的。
>
> *Abraham Maslow*, 1966.

《安娜·卡列尼娜》中的第一句话是“幸福的家庭都是相似的，每个不幸的家庭则各有各的不幸。”

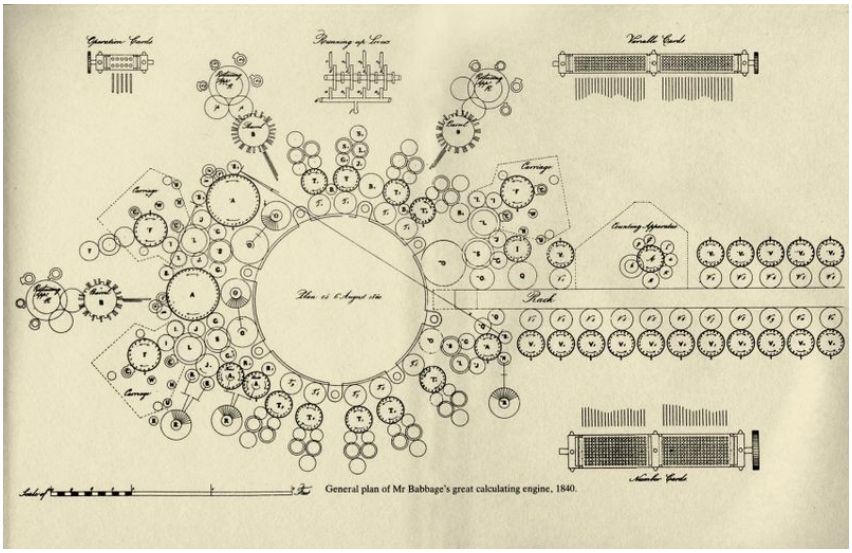

Figure 3: The analytical engine designed by Charles Babbage was never built in part because he had difficulty fabricating parts with the correct precision. This image depicts the general plan of the analytical machine in 1840.

unhappy family is unhappy in it's own way." (Tolstoy & Bartlett, 2016). Tolstoy is saying that it takes many different things for a marriage to be happy – financial stability, chemistry, shared values, healthy offspring. However, it only takes one of these aspects to not be present for a family to be unhappy. This has been popularized as the Anna Karenina principle – "a deficiency in any one of a number of factors dooms an endeavor to failure." (Moore, 2001).

Despite our preference to believe algorithms succeed or fail in isolation, history tells us that most computer science breakthroughs follow the Anna Karenina principle. Successful breakthroughs are often distinguished from failures by benefiting from multiple criteria aligning serendipitously. For AI research, this often depends upon winning what this essay terms the *hardware lottery* — avoiding possible points of failure in downstream hardware and software choices.

An early example of a hardware lottery is the analytical machine (1837). Charles Babbage was a computer pioneer who designed a machine that (at least in theory) could be programmed to solve any type of computation. His analytical engine was never built in part because he had difficulty fabricating parts with the correct precision (Kurzweil, 1990). The electromagnetic technology to actually build the theoretical foundations laid down by Babbage only surfaced during WWII. In the first part of the 20th century, electronic vacuum tubes were heavily used for radio communication and radar. During WWII, these vacuum tubes were re-purposed to provide the compute power necessary to break the German enigma code (Project, 2018).

As noted in the TV show Silicon Valley, often "being too early is the same as being wrong." When Babbage passed away in 1871, there was no continuous path between his ideas and modern day computing. The concept of a stored program, modifiable code, memory and conditional branching were rediscovered a century later because the right tools existed to empirically show that the idea worked.

3.1 The Lost Decades

Perhaps the most salient example of the damage caused by not winning the hardware lottery is the delayed recognition of deep neural networks as a promising direction of research. Most of the algorithmic components to make deep neural networks work had already been in place for a few decades: backpropagation (invented in 1963 (K & Piske, 1963), reinvented in 1976 (Linnainmaa, 1976), and again in 1988 (Rumelhart et al., 1988)), deep convolutional neural networks ((Fukushima & Miyake, 1982), paired with backpropagation in 1989 (LeCun et al., 1989)). However,

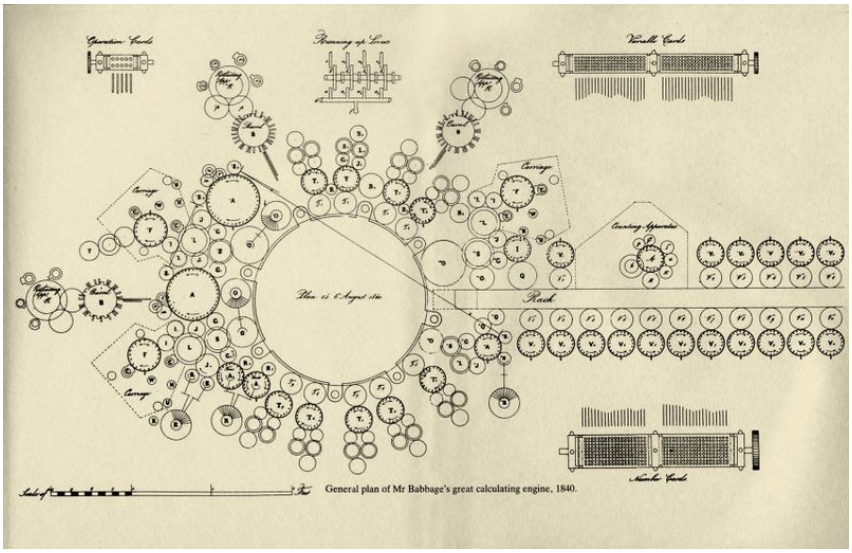

图3：查尔斯·巴贝奇设计的分析引擎从未建成，部分原因是他在制造精确零件方面遇到了困难。此图描绘了1840年分析机器的总体规划。

不幸的家庭各有各的不幸。”（托尔斯泰与巴特利特，2016）。托尔斯泰说，婚姻幸福需要许多不同的因素——经济稳定、化学反应、共同的价值观、健康的后代。然而，只需缺少其中一个方面，家庭就会变得不幸。这被称为安娜·卡列尼娜原则——“任何一个因素的缺失都会使一个努力注定失败。”（摩尔，2001）。

尽管我们倾向于相信算法的成功或失败是孤立发生的，但历史告诉我们，大多数计算机科学的突破遵循安娜·卡列尼娜原则。成功的突破通常与失败的区别在于它们受益于多个标准的偶然一致。对于人工智能研究，这往往依赖于赢得本文所称的硬件彩票——避免下游硬件和软件选择中的可能失败点。

早期的硬件彩票示例是分析机（1837年）。查尔斯·巴贝奇是一位计算机先驱，他设计了一台（至少在理论上）可以被编程以解决任何类型计算的机器。他的分析引擎从未被制造出来，部分原因是他在制造具有正确精度的部件时遇到了困难（库兹韦尔，1990年）。实际上构建理论基础所需的电磁技术被奠定。由巴贝奇提出的概念仅在二战期间浮出水面。在20世纪的前半部分，电子真空管被广泛用于无线电通信和雷达。在二战期间，这些真空管被重新利用，以提供破解德国恩尼格码所需的计算能力 {v*}（项目，2018）。

正如电视节目《硅谷》中所提到的，“过早就像是错误。”当巴贝奇于1871年去世时，他的思想与现代计算之间没有连续的路径。存储程序、可修改代码、内存和条件分支的概念在一个世纪后被重新发现，因为存在合适的工具可以实证地证明这个想法是可行的。

3.1 失落的十年

也许未能赢得硬件彩票造成的损害最显著的例子是深度神经网络作为一个有前景的研究方向的认可延迟。使深度神经网络能够工作的算法组件大约已经存在了几十年：反向传播（1963年发明（K & Piske, 1963），1976年重新发明（Linnainmaa, 1976），1988年再次发明（Rumelhart et al., 1988）），深度卷积神经网络（（Fukushima & Miyake, 1982），与1989年的反向传播配对（LeCun et al., 1989））。然而，

it was only three decades later that deep neural networks were widely accepted as a promising research direction.

This gap between algorithmic advances and empirical success is in large part due to incompatible hardware. During the general purpose computing era, hardware like CPUs were heavily favored and widely available. CPUs are very good at executing any set of complex instructions but incur high memory costs because of the need to cache intermediate results and process one instruction at a time (Sato, 2018). This is known as the von Neumann Bottleneck — the available compute is restricted by "the lone channel between the CPU and memory along which data has to travel sequentially" (Time, 1985).

The von Neumann bottleneck was terribly ill-suited to matrix multiplies, a core component of deep neural network architectures. Thus, training on CPUs quickly exhausted memory bandwidth and it simply wasn't possible to train deep neural networks with multiple layers. The need for hardware that supported tasks with lots of parallelism was pointed out as far back as the early 1980s in a series of essays titled "Parallel Models of Associative Memory" (Hinton & Anderson, 1989). The essays argued persuasively that biological evidence suggested massive parallelism was needed to make deep neural network approaches work (Rumelhart et al., 1986).

In the late 1980/90s, the idea of specialized hardware for neural networks had passed the novelty stage (Misra & Saha, 2010; Lindsey & Lindblad, 1994; Dean, 1990). However, efforts remained fractured by lack of shared software and the cost of hardware development. Most of the attempts that were actually operationalized like the Connection Machine in 1985 (Taubes, 1995), Space in 1992 (Howe & Asanović, 1994), Ring Array Processor in 1989 (Morgan et al., 1992) and the Japanese 5th generation computer project (Morgan, 1983) were designed to favor logic programming such as PROLOG and LISP that were poorly suited to connectionist deep neural networks. Later iterations such as HipNet-1 (Kingsbury et al., 1998), and the Analog Neural Network Chip in 1991 (Sackinger et al., 1992) were promising but short lived because of the intolerable cost of iteration and the need for custom silicon. Without a consumer market, there was simply not the critical mass of end users to be financially viable.

Figure 4: The connection machine was one of the few examples of hardware that deviated from general purpose cpus in the 1980s/90s. Thinking Machines ultimately went bankrupt after the inital funding from DARPA dried up.

It would take a hardware fluke in the early 2000s, a full four decades after the first paper about backpropagation was published, for the insight about massive parallelism to be operationalized in a useful way for connectionist deep neural networks. Many inventions are re-purposed for means unintended by their designers. Edison's phonograph was never intended to play music. He envisioned it as preserving the last words of dying people or teaching spelling. In fact, he was disappointed by its use playing popular music as he thought this was too "base" an application of his invention (Diamond et al., 1999). In a similar vein, deep neural networks only began to work when an existing technology was unexpectedly re-purposed.

A graphical processing unit (GPU) was originally introduced in the 1970s as a specialized accelerator for video games and for developing graphics for movies and animation. In the 2000s, like Edison's phonograph, GPUs were re-purposed for an entirely unimagined use case – to train deep neural networks (Chellapilla et al., 2006; Oh & Jung, 2004; Claudiu Ciresan et al., 2010; Fatahalian et al., 2004; Payne et al., 2005). GPUs had one critical advantage over CPUs - they were far better at parallelizing a set of simple decomposable instructions such as matrix multiples (Brodtkorb et al., 2013; Dettmers, 2020). This higher number of floating operation points per second (FLOPS) combined with clever distribution of training between GPUs unblocked the training of deeper networks. The number of layers in a network

三十年后，深度神经网络才被广泛接受为一个有前景的研究方向。

算法进步与经验成功之间的差距在很大程度上是由于不兼容的硬件。在通用计算时代，像CPU这样的硬件受到极大的青睐并广泛可用。CPU在执行任何一组复杂指令方面表现非常出色，但由于需要缓存中间结果并一次处理一条指令，因此会产生高内存成本（Sato, 2018）。这被称为冯·诺依曼瓶颈——可用的计算受到“CPU和内存之间的唯一通道的限制，数据必须沿着这个通道顺序传输”（Time, 1985）。

冯·诺依曼瓶颈对矩阵乘法极为不适合，而矩阵乘法是深度神经网络架构的核心组成部分。因此，在CPU上训练很快耗尽了内存带宽，根本无法训练具有多个层的深度神经网络。早在1980年代初，一系列名为“关联记忆的并行模型”（Hinton & Anderson, 1989）的文章就指出了对支持大量并行任务的硬件的需求。这些文章有力地论证了生物证据表明，需要大规模并行性才能使深度神经网络方法有效（Rumelhart et al., 1986）。

在1980/90年代末，专用神经网络硬件的理念已经超越了新奇阶段（Misra & Saha, 2010; Lindsey & Lindblad, 1994; Dean, 1990）。然而，由于缺乏共享软件和硬件开发成本，努力仍然分散。实际上投入使用的大多数尝试，如1985年的连接机器（Taubes, 1995）、1992年的空间（Howe & Asanovi , 1994）、1989年的环阵列处理器（Morgan et al., 1992）以及日本第五代计算机项目（Morgan, 1983），都是为了支持逻辑编程如PROLOG和LISP而设计的，这些编程语言并不适合连接主义深度神经网络。后来的迭代，如HipNet-1（Kingsbury et al., 1998）和1991年的模拟神经网络芯片（Sackinger et al., 1992）虽然有前景，但由于迭代成本不可承受和对定制硅的需求而短命。没有消费市场，根本没有足够的终端用户来实现财务可行性。

图4：连接机器是1980年代/90年代少数几种偏离通用CPU的硬件示例之一。思维机器在DARPA的初始资金枯竭后最终破产。

在2000年代初期，硬件的偶然因素使得关于大规模并行处理的洞察在连接主义深度神经网络中以有用的方式得以实现，这距离第一篇关于反向传播的论文发表已经过去了整整四十年。许多发明被重新用于设计者未曾预料的目的。爱迪生的留声机从未打算用来播放音乐。他设想它是用来保存临终者的最后遗言或教授拼写。事实上，他对其用于播放流行音乐感到失望，因为他认为这对他的发明来说是一个过于“低级”的应用（Diamond et al., 1999）。以类似的方式，深度神经网络只有在现有技术被意外重新利用时才开始发挥作用。

图形处理单元（GPU）最初在1970年代被引入，作为视频游戏和电影动画开发的专用加速器。在2000年代，像爱迪生的留声机一样，GPU被重新用于一个完全未曾想象的用例——训练深度神经网络（Chellapilla等，2006；Oh & Jung，2004；Claudiu Ciresan等，2010；Fatahalian等，2004；Payne等，2005）。GPU相较于CPU有一个关键优势——它们在并行处理一组简单可分解指令（如矩阵乘法）方面表现得更好（Brodtkorb等，2013；Dettmers，2020）。每秒浮点运算次数（FLOPS）的增加，加上在GPU之间巧妙的训练分配，解锁了更深层网络的训练。网络中的层数

turned out to be the key. Performance on ImageNet jumped with ever deeper networks in 2011 (Ciresan et al., 2011), 2012 (Krizhevsky et al., 2012) and 2015 (Szegedy et al., 2015b). A striking example of this jump in efficiency is a comparison of the now famous 2012 Google paper which used 16,000 CPU cores to classify cats (Le et al., 2012) to a paper published a mere year later that solved the same task with only two CPU cores and four GPUs (Coates et al., 2013).

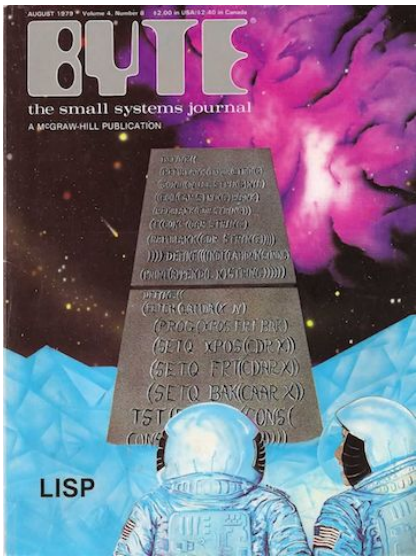

Figure 5: Byte magazine cover, August 1979, volume 4. LISP was the dominant language for artificial intelligence research through the 1990's. LISP was particularly well suited to handling logic expressions, which were a core component of reasoning and expert systems.

3.2 Software Lotteries

Software also plays a role in deciding which research ideas win and lose. Prolog and LISP were two languages heavily favored until the mid-90's in the AI community. For most of this period, students of AI were expected to actively master one or both of these languages (Lucas & van der Gaag, 1991). LISP and Prolog were particularly well suited to handling logic expressions, which were a core component of reasoning and expert systems.

For researchers who wanted to work on connectionist ideas like deep neural networks there was not a clearly suited language of choice until the emergence of Matlab in 1992 (Demuth & Beale, 1993). Implementing connectionist networks in LISP or Prolog was cumbersome and most researchers worked in low level languages like c++ (Touretzky & Waibel, 1995). It was only in the 2000's that there started to be a more healthy ecosystem around software developed for deep neural network approaches with the emergence of LUSH (Lecun & Bottou, 2002) and subsequently TORCH (Collobert et al., 2002).

Where there is a loser, there is also a winner. From the 1960s through the mid 80s, most mainstream research was focused on symbolic approaches to AI (Haugeland, 1985). Unlike deep neural networks where learning an adequate representation is delegated to the model itself, symbolic approaches aimed to build up a knowledge base and use decision rules to replicate how humans would approach a problem. This was often codified as a sequence of logic what-if statements that were well suited to LISP and PROLOG. The widespread and sustained popularity of symbolic approaches to AI cannot easily be seen as independent of how readily it fit into existing programming and hardware frameworks.

4 The Persistence of the Hardware Lottery

Today, there is renewed interest in joint collaboration between hardware, software and machine learning communities. We are experiencing a second pendulum swing back to specialized hardware. The catalysts include changing hardware economics prompted by the end of Moore's law and the breakdown of Dennard scaling (Hennessy, 2019), a "bigger is better" race in the number of model parameters that has gripped the field of machine learning (Amodei et al., 2018), spiralling energy costs (Horowitz, 2014; Strubell et al., 2019) and the dizzying requirements of deploying machine learning to edge devices (Warden & Situnayake, 2019).

The end of Moore's law means we are not guaranteed more compute, hardware will have to earn it. To improve efficiency, there is a shift from task agnostic hardware like CPUs to domain specialized hardware that tailor the design to make certain tasks more efficient. The first examples of domain specialized hardware released over the last few years – TPUs (Jouppi et al., 2017), edge-TPUs (Gupta & Tan, 2019), Arm Cortex-M55 (ARM, 2020), Facebook's *big sur* (Lee & Wang, 2018) – optimize explicitly for

结果证明这是关键。2011年（Ciresan et al., 2011）、2012年（Krizhevsky et al., 2012）和2015年（Szegedy et al., 2015b），在ImageNet上的表现随着网络的不断加深而跃升。一个引人注目的效率跃升例子是对比现在著名的2012年谷歌论文，该论文使用了16,000个CPU核心来分类猫（Le et al., 2012），而仅仅一年后发表的一篇论文则仅用两个CPU核心和四个GPU就解决了同样的任务（Coates et al., 2013）。

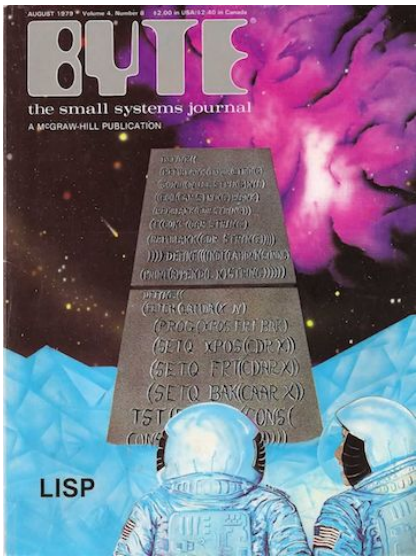

图5：Byte杂志封面，1979年8月，第4卷。LISP是1990年代人工智能研究的主导语言。LISP特别适合处理逻辑表达式，这些表达式是推理和专家系统的核心组成部分。

3.2 软件抽奖

软件在决定哪些研究想法获胜和失败方面也发挥了作用。Prolog 和 LISP 是在人工智能社区中直到90年代中期受到高度青睐的两种语言。在这段时间的大部分时间里，人工智能的学生被期望积极掌握这两种语言中的一种或两种（Lucas & van der Gaag, 1991）。LISP 和 Prolog 特别适合处理逻辑表达式，而逻辑表达式是推理和专家系统的核心组成部分。

对于希望研究连接主义思想如深度神经网络的研究人员来说，在1992年Matlab出现之前，没有一种明确适合的语言可供选择（Demuth & Beale，1993）。实现连接主义的{v*}在LISP或Prolog中实现连接主义网络是繁琐的，大多数研究人员使用低级语言如c++（Touretzky & Waibel, 1995）。直到2000年代，围绕深度神经网络方法开发的软件才开始形成一个更健康的生态系统，LUSH（Lecun & Bottou, 2002）的出现以及随后TORCH（Collobert et al., 2002）的推出。

在有输家的地方，也有赢家。从20世纪60年代到80年代中期，大多数主流研究集中在符号方法的人工智能上（Haugeland，1985）。与深度神经网络将学习适当表示的任务委托给模型本身不同，符号方法旨在建立知识库，并使用决策规则来复制人类解决问题的方式。这通常被编码为一系列逻辑的假设语句，非常适合LISP和PROLOG。符号方法在人工智能领域的广泛和持续的受欢迎程度，不能轻易地被视为与其如何适应现有编程和硬件框架无关。

4 硬件彩票的持久性

今天，硬件、软件和机器学习社区之间的联合合作重新引起了人们的兴趣。我们正经历第二次摆动，回归到专用硬件。催化剂包括由于摩尔定律的结束和丹纳德缩放的崩溃（Hennessy, 2019）而引发的硬件经济学变化，机器学习领域中对模型参数数量的“越大越好”竞赛（Amodei et al., 2018），不断攀升的能源成本（Horowitz, 2014; Strubell et al., 2019）以及将机器学习部署到边缘设备的令人眼花缭乱的要求（Warden & Situnayake, 2019）。

摩尔定律的结束意味着我们不再保证有更多的计算能力，硬件必须通过自身的价值来获得。为了提高效率，硬件的方向从像 CPU 这样的任务无关硬件转向专门针对特定领域的硬件，这些硬件的设计旨在使某些任务更加高效。过去几年发布的第一个专门领域硬件的例子——TPU（Jouppi 等，2017），边缘 TPU（Gupta & Tan，2019），Arm Cortex-M55（ARM，2020），Facebook 的 Big Sur（Lee & Wang，2018）——明确优化了 {v*}。

costly operations common to deep neural networks like matrix multiplies.

Closer collaboration between hardware and research communities will undoubtedly continue to make the training and deployment of deep neural networks more efficient. For example, unstructured pruning (Hooker et al., 2019; Gale et al., 2019; Evci et al., 2019) and weight specific quantization (Zhen et al., 2019) are very successful compression techniques in deep neural networks but are incompatible with current hardware and compilation kernels.

While these compression techniques are currently not supported, many clever hardware architects are currently thinking about how to solve for this. It is a reasonable prediction that the next few generations of chips or specialized kernels will correct for hardware biases against these techniques (Wang et al., 2018; Sun et al., 2020). Some of the first designs to facilitate sparsity have already hit the market (Krashinsky et al., 2020). In parallel, there is interesting research developing specialized software kernels to support unstructured sparsity (Elsen et al., 2020; Gale et al., 2020; Gray et al., 2017).

In many ways, hardware is catching up to the present state of machine learning research. Hardware is only economically viable if the lifetime of the use case lasts more than three years (Dean, 2020). Betting on ideas which have longevity is a key consideration for hardware developers. Thus, co-design effort has focused almost entirely on optimizing an older generation of models with known commercial use cases. For example, matrix multiplies are a safe target to optimize for because they are here to stay — anchored by the widespread use and adoption of deep neural networks in production systems. Allowing for unstructured sparsity and weight specific quantization are also safe targets because there is wide consensus that these will enable higher levels of compression.

There is still a separate question of whether hardware innovation is versatile enough to unlock or keep pace with entirely new machine learning research directions. It is difficult to answer this question because data points here are limited – it is hard to model the counterfactual of would this idea succeed given different hardware. However, despite the inherent challenge of this task, there is already compelling evidence that domain specialized hardware makes it more costly for research ideas that stray outside of the mainstream to succeed.

In 2019, a paper was published called "Machine learning is stuck in a rut." (Barham & Isard, 2019). The authors consider the difficulty of training a new type of computer vision architecture called capsule networks (Sabour et al., 2017) on domain specialized hardware. Capsule networks include novel components like squashing operations and routing by agreement. These architecture choices aimed to solve for key deficiencies in convolutional neural networks (lack of rotational invariance and spatial hierarchy understanding) but strayed from the typical architecture of neural networks. As a result, while capsule networks operations can be implemented reasonably well on CPUs, performance falls off a cliff on accelerators like GPUs and TPUs which have been overly optimized for matrix multiplies.

Whether or not you agree that capsule networks are the future of computer vision, the authors say something interesting about the difficulty of trying to train a new type of image classification architecture on domain specialized hardware. Hardware design has prioritized delivering on commercial use cases, while built-in flexibility to accommodate the next generation of research ideas remains a distant secondary consideration.

While specialization makes deep neural networks more efficient, it also makes it far more costly to stray from accepted building blocks. It prompts the question of how much researchers will implicitly overfit to ideas that operationalize well on available hardware rather than take a risk on ideas that are not currently feasible? What are the failures we still don't have the hardware and software to see as a success?

5 The Likelyhood of Future Hardware Lotteries

> What we have before us are some breathtaking opportunities disguised as insoluble problems.
>
> *John Gardner, 1965.*

It is an ongoing, open debate within the machine learning community about how much future algorithms will differ from models like deep neural networks (Sutton, 2019; Welling, 2019). The risk you attach to depending on domain specialized hardware is tied to your position on this debate. Betting heavily on

深度神经网络中常见的代价高昂的操作，如矩阵乘法。

硬件和研究社区之间更紧密的合作无疑将继续使深度神经网络的训练和部署更加高效。例如，非结构化剪枝（Hooker et al., 2019; Gale et al., 2019; Evci et al., 2019）和权重特定量化（Zhen et al., 2019）是深度神经网络中非常成功的压缩技术，但与当前的硬件和编译内核不兼容。

虽然这些压缩技术目前尚不支持，但许多聪明的硬件架构师正在思考如何解决这个问题。可以合理预测，下一代几代芯片或专用内核将会纠正对这些技术的硬件偏见（Wang et al., 2018; Sun et al., 2020）。一些促进稀疏性的初步设计已经进入市场（Krashinsky et al., 2020）。与此同时，正在进行有趣的研究，开发专用软件内核以支持非结构化稀疏性（Elsen et al., 2020; Gale et al., 2020; Gray et al., 2017）。

在许多方面，硬件正在赶上机器学习研究的现状。只有在使用案例的生命周期超过三年时，硬件才具有经济可行性（Dean，2020）。押注于具有持久性的想法是硬件开发者的关键考虑。因此，协同设计的努力几乎完全集中在优化具有已知商业用例的旧一代模型上。例如，矩阵乘法是一个安全的优化目标，因为它们将继续存在——受到深度神经网络在生产系统中广泛使用和采用的支持。允许非结构稀疏性和权重特定量化也是安全的目标，因为广泛达成共识认为这些将实现更高水平的压缩。

仍然有一个单独的问题，即硬件创新是否足够多功能，以解锁或跟上全新的机器学习研究方向。回答这个问题很困难，因为这里的数据点有限——很难建模在不同硬件下这个想法是否会成功的反事实。然而，尽管这个任务固有的挑战，已经有令人信服的证据表明，领域专用硬件使得它对{v*}的成本更高。

研究那些偏离主流的想法以取得成功。

在2019年，发表了一篇名为“机器学习陷入困境”的论文。（Barham & Isard，2019）。作者考虑了在领域专用硬件上训练一种新型计算机视觉架构——胶囊网络（Sabour et al.，2017）的困难。胶囊网络包括新颖的组件，如压缩操作和通过协议进行路由。这些架构选择旨在解决卷积神经网络中的关键缺陷（缺乏旋转不变性和空间层次理解），但偏离了神经网络的典型架构。因此，尽管胶囊网络的操作可以在CPU上合理实现，但在像GPU和TPU这样的加速器上，性能却急剧下降，因为这些加速器过于优化了矩阵乘法。

无论你是否同意胶囊网络是计算机视觉的未来，作者提到了一些关于在领域专用硬件上训练新型图像分类架构的困难的有趣观点。硬件设计优先考虑满足商业用例，而内置的灵活性以适应下一代研究思想仍然是一个遥远的次要考虑。

虽然专业化使深度神经网络更高效，但它也使偏离公认构建模块的成本大大增加。这引发了一个问题：研究人员在多大程度上会隐性地过度拟合那些在现有硬件上运行良好的想法，而不是冒险尝试那些目前不可行的想法？我们仍然没有硬件和软件来看到的成功是什么样的失败？

5 未来硬件抽奖的可能性

> 我们面前的是一些伪装成无法解决的问题的令人惊叹的机会。
>
> *John Gardner, 1965.*

在机器学习社区中，关于未来算法将与深度神经网络等模型有多大不同的讨论仍在进行中（Sutton, 2019; Welling, 2019）。你对依赖领域专用硬件的风险与您在这一辩论中的立场密切相关。重注于

specialized hardware makes sense if you think that future breakthroughs depend upon pairing deep neural networks with ever increasing amounts of data and computation.

Several major research labs are making this bet, engaging in a "bigger is better" race in the number of model parameters and collecting ever more expansive datasets. However, it is unclear whether this is sustainable. An algorithms scalability is often thought of as the performance gradient relative to the available resources. Given more resources, how does performance increase?

For many subfields, we are now in a regime where the rate of return for additional parameters is decreasing (Thompson et al., 2020a; Brown et al., 2020). For example, while the parameters almost double between Inception V3 (Szegedy et al., 2016)and Inception V4 architectures (Szegedy et al., 2015a) (from 21.8 to 41.1 million parameters), accuracy on ImageNet differs by less than 2% between the two networks (78.8 vs 80 %) (Kornblith et al., 2018). The cost of throwing additional parameters at a problem is becoming painfully obvious. The training of GPT-3 alone is estimated to exceed $12 million dollars (Wiggers, 2020).

Perhaps more troubling is how far away we are from the type of intelligence humans demonstrate. Human brains despite their complexity remain extremely energy efficient. Our brain has over 85 billion neurons but runs on the energy equivalent of an electric shaver (Sainani, 2017). While deep neural networks may be scalable, it may be prohibitively expensive to do so in a regime of comparable intelligence to humans. An apt metaphor is that we appear to be trying to build a ladder to the moon.

Biological examples of intelligence differ from deep neural networks in enough ways to suggest it is a risky bet to say that deep neural networks are the only way forward. While algorithms like deep neural networks rely on global updates in order to learn a useful representation, our brains do not. Our own intelligence relies on decentralized local updates which surface a global signal in ways that are still not well understood (Lillicrap & Santoro, 2019; Marblestone et al., 2016; Bi & Poo, 1998).

In addition, our brains are able to learn efficient representations from far fewer labelled examples than deep neural networks (Zador, 2019). For typical deep learning models the entire model is activated for every example which leads to a quadratic blow-up in training cost. In contrast, evidence suggests that the brain does not perform a full forward and backward pass for all inputs. Instead, the brain simulates what inputs are expected against incoming sensory data. Based upon the certainty of the match, the brain simply infills. What we see is largely virtual reality computed from memory (Eagleman & Sejnowski, 2000; Bubic et al., 2010; Heeger, 2017).

Figure 6: Human latency for certain tasks suggests we have specialized pathways for different stimuli. For example, it is easy for a human to walk and talk at the same time. However, it is far more cognitively taxing to attempt to read and talk.

Humans have highly optimized and specific pathways developed in our biological hardware for different tasks (Von Neumann et al., 2000; Marcus et al., 2014; Kennedy, 2000). For example, it is easy for a human to walk and talk at the same time. However, it is far more cognitively taxing to attempt to read and talk (Stroop, 1935). This suggests the way a network is organized and our inductive biases is as important as the overall size of the network (Herculano-Houzel et al., 2014; Battaglia et al., 2018; Spelke & Kinzler, 2007). Our brains are able to fine-tune and retain skills across our lifetime (Benna & Fusi, 2016; Bremner et al., 2013; Stein et al., 2004; Tani & Press, 2016; Gallistel & King, 2009; Tulving, 2002; Barnett & Ceci, 2002). In contrast, deep neural networks that are trained upon new data often evidence

专业硬件是有意义的，如果你认为未来的突破依赖于将深度神经网络与不断增加的数据和计算能力相结合。

几个主要的研究实验室正在进行这一赌注，参与“更大更好”的模型参数数量竞赛，并收集越来越广泛的数据集。然而，目前尚不清楚这是否可持续。算法的可扩展性通常被视为相对于可用资源的性能梯度。给定更多资源，性能如何提高?

对于许多子领域，我们现在处于一个额外参数的回报率正在下降的状态（Thompson et al., 2020a; Brown et al., 2020）。例如，尽管在 Inception V3 (Szegedy et al., 2016) 和 Inception V4 架构 (Szegedy et al., 2015a) 之间，参数几乎翻倍（从 21.8 到 41.1 百万个参数），但这两个网络在 ImageNet 上的准确率差异不到 2%（78.8% 对 80%）（Kornblith et al., 2018）。向一个问题投入额外参数的成本变得显而易见。仅 GPT-3 的训练估计就超过 1200 万美元（Wiggers, 2020）。

或许更令人担忧的是，我们与人类所展示的智力类型相距多远。尽管人脑复杂，但仍然极其节能。我们的大脑有超过850亿个神经元，但其能耗相当于一台电动剃须刀的能量（Sainani，2017）。虽然深度神经网络可能具有可扩展性，但在与人类相当的智力水平下，这样做可能代价高昂。一个恰当的比喻是，我们似乎在试图建造一架通往月球的梯子。

生物智能的例子与深度神经网络在许多方面存在差异，这表明认为深度神经网络是唯一的前进方向是一种冒险的赌注。虽然像深度神经网络这样的算法依赖于全局更新以学习有用的表示，但我们的大脑并不依赖于此。我们自己的智能依赖于去中心化的局部更新，这些更新以仍然不太理解的方式呈现出全局信号（Lillicrap & Santoro, 2019; Marblestone et al., 2016; Bi & Poo, 1998）。

此外，我们的大脑能够从远少于深度神经网络的标记示例中学习有效的表示（Zador，2019）。对于典型的深度学习模型，

图6：人类在某些任务上的延迟表明我们对不同刺激有专门的通路。例如，人类很容易同时走路和说话。然而，尝试阅读和说话则在认知上要困难得多。

整个模型在每个示例中都被激活，这导致训练成本呈二次增长。相反，证据表明大脑并不会对所有输入执行完整的前向和反向传递。相反，大脑会根据即将到来的感官数据模拟预期的输入。根据匹配的确定性，大脑简单地进行填充。我们所看到的在很大程度上是从记忆中计算出的虚拟现实（Eagleman & Sejnowski, 2000; Bubic et al., 2010; Heeger, 2017）。

人类在我们的生物硬件中为不同任务开发了高度优化和特定的路径（Von Neumann et al., 2000; Marcus et al., 2014; Kennedy, 2000）。例如，人类同时走路和说话是很容易的。然而，尝试阅读和说话则在认知上要困难得多（Stroop, 1935）。这表明，网络的组织方式和我们的归纳偏见与网络的整体规模同样重要（Herculano-Houzel et al., 2014; Battaglia et al., 2018; Spelke & Kinzler, 2007）。我们的脑能够在一生中微调和保留技能（Benna & Fusi, 2016; Bremner et al., 2013; Stein et al., 2004; Tani & Press, 2016; Gallistel & King, 2009; Tulving, 2002; Barnett & Ceci, 2002）。相比之下，基于新数据训练的深度神经网络往往表现出

catastrophic forgetting, where performance deteriorates on the original task because the new information interferes with previously learned behavior (Mcclelland et al., 1995; McCloskey & Cohen, 1989; Parisi et al., 2018).

The point of these examples is not to convince you that deep neural networks are not the way forward. But, rather that there are clearly other models of intelligence which suggest it may not be the only way. It is possible that the next breakthrough will require a fundamentally different way of modelling the world with a different combination of hardware, software and algorithm. We may very well be in the midst of a present day hardware lottery.

6 The Way Forward

> Any machine coding system should be judged quite largely from the point of view of how easy it is for the operator to obtain results.
>
> *John Mauchly, 1973.*

Scientific progress occurs when there is a confluence of factors which allows the scientist to overcome the "stickyness" of the existing paradigm. The speed at which paradigm shifts have happened in AI research have been disproportionately determined by the degree of alignment between hardware, software and algorithm. Thus, any attempt to avoid hardware lotteries must be concerned with making it cheaper and less time-consuming to explore different hardware-software-algorithm combinations.

This is easier said than done. Expanding the search space of possible hardware-software-algorithm combinations is a formidable goal. It is expensive to explore new types of hardware, both in terms of time and capital required. Producing a next generation chip typically costs $30-80 million dollars and takes 2-3 years to develop (Feldman, 2019). The fixed costs alone of building a manufacturing plant are enormous; estimated at $7 billion dollars in 2017 (Thompson & Spanuth, 2018).

Experiments using reinforcement learning to optimize chip placement may help decrease cost (Mirhoseini et al., 2020). There is also renewed interest in re-configurable hardware such as field program gate array (FPGAs) (Hauck & DeHon, 2007) and coarse-grained reconfigurable arrays (CGRAs) (Prabhakar et al., 2017). These devices allow the chip logic to be re-configured to avoid being locked into a single use case. However, the trade-off for flexibility is far higher FLOPS and the need for tailored software development. Coding even simple algorithms on FPGAs remains very painful and time consuming (Shalf, 2020).

In the short to medium term hardware development is likely to remain expensive. The cost of producing hardware is important because it determines the amount of risk and experimentation hardware developers are willing to tolerate. Investment in hardware tailored to deep neural networks is assured because neural networks are a cornerstone of enough commercial use cases. The widespread profitability of deep learning has spurred a healthy ecosystem of hardware startups that aim to further accelerate deep neural networks (Metz, 2018) and has encouraged large companies to develop custom hardware in-house (Falsafi et al., 2017; Jouppi et al., 2017; Lee & Wang, 2018).

The bottleneck will continue to be funding hardware for use cases that are not immediately commercially viable. These more risky directions include biological hardware (Tan et al., 2007; Macía & Sole, 2014; Kriegman et al., 2020), analog hardware with in-memory computation (Ambrogio et al., 2018), neuromorphic computing (Davies, 2019), optical computing (Lin et al., 2018), and quantum computing based approaches (Cross et al., 2019). There are also high risk efforts to explore the development of transistors using new materials (Colwell, 2013; Nikonov & Young, 2013).

Lessons from previous hardware lotteries suggest that investment must be sustained and come from both private and public funding programs. There is a slow awakening of public interest in providing such dedicated resources, such as the 2018 DARPA Electronics Resurgence Initiative which has committed to 1.5 billion dollars in funding for microelectronic technology research (DARPA, 2018). China has also announced a 47 billion dollar fund to support semiconductor research (Kubota, 2018). However, even investment of this magnitude may still be woefully inadequate, as hardware based on new materials requires long lead times of 10-20 years and public investment is currently far below industry levels of R&D (Shalf, 2020).

灾难性遗忘，指的是由于新信息干扰了先前学习的行为，导致在原始任务上的表现下降（Mcclelland et al., 1995; McCloskey & Cohen, 1989; Parisi et al., 2018）。

这些例子的重点并不是要说服你深度神经网络不是前进的方向。而是显然还有其他智能模型，这表明它可能不是唯一的方式。下一个突破可能需要一种根本不同的方式来建模世界，结合不同的硬件、软件和算法。我们很可能正处于当今硬件彩票的中间。

6 前进的道路

> 任何机器编码系统都应该从操作员获得结果的难易程度来进行评判。
>
> *John Mauchly, 1973.*

科学进步发生在多种因素的汇聚下，这使得科学家能够克服现有范式的“粘性”。在人工智能研究中，范式转变发生的速度在很大程度上取决于硬件、软件和算法之间的对齐程度。因此，任何试图避免硬件彩票的努力都必须关注于使探索不同硬件-软件-算法组合变得更便宜和更省时。

这说起来容易，做起来难。扩展可能的硬件-软件-算法组合的搜索空间是一个艰巨的目标。探索新类型的硬件既耗时又耗资。生产下一代芯片通常需要3000万到8000万美元，并且需要2到3年的开发时间（Feldman，2019）。仅建造一个制造工厂的固定成本就非常庞大；2017年的估计为70亿美元（Thompson & Spanuth，2018）。

使用强化学习优化芯片布局的实验可能有助于降低成本（Mirhoseini 等，2020）。对可重构硬件的兴趣也重新燃起，例如现场可编程门阵列（FPGAs）（Hauck & DeHon，2007）和粗粒度可重构阵列（CGRAs）（Prabhakar）。et al., 2017)。这些设备允许芯片逻辑重新配置，以避免被锁定在单一用例中。然而，灵活性的权衡是更高的FLOPS和对定制软件开发的需求。在FP-GAs上编码甚至简单的算法仍然非常痛苦且耗时（Shalf, 2020）。

在短期到中期，硬件开发可能仍然昂贵。生产硬件的成本很重要，因为它决定了硬件开发者愿意容忍的风险和实验量。针对深度神经网络定制的硬件投资是有保障的，因为神经网络是足够商业用例的基石。深度学习的广泛盈利能力促进了一个健康的硬件初创企业生态系统，这些初创企业旨在进一步加速深度神经网络（Metz, 2018），并鼓励大型公司在内部开发定制硬件（Falsafi et al., 2017; Jouppi et al., 2017; Lee & Wang, 2018）。

瓶颈将继续是为那些尚未立即具有商业可行性的用例提供硬件资金。这些风险较高的方向包括生物硬件（Tan et al., 2007; Macía & Sole, 2014; Kriegman et al., 2020）、具有内存计算的模拟硬件（Ambrogio et al., 2018）、类脑计算（Davies, 2019）、光计算（Lin et al., 2018）以及基于量子计算的方法（Cross et al., 2019）。还有高风险的努力在探索使用新材料开发晶体管（Colwell, 2013; Nikonov & Young, 2013）。

来自以往硬件抽奖的经验表明，投资必须持续，并且必须来自私人和公共资金项目。公众对提供此类专用资源的兴趣正在缓慢觉醒，例如2018年DARPA电子复兴倡议已承诺为微电子技术研究提供15亿美元的资金（DARPA，2018）。中国也宣布了一项470亿美元的基金以支持半导体研究（久保田，2018）。然而，即使是如此规模的投资也可能仍然严重不足，因为基于新材料的硬件需要10-20年的长周期，而公共投资目前远低于行业的研发水平（Shalf，2020）。

Figure 7: Byte magazine cover, March 1979, volume 4. Hardware design remains risk adverse due to the large amount of capital and time required to fabricate each new generation of hardware.

6.1 A Software Revolution

An interim goal should be to provide better feedback loops to researchers about how our algorithms interact with the hardware we do have. Machine learning researchers do not spend much time talking about how hardware chooses which ideas succeed and which fail. This is primarily because it is hard to quantify the cost of being concerned. At present, there are no easy and cheap to use interfaces to benchmark algorithm performance against multiple types of hardware at once. There are frustrating differences in the subset of software operations supported on different types of hardware which prevent the portability of algorithms across hardware types (Hotel et al., 2014). Software kernels are often overly optimized for a specific type of hardware which causes large discrepencies in efficiency when used with different hardware (Hennessy, 2019).

These challenges are compounded by an ever more formidable and heterogeneous hardware landscape (Reddi et al., 2020; Fursin et al., 2016). As the hardware landscape becomes increasingly fragmented and specialized, fast and efficient code will require ever more niche and specialized skills to write (Lee et al., 2011). This means that there will be increasingly uneven gains from progress in computer science research. While some types of hardware will benefit from a healthy software ecosystem, progress on other languages will be sporadic and often stymied by a lack of critical end users (Thompson & Spanuth, 2018; Leiserson et al., 2020).

One way to mitigate this need for specialized software expertise is to focus on the development of domain-specific languages which cater to a narrow domain. While you give up expressive power, domain-specific languages permit greater portability across different types of hardware. It allows developers to focus on the intent of the code without worrying about implementation details. (Olukotun, 2014; Mernik et al., 2005; Cong et al., 2011). Another promising direction is automatically auto-tuning the algorithmic parameters of a program based upon the downstream choice of hardware. This facilitates easier deployment by tailoring the program to achieve good performance and load balancing on a variety of hardware (Dongarra et al., 2018; Clint Whaley et al., 2001; Asanović et al., 2006; Ansel et al., 2014).

The difficulty of both these approaches is that if successful, this further abstracts humans from the details of the implementation. In parallel, we need better profiling tools to allow researchers to have a more informed opinion about how hardware and software should evolve. Ideally, software could even surface recommendations about what type of hardware to use given the configuration of an algorithm. Registering what differs from our expectations remains a key catalyst in driving new scientific discoveries.

Software needs to do more work, but it is also well positioned to do so. We have neglected efficient software throughout the era of Moore's law, trusting that predictable gains in compute would compensate for inefficiencies in the software stack. This means there are many low hanging fruit as we begin to optimize for more efficient code (Larus, 2009; Xu et al., 2010).

7 Conclusion

George Gilder, an American investor, described the computer chip as "inscribing worlds on grains of sand" (Gilder, 2000). The performance of an algorithm is fundamentally intertwined with the hardware and software it runs on. This essay proposes the term hardware lottery to describe how these downstream choices determine whether

图7：Byte杂志封面，1979年3月，第4卷。由于制造每一代新硬件所需的大量资本和时间，硬件设计仍然保持风险规避。

6.1 软件革命

一个中期目标应该是为研究人员提供更好的反馈循环，让他们了解我们的算法如何与我们现有的硬件互动。机器学习研究人员并没有花太多时间讨论硬件如何选择哪些想法成功，哪些失败。这主要是因为很难量化关注的成本。目前，没有简单且便宜的接口可以同时对多种类型的硬件进行算法性能基准测试。不同类型硬件上支持的软件操作子集存在令人沮丧的差异，这阻碍了算法在硬件类型之间的可移植性（Hotel et al., 2014）。软件内核通常针对特定类型的硬件进行了过度优化，这在与不同硬件一起使用时会导致效率上的巨大差异（Hennessy, 2019）。

这些挑战因日益强大且异质的硬件环境而加剧（Reddi 等，2020；Fursin 等，2016）。随着硬件环境变得越来越碎片化和专业化，快速高效的代码将需要越来越多的细分和专业技能来编写（Lee 等，2011）。这意味着在进步中将会有越来越不均衡的收益。

计算机科学研究。虽然某些类型的硬件将受益于健康的软件生态系统，但其他语言的进展将是零星的，并且常常因缺乏关键终端用户而受到阻碍（Thompson & Spanuth, 2018; Leiserson et al., 2020）。

减轻对专业软件专业知识需求的一种方法是专注于开发针对狭窄领域的领域特定语言。虽然你放弃了表达能力，但领域特定语言允许在不同类型的硬件之间实现更大的可移植性。它使开发人员能够专注于代码的意图，而不必担心实现细节。（Olukotun, 2014; Mernik et al., 2005; Cong et al., 2011）。另一个有前景的方向是根据下游硬件的选择自动调整程序的算法参数。这通过定制程序以在各种硬件上实现良好的性能和负载平衡，从而促进了更容易的部署。（Dongarra et al., 2018; Clint Whaley et al., 2001; Asanovi et al., 2006; Ansel et al., 2014）。

这两种方法的困难在于，如果成功，这将进一步使人类与实现的细节抽象化。同时，我们需要更好的分析工具，以便研究人员能够对硬件和软件应该如何演变有更明智的看法。理想情况下，软件甚至可以根据算法的配置提出关于使用何种硬件的建议。记录与我们期望的不同之处仍然是推动新科学发现的关键催化剂。

软件需要做更多的工作，但它也处于良好的位置来做到这一点。在摩尔定律时代，我们忽视了高效软件，信任可预测的计算增益能够弥补软件栈中的低效。这意味着当我们开始优化更高效的代码时，有许多触手可及的机会（Larus, 2009; Xu et al., 2010）。

7 结论

乔治·吉尔德，一位美国投资者，将计算机芯片描述为“在沙粒上铭刻世界”（吉尔德，2000）。算法的性能与其运行的硬件和软件基本上是交织在一起的。本文提出“硬件彩票”这一术语，以描述这些下游选择如何决定是否

a research idea succeeds or fails. Today the hardware landscape is increasingly heterogeneous. This essay posits that the hardware lottery has not gone away, and the gap between the winners and losers will grow increasingly larger. In order to avoid future hardware lotteries, we need to make it easier to quantify the opportunity cost of settling for the hardware and software we have.

8 Acknowledgments

Thank you to many of my wonderful colleagues and peers who took time to provide valuable feedback on earlier versions of this essay. In particular, I would like to acknowledge the valuable input of Utku Evci, Erich Elsen, Melissa Fabros, Amanda Su, Simon Kornblith, Cliff Young, Eric Jang, Sean McPherson, Jonathan Frankle, Carles Gelada, David Ha, Brian Spiering, Samy Bengio, Stephanie Sher, Jonathan Binas, Pete Warden, Sean Mcpherson, Laura Florescu, Jacques Pienaar, Chip Huyen, Raziel Alvarez, Dan Hurt and Kevin Swersky. Thanks for the institutional support and encouragement of Natacha Mainville and Alexander Popper.

一个研究想法的成功与否。今天，硬件环境越来越异质化。本文认为，硬件彩票并没有消失，赢家和输家之间的差距将越来越大。为了避免未来的硬件彩票，我们需要更容易地量化满足于我们拥有的硬件和软件的机会成本。

8 致谢

感谢我许多出色的同事和同行，他们抽出时间对这篇文章的早期版本提供了宝贵的反馈。特别感谢 Utku Evci、Erich Elsen、Melissa Fabros、Amanda Su、Simon Kornblith、Cliff Young、Eric Jang、Sean McPherson、Jonathan Frankle、Carles Gelada、David Ha、Brian Spiering、Samy Bengio、Stephanie Sher、Jonathan Binas、Pete Warden、Sean Mcpherson、Laura Flo- rescu、Jacques Pienaar、Chip Huyen、Ra- ziel Alvarez、Dan Hurt 和 Kevin Swersky 的宝贵意见。感谢 Natacha Mainville 和 Alexander Popper 的机构支持和鼓励。

References

Ambrogio, S., Narayanan, P., Tsai, H., Shelby, R., Boybat, I., Nolfo, C., Sidler, S., Giordano, M., Bodini, M., Farinha, N., Killeen, B., Cheng, C., Jaoudi, Y., and Burr, G. Equivalent-accuracy accelerated neural-network training using analogue memory. *Nature*, 558, 06 2018. doi: 10.1038/s41586-018-0180-5.

Amodei, D., Hernandez, D., Sastry, G., Clark, J., Brockman, G., and Sutskever, I. Ai and compute, 2018. URL https://openai.com/blog/ai-and-compute/.

Ansel, J., Kamil, S., Veeramachaneni, K., Ragan-Kelley, J., Bosboom, J., O'Reilly, U.-M., and Amarasinghe, S. Opentuner: An extensible framework for program autotuning. In *Proceedings of the 23rd International Conference on Parallel Architectures and Compilation*, PACT '14, pp. 303–316, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450328098. doi: 10.1145/2628071.2628092. URL https://doi.org/10.1145/2628071.2628092.

ARM. Enhancing ai performance for iot endpoint devices, 2020. URL https://www.arm.com/company/news/2020/02/new-ai-technology-from-arm.

Asanovic, K. Accelerating ai: Past, present, and future, 2018. URL https://www.youtube.com/watch?v=8n2HLp2gtYs&t=2116s.

Asanović, K., Bodik, R., Catanzaro, B. C., Gebis, J. J., Husbands, P., Keutzer, K., Patterson, D. A., Plishker, W. L., Shalf, J., Williams, S. W., and Yelick, K. A. The landscape of parallel computing research: A view from berkeley. Technical Report UCB/EECS-2006-183, EECS Department, University of California, Berkeley, Dec 2006. URL http://www2.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-183.html.

Barham, P. and Isard, M. Machine learning systems are stuck in a rut. In *Proceedings of the Workshop on Hot Topics in Operating Systems*, HotOS '19, pp. 177–183, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367271. doi: 10.1145/3317550.3321441. URL https://doi.org/10.1145/3317550.3321441.

Barnett, S. M. and Ceci, S. When and where do we apply what we learn? a taxonomy for far transfer. *Psychological bulletin*, 128 4:612–37, 2002.

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V. F., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., Gülçehre, Ç., Song, H. F., Ballard, A. J., Gilmer, J., Dahl, G. E., Vaswani, A., Allen, K. R., Nash, C., Langston, V., Dyer, C., Heess, N., Wierstra, D., Kohli, P., Botvinick, M., Vinyals, O., Li, Y., and Pascanu, R. Relational inductive biases, deep learning, and graph networks. *CoRR*, abs/1806.01261, 2018. URL http://arxiv.org/abs/1806.01261.

Benna, M. and Fusi, S. Computational principles of synaptic memory consolidation. *Nature Neuroscience*, 19, 10 2016. doi: 10.1038/nn.4401.

Bi, G.-q. and Poo, M.-m. Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of Neuroscience*, 18(24):10464–10472, 1998. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.18-24-10464.1998. URL https://www.jneurosci.org/content/18/24/10464.

Bremner, A., Lewkowicz, D., and Spence, C. Multisensory development, 11 2013.

Brodtkorb, A. R., Hagen, T. R., and Sætra, M. L. Graphics processing unit (gpu) programming strategies and trends in gpu computing. *Journal of Parallel and Distributed Computing*, 73(1):4 – 13, 2013. ISSN 0743-7315. doi: https://doi.org/10.1016/j.jpdc.2012.04.003. URL http://www.sciencedirect.com/science/article/pii/S0743731512000998. Metaheuristics on GPUs.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCand lish, S., Radford, A., Sutskever, I., and Amodei, D. Language Models are Few-Shot Learners. *arXiv e-prints*, May 2020.

Bubic, A., Von Cramon, D. Y., and Schubotz, R. Prediction, cognition and the brain. *Frontiers in Human Neuroscience*, 4:25, 2010. ISSN 1662-5161. doi: 10.3389/fnhum.2010.00025. URL https://www.frontiersin.org/article/10.3389/fnhum.2010.00025.

参考文献

Ambrogio, S., Narayanan, P., Tsai, H., Shelby, R., Boybat, I., Nolfo, C., Sidler, S., Giordano, M., Bodini, M., Farinha, N., Killeen, B., Cheng, C., Jaoudi, Y., 和 Burr, G. 使用模拟内存的等效精度加速神经网络训练。*Nature*, 558, 2018年6月。doi: 10.1038/s41586-018-0180-5。Amodei, D., Hernandez, D., Sastry, G., Clark, J., Brockman, G., 和 Sutskever, I. 人工智能与计算，2018。网址 https://openai.com/blog/ai-and-compute/。Ansel, J., Kamil, S., Veeramachaneni, K., Ragan-Kelley, J., Bosboom, J., O'Reilly, U.-M., 和 Amarasinghe, S. Opentuner：一个可扩展的程序自动调优框架。在 *Proceedings of the 23rd International Conference on Parallel Architectures and Compilation*，PACT'14，第303–316页，美国纽约，2014年。计算机协会。ISBN 9781450328098。doi: 10.1145/2628071.2628092。网址 https://doi.org/10.1145/2628071.2628092。ARM。增强物联网终端设备的人工智能性能，2020。网址 https://www.arm.com/company/news/2020/02/new-ai-technology-from-arm。Asanovic, K. 加速人工智能：过去、现在和未来，2018。网址 https://www.youtube.com/watch?v=8n2HLp2gtYs&t=2116s。Asanović, K., Bodik, R., Catanzaro, B. C., Gebis, J. J., Husbands, P., Keutzer, K., Patterson, D. A., Plishker, W. L., Shalf, J., Williams, S. W., 和 Yelick, K. A. 并行计算研究的现状：来自伯克利的视角。技术报告 UCB/EECS-2006-183，加州大学伯克利分校电子工程与计算机科学系，2006年12月。网址 http://www2.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-183.html。Barham, P. 和 Isard, M. 机器学习系统陷入困境。在 *Proceedings of the Workshop on Hot Topics in Operating Systems*，HotOS'19，第177–183页，美国纽约，2019年。计算机协会。ISBN 9781450367271。doi: 10.1145/3317550.3321441。网址 https://doi.org/10.1145/3317550.3321441。Barnett, S. M. 和 Ceci, S. 我们何时何地应用所学？远程转移的分类法。*Psychological bulletin*, 128 4:612–37, 2002。

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V. F., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., Gülçehre, Ç., Song, H. F., Ballard, A. J., Gilmer, J., Dahl, G. E., Vaswani, A., Allen, K. R., Nash, C., Langston, V., Dyer, C., Heess, N., Wierstra, D., Kohli, P., Botvinick, M., Vinyals, O., Li, Y., 和 Pascanu, R. 关系归纳偏差、深度学习和图网络。*CoRR*，abs/1806.01261，2018。网址 http://arxiv.org/abs/1806.01261。

Benna, M. 和 Fusi, S. 突触记忆巩固的计算原理。*Na- ture Neuroscience*, 19, 2016年10月. doi: 10.1038/ nn.4401.

Bi, G.-q. 和 Poo, M.-m. 培养的海马神经元中的突触修饰：依赖于尖峰时序、突触强度和突触后细胞类型。 *Journal of Neu- roscience*, 18(24):10464–10472, 1998. ISSN 0270-6474. doi: 10.1523/JNEUROSCI. 18-24-10464.1998. URL https://www. jneurosci.org/content/18/24/10464.

布雷姆纳，A.，莱科维茨，D.，和斯宾塞，C. 多感官发展，2013年11月。

Brodtkorb, A. R., Hagen, T. R., 和 Sætra, M. L. 图形处理单元（gpu）编程策略及gpu计算中的趋势。 *Journal of Parallel and Distributed Computing*, 73(1):4 – 13, 2013. ISSN 0743-7315. doi: https://doi.org/ 10.1016/j.jpdc.2012.04.003. URL http: //www.sciencedirect.com/science/ article/pii/S0743731512000998. 在GPU上的元启发式算法。

布朗，T. B.，曼恩，B.，赖德，N.，苏比亚，M.，卡普兰，J.，达里瓦尔，P.，尼拉坎坦，A.，夏姆，P.，萨斯特里，G.，阿斯凯尔，A.，阿加瓦尔，S.，赫伯特-沃斯，A.，克鲁格，G.，亨尼根，T.，查尔德，R.，拉梅什，A.，齐格勒，D. M.，吴，J.，温特，C.，赫塞，C.，陈，M.，西格勒，E.，利特温，M.，格雷，S.，切斯，B.，克拉克，J.，伯纳，C.，麦肯德利什，S.，拉德福德，A.，萨茨克弗，I.，和阿莫代伊，D. 语言模型是少量学习者。*arXiv e-prints*，2020年5月。

Bubic, A., Von Cramon, D. Y., 和 Schubotz, R. 预测、认知与大脑。 *Frontiers in Human Neuroscience*, 4:25, 2010. ISSN 1662-5161. doi: 10.3389/fnhum.2010.00025. URL https://www.frontiersin.org/article/10.3389/fnhum.2010.00025.

Chellapilla, K., Puri, S., and Simard, P. High performance convolutional neural networks for document processing, 10 2006.

CHM. Moore's law, 2020. URL https://www.computerhistory.org/revolution/digital-logic/12/267.

Ciresan, D., Meier, U., Masci, J., Gambardella, L. M., and Schmidhuber, J. Flexible, high performance convolutional neural networks for image classification. *International Joint Conference on Artificial Intelligence IJCAI-2011*, pp. 1237–1242, 07 2011. doi: 10.5591/978-1-57735-516-8/IJCAI11-210.

Claudiu Ciresan, D., Meier, U., Gambardella, L. M., and Schmidhuber, J. Deep Big Simple Neural Nets Excel on Handwritten Digit Recognition. *arXiv e-prints*, art. arXiv:1003.0358, March 2010.

Clint Whaley, R., Petitet, A., and Dongarra, J. J. Automated empirical optimizations of software and the atlas project. *Parallel Computing*, 27(1):3 – 35, 2001. ISSN 0167-8191. doi: https://doi.org/10.1016/S0167-8191(00)00087-9. URL http://www.sciencedirect.com/science/article/pii/S0167819100000879. New Trends in High Performance Computing.

Coates, A., Huval, B., Wang, T., Wu, D., Catanzaro, B., and Andrew, N. Deep learning with cots hpc systems. In Dasgupta, S. and McAllester, D. (eds.), *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 1337–1345, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL http://proceedings.mlr.press/v28/coates13.html.

Collier, B. *Little Engines That Could've: The Calculating Machines of Charles Babbage*. Garland Publishing, Inc., USA, 1991. ISBN 0824000439.

Collobert, R., Bengio, S., and Marithoz, J. Torch: A modular machine learning software library, 11 2002.

Colwell, R. The chip design game at the end of moore's law. In *2013 IEEE Hot Chips 25 Symposium (HCS)*, pp. 1–16, 2013.

Cong, J., Sarkar, V., Reinman, G., and Bui, A. Customizable domain-specific computing. *IEEE Design Test of Computers*, 28(2):6–15, 2011.

Cross, A. W., Bishop, L. S., Sheldon, S., Nation, P. D., and Gambetta, J. M. Validating quantum computers using randomized model circuits, September 2019.

DARPA. Darpa announces next phase of electronics resurgence initiative, 2018. URL https://www.darpa.mil/news-events/2018-11-01a.

Davies, M. Progress in neuromorphic computing : Drawing inspiration from nature for gains in ai and computing. In *2019 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, pp. 1–1, 2019.

Dean, J. Parallel implementations of neural network training: two back-propagation approaches., 1990. URL https://drive.google.com/file/d/1I1fs4sczbCaACzA9XwxR3DiuXVtqmejL/view.

Dean, J. 1.1 the deep learning revolution and its implications for computer architecture and chip design. *2020 IEEE International Solid-State Circuits Conference - (ISSCC)*, pp. 8–14, 2020.

Demuth, H. and Beale, M. Neural network toolbox for use with matlab - user guide verion 3.0, 1993.

Dennard, R. H., Gaensslen, F. H., Yu, H., Rideout, V. L., Bassous, E., and LeBlanc, A. R. Design of ion-implanted mosfet's with very small physical dimensions. *IEEE Journal of Solid-State Circuits*, 9(5):256–268, 1974.

Dettmers, T. Which gpu for deep learning?, 2020. URL https://bit.ly/35qq8xe.

Diamond, J., Diamond, P., and Collection, B. H. *Guns, Germs, and Steel: The Fates of Human Societies*. National bestseller / W.W. Norton & Company. W.W. Norton, 1999. ISBN 9780393317558. URL https://books.google.com/books?id=1lBu_bqSsSMC.

Dongarra, J., Gates, M., Kurzak, J., Luszczek, P., and Tsai, Y. M. Autotuning numerical dense linear algebra for batched computation with gpu hardware accelerators. *Proceedings of the IEEE*, 106(11):2040–2055, 2018.

Eagleman, D. M. and Sejnowski, T. J. Motion integration and postdiction in visual awareness. *Science*, 287(5460):2036–2038, 2000. ISSN 0036-8075. doi: 10.1126/science.287.5460.2036. URL

Chellapilla, K., Puri, S., 和 Simard, P. 高性能卷积神经网络用于文档处理, 2006年10月。

CHM. 摩尔定律，2020。网址 https://www.computerhistory.org/ revolution/digital-logic/12/267。

Ciresan, D., Meier, U., Masci, J., Gambardella, L. M., 和 Schmidhuber, J. 灵活的高性能卷积神经网络用于图像分类。 *International Joint Conference on Artificial Intelligence IJCAI-2011*, 第 1237–1242 页, 2011 年 7 月. doi: 10.5591/ 978-1-57735-516-8/IJCAI11-210.

Claudiu Ciresan, D., Meier, U., Gambardella, L. M., 和 Schmidhuber, J. 深度大简单神经网络在手写数字识别中表现出色。*arXiv e-prints*，文章 arXiv:1003.0358，2010年3月。

Clint Whaley, R., Petitet, A., 和 Dongarra, J. J. 软件的自动经验优化与阿特拉斯项目。*Parallel Computing*, 27(1):3 − 35, 2001. ISSN 0167-8191. doi: https://doi.org/10.1016/ S0167-8191(00)00087-9. URL http: //www.sciencedirect.com/science/ article/pii/S0167819100000879. 高性能计算的新趋势。

Coates, A., Huval, B., Wang, T., Wu, D., Catanzaro, B., 和 Andrew, N. 使用商用高性能计算系统的深度学习。在 Dasgupta, S. 和 McAllester, D. (编辑), *Proceedings of the 30th International Conference on Machine Learning*, *Proceedings of Machine Learning Research* 第28卷, 第1337–1345页, 美国乔治亚州亚特兰大, 2013年6月17–19日. PMLR. URL http://proceedings.mlr.press/v28/coates13.html.

科利尔, B. *Little Engines That Could've: The Calculating Machines of Charles Babbage*. 加兰出版公司, 美国, 1991. ISBN 0824000439.

Collobert, R., Bengio, S., 和 Marithoz, J. Torch：一个模块化的机器学习软件库，2002年11月。

Colwell, R. 摩尔定律结束时的芯片设计游戏。在 *2013 IEEE Hot Chips 25 Symposium (HCS)*，第 1–16 页，2013 年。

Cong, J., Sarkar, V., Reinman, G., 和 Bui, A. 可定制的领域特定计算. *IEEE Design Test of Computers*, 28(2):6–15, 2011.

Cross, A. W., Bishop, L. S., Sheldon, S., Nation, P. D., 和 Gambetta, J. M. 使用随机模型电路验证量子计算机，2019年9月。

DARPA。达尔帕宣布电子复兴计划的下一阶段，2018年。网址 https://www.darpa.mil/news-events/ 2018-11-01a。

戴维斯，M. 神经形态计算的进展：从自然中汲取灵感以提升人工智能和计算能力。在 *2019 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*，第 1–1 页，2019 年。

迪恩，J. 神经网络训练的并行实现：两种反向传播方法，1990年。网址 https://drive.google.com/file/d/ 1I1fs4sczbCaACzA9XwxR3DiuXVtqmejL/ view.

迪恩，J. 1.1 深度学习革命及其对计算机架构和芯片设计的影响。*2020 IEEE International Solid-State Circuits Conference - (ISSCC)*，第8–14页，2020年。

Demuth, H. 和 Beale, M. 用于 MATLAB 的神经网络工具箱 - 用户指南版本 3.0, 1993.

Dennard, R. H., Gaensslen, F. H., Yu, H., Rideout, V. L., Bassous, E., 和 LeBlanc, A. R. 设计具有非常小物理尺寸的离子注入MOSFET。*IEEE Journal of Solid-State Circuits*, 9(5):256–268, 1974.

Dettmers, T. 哪种GPU适合深度学习？，2020。网址 https://bit.ly/35qq8xe。

钻石，J.，钻石，P.，和收藏，B. H. *Guns, Germs, and Steel: The Fates of Human Societies*。全国畅销书 / W.W. 诺顿公司。W. W. 诺顿，1999年。ISBN 9780393317558。网址 https://books.google.com/books?id=1lBu_bqSsSMC。

Dongarra, J., Gates, M., Kurzak, J., Luszczek, P., 和 Tsai, Y. M. 使用 GPU 硬件加速器对批处理计算的数值稠密线性代数进行自调优。*Proceedings of the IEEE*, 106(11):2040–2055, 2018.

Eagleman, D. M. 和 Sejnowski, T. J. 视觉意识中的运动整合和后验推断。*Science*, 287(5460):2036–2038, 2000. ISSN 0036-8075. doi: 10.1126/science.287.5460.2036. URL

https://science.sciencemag.org/content/287/5460/2036.

Elsen, E., Dukhan, M., Gale, T., and Simonyan, K. Fast sparse convnets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Evci, U., Gale, T., Menick, J., Castro, P. S., and Elsen, E. Rigging the Lottery: Making All Tickets Winners. *arXiv e-prints*, November 2019.

Falsafi, B., Dally, B., Singh, D., Chiou, D., Yi, J. J., and Sendag, R. Fpgas versus gpus in data centers. *IEEE Micro*, 37(1):60–72, 2017.

Fatahalian, K., Sugerman, J., and Hanrahan, P. Understanding the efficiency of gpu algorithms for matrix-matrix multiplication. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware*, HWWS '04, pp. 133–137, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 3905673150. doi: 10.1145/1058129.1058148. URL https://doi.org/10.1145/1058129.1058148.

Feldman, M. The era of general purpose computers is ending, 2019. URL https://bit.ly/3hP8XJh.

Fukushima, K. and Miyake, S. Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. *Pattern Recognition*, 15(6):455 – 469, 1982. ISSN 0031-3203. URL http://www.sciencedirect.com/science/article/pii/0031320382900243.

Fursin, G., Lokhmotov, A., and Plowman, E. Collective knowledge: Towards r d sustainability. In *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 864–869, 2016.

Gale, T., Elsen, E., and Hooker, S. The state of sparsity in deep neural networks, 2019.

Gale, T., Zaharia, M., Young, C., and Elsen, E. Sparse GPU Kernels for Deep Learning. *arXiv e-prints*, June 2020.

Gallistel, C. and King, A. Memory and the computational brain: Why cognitive science will transform neuroscience, 04 2009.

Gilder, G. *Telecosm: How Infinite Bandwidth Will Revolutionize Our World*. Free Press, 2000. ISBN 9780743215947. URL https://books.google.com/books?id=Kzo-KTxdwcEC.

Gray, S., Radford, A., and Kingma, D. P. Gpu kernels for block-sparse weights, 2017.

Gupta, S. and Tan, M. Efficientnet-edgetpu: Creating accelerator-optimized neural networks with automl, 2019. URL https://ai.googleblog.com/2019/08/efficientnet-edgetpu-creating.html.

Hauck, S. and DeHon, A. *Reconfigurable Computing: The Theory and Practice of FPGA-Based Computation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2007. ISBN 9780080556017.

Haugeland, J. *Artificial Intelligence: The Very Idea*. Massachusetts Institute of Technology, USA, 1985. ISBN 0262081539.

Heeger, D. J. Theory of cortical function. *Proceedings of the National Academy of Sciences*, 114(8):1773–1782, 2017. ISSN 0027-8424. doi: 10.1073/pnas.1619788114. URL https://www.pnas.org/content/114/8/1773.

Hennessy, J. The end of moore's law, cpus (as we know them), and the rise of domain specific architectures, 2019. URL https://www.kisacoresearch.com/sites/default/files/presentations/09.00_-_alphabet_-_john_hennessy.pdf.

Herculano-Houzel, S., de Souza, K. A., Neves, K., Porfirio, J., Messeder, D. J., Feijó, L. M., Maldonado, J., and Manger, P. R. The elephant brain in numbers. *Frontiers in Neuroanatomy*, 8, 2014.

Hinton, G. E. and Anderson, J. A. *Parallel Models of Associative Memory*. L. Erlbaum Associates Inc., USA, 1989. ISBN 080580269X.

Hooker, S., Courville, A., Clark, G., Dauphin, Y., and Frome, A. What Do Compressed Deep Neural Networks Forget? *arXiv e-prints*, art. arXiv:1911.05248, November 2019.

Horowitz, M. 1.1 computing's energy problem (and what we can do about it). In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 10–14, 2014.

https://science.sciencemag.org/ content/287/5460/2036。Elsen, E., Dukhan, M., Gale, T. 和 Simonyan, K. 快速稀疏卷积网络。在 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*，2020年6月。Evci, U., Gale, T., Menick, J., Castro, P. S. 和 Elsen, E. 操控彩票：让所有票据成为赢家。*arXiv e-prints*，2019年11月。Falsafi, B., Dally, B., Singh, D., Chiou, D., Yi, J. J. 和 Sendag, R. 数据中心中的FPGA与GPU。*IEEE Micro*，37(1):60–72，2017年。Fatahalian, K., Sugerman, J. 和 Hanra- han, P. 理解GPU算法在矩阵-矩阵乘法中的效率。在 *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware*，HWWS '04，第133–137页，美国纽约，2004年。计算机协会。ISBN 3905673150。doi: 10.1145/1058129.1058148。网址 https://doi.org/10.1145/1058129.1058148。Feldman, M. 通用计算机的时代正在结束，2019年。网址 https://bit.ly/3hP8XJh。Fukushima, K. 和 Miyake, S. Neocognitron：一种对变形和位置偏移具有容忍性的模式识别新算法。*Pattern Recognition*，15(6):455 – 469，1982年。ISSN 0031-3203。网址 http://www.sciencedirect.com/science/article/pii/0031320382900243。Fursin, G., Lokhmotov, A. 和 Plowman, E. 集体知识：朝着研发可持续性迈进。在 *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*，第864–869页，2016年。Gale, T., Elsen, E. 和 Hooker, S. 深度神经网络中的稀疏状态，2019年。Gale, T., Zaharia, M., Young, C. 和 Elsen, E. 深度学习的稀疏GPU内核。*arXiv e-prints*，2020年6月。Gallistel, C. 和 King, A. 记忆与计算大脑：为什么认知科学将改变神经科学，2009年4月。

吉尔德，G. *Telecosm: How Infinite Bandwidth Will Revolutionize Our World*。自由出版社，2000年。ISBN 9780743215947。网址 https://books.google.com/books?id= Kzo-KTxdwcEC。

Gray, S., Radford, A., 和 Kingma, D. P. 用于块稀疏权重的 GPU 核心，2017。

Gupta, S. 和 Tan, M. Efficientnet-edgetpu：使用 automl 创建加速器优化的神经网络，2019。网址 https://ai.googleblog.com/2019/08/efficientnet-edgetpu-creating.html。

Hauck, S. 和 DeHon, A. *Reconfigurable Computing: The Theory and Practice of FPGA-Based Computation*。摩根考夫曼出版社，旧金山，加利福尼亚州，美国，2007年。ISBN 9780080556017。

霍格兰德, J. *Artificial Intelligence: The Very Idea*. 麻省理工学院, 美国, 1985. ISBN 026208 1539.

Heeger, D. J. 皮层功能理论。*Proceedings of the National Academy of Sciences*, 114(8):1773–1782, 2017. ISSN 0027-8424. doi: 10.1073/pnas.1619788114. URL https://www.pnas.org/content/114/8/1773.

亨尼西，J. 摩尔定律的终结，CPU（如我们所知）以及领域特定架构的崛起，2019年。网址 https://www.kisacoresearch.com/sites/default/files/presentations/09.00_-_alphabet_-_john_hennessy.pdf。

Herculano-Houzel, S., de Souza, K. A., Neves, K., Porfirio, J., Messeder, D. J., Feijó, L. M., Maldonado, J., 和 Manger, P. R. 大象的大脑数字. *Frontiers in Neu- roanatomy*, 8, 2014.

Hinton, G. E. 和 Anderson, J. A. *Parallel Models of Associative Memory*. L. Erlbaum Associates Inc., 美国, 1989. ISBN 080580269X.

Hooker, S., Courville, A., Clark, G., Dauphin, Y., 和 Frome, A. 压缩深度神经网络遗忘了什么？ *arXiv e-prints*, 文章 arXiv:1911.05248, 2019年11月。

霍洛维茨，M. 1.1 计算的能源问题（以及我们可以做些什么）。在 *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*，第 10–14 页，2014 年。

Hotel, H., Johansen, H., Bernholdt, D., Héroux, M., and Hornung, R. Software productivity for extreme-scale science, 2014.

Howe, D. B. and Asanović, K. *SPACE: Symbolic Processing in Associative Computing Elements*, pp. 243–252. Springer US, Boston, MA, 1994. ISBN 978-1-4899-1331-9. doi: 10.1007/978-1-4899-1331-9_24. URL https://doi.org/10.1007/978-1-4899-1331-9_24.

Isaacson, W. Grace hopper, computing pioneer. *The Harvard Gazette*, 2014. URL https://news.harvard.edu/gazette/story/2014/12/grace-hopper-computing-pioneer/.

Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., Bates, S., Bhatia, S., Boden, N., Borchers, A., Boyle, R., Cantin, P.-l., Chao, C., Clark, C., Coriell, J., Daley, M., Dau, M., Dean, J., Gelb, B., Ghaemmaghami, T. V., Gottipati, R., Gulland, W., Hagmann, R., Ho, C. R., Hogberg, D., Hu, J., Hundt, R., Hurt, D., Ibarz, J., Jaffey, A., Jaworski, A., Kaplan, A., Khaitan, H., Killebrew, D., Koch, A., Kumar, N., Lacy, S., Laudon, J., Law, J., Le, D., Leary, C., Liu, Z., Lucke, K., Lundin, A., MacKean, G., Maggiore, A., Mahony, M., Miller, K., Nagarajan, R., Narayanaswami, R., Ni, R., Nix, K., Norrie, T., Omernick, M., Penukonda, N., Phelps, A., Ross, J., Ross, M., Salek, A., Samadiani, E., Severn, C., Sizikov, G., Snelham, M., Souter, J., Steinberg, D., Swing, A., Tan, M., Thorson, G., Tian, B., Toma, H., Tuttle, E., Vasudevan, V., Walter, R., Wang, W., Wilcox, E., and Yoon, D. H. In-datacenter performance analysis of a tensor processing unit. *SIGARCH Comput. Archit. News*, 45 (2):1–12, June 2017. ISSN 0163-5964. doi: 10.1145/3140659.3080246. URL https://doi.org/10.1145/3140659.3080246.

K, S. and Piske, U. Learning matrices and their applications. *IEEE Transactions on Electronic Computers*, EC-12(6):846–862, 1963.

Kennedy, M. B. Signal-processing machines at the postsynaptic density. *Science*, 290 5492: 750–4, 2000.

Kingsbury, B., Morgan, N., and Wawrzynek, J. Hipnet-1: A highly pipelined architecture for neural network training, 03 1998.

Kornblith, S., Shlens, J., and Le, Q. V. Do better imagenet models transfer better? *CoRR*, abs/1805.08974, 2018. URL http://arxiv.org/abs/1805.08974.

Krashinsky, R., Giroux, O., Jones, S., Stam, N., and Ramaswamy, S. Nvidia ampere architecture in-depth., 2020. URL https://developer.nvidia.com/blog/nvidia-ampere-architecture-in-depth/.

Kriegman, S., Blackiston, D., Levin, M., and Bongard, J. A scalable pipeline for designing reconfigurable organisms. *Proceedings of the National Academy of Sciences*, 117(4): 1853–1859, 2020. ISSN 0027-8424. doi: 10.1073/pnas.1910837117. URL https://www.pnas.org/content/117/4/1853.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks, 2012. URL https://bit.ly/2GneDwp.

Kubota, Y. China plans 47 billion fund to boost its semiconductor industry, 2018. URL https://on.wsj.com/32L7Kwn.

Kuhn, T. S. *The Structure of Scientific Revolutions*. University of Chicago Press, Chicago, 1962.

Kurzweil, R. *The Age of Intelligent Machines*. MIT Press, Cambridge, MA, USA, 1990.

Larus, J. Spending moore's dividend. *Commun. ACM*, 52(5):62–69, May 2009. ISSN 0001-0782. doi: 10.1145/1506409.1506425. URL https://doi.org/10.1145/1506409.1506425.

Le, Q. V., Ranzato, M., Monga, R., Devin, M., Chen, K., Corrado, G. S., Dean, J., and Ng, A. Y. Building high-level features using large scale unsupervised learning. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, pp. 507–514, Madison, WI, USA, 2012. Omnipress. ISBN 9781450312851.

Lecun, Y. and Bottou, L. Technical report: Lush reference manual, code available at http://lush.sourceforge.net, 2002.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. Backpropagation applied to handwritten zip code recognition, 1989. URL https://doi.org/10.1162/neco.1989.1.4.541.

酒店, H., 约翰森, H., 伯恩霍尔特, D., 赫鲁克斯, M., 和 霍尔农, R. 极大规模科学的软件生产力, 2014.

Howe, D. B. 和 Asanović, K. *SPACE: Symbolic Processing in Associative Computing Elements*, 第 243–252 页。斯普林格美国，波士顿，马萨诸塞州，1994年。ISBN 978-1-4899-1331-9。doi: 10.1007/978-1-4899-1331-9_24。网址 https://doi.org/10.1007/978-1-4899-1331-9_24。

艾萨克森，W. 格蕾丝·霍普，计算先锋。*The Harvard Gazette*，2014年。网址 https://news.harvard.edu/gazette/story/2014/12/grace-hopper-computing-pioneer/.

Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., Bates, S., Bha- tia, S., Boden, N., Borchers, A., Boyle, R., Cantin, P.-l., Chao, C., Clark, C., Coriell, J., Daley, M., Dau, M., Dean, J., Gelb, B., Ghaemmaghami, T. V., Gottipati, R., Gulland, W., Hagmann, R., Ho, C. R., Hogberg, D., Hu, J., Hundt, R., Hurt, D., Ibarz, J., Jaffey, A., Jaworski, A., Kaplan, A., Khaitan, H., Killebrew, D., Koch, A., Kumar, N., Lacy, S., Laudon, J., Law, J., Le, D., Leary, C., Liu, Z., Lucke, K., Lundin, A., MacKean, G., Maggiore, A., Mahony, M., Miller, K., Na- garajan, R., Narayanaswami, R., Ni, R., Nix, K., Norrie, T., Omernick, M., Penukonda, N., Phelps, A., Ross, J., Ross, M., Salek, A., Samadiani, E., Severn, C., Sizikov, G., Snel- ham, M., Souter, J., Steinberg, D., Swing, A., Tan, M., Thorson, G., Tian, B., Toma, H., Tut- tle, E., Vasudevan, V., Walter, R., Wang, W., Wilcox, E., 和 Yoon, D. H. 数据中心内张量处理单元的性能分析。*SIGARCH Comput. Archit. News*, 45 (2):1–12, 2017年6月。ISSN 0163-5964。doi: 10.1145/3140659.3080246。网址 https:// doi.org/10.1145/3140659.3080246。

K, S. 和 Piske, U. 学习矩阵及其应用。*IEEE Transactions on Elec- tronic Computers*, EC-12(6):846–862, 1963.

肯尼迪，M. B. 突触后密度的信号处理机器。*Science*，290 5492：750–4，2000。

Kingsbury, B., Morgan, N., 和 Wawrzynek, J. Hipnet-1：一种用于神经网络训练的高度流水线架构，1998年3月。

Kornblith, S.， Shlens, J. 和 Le, Q. V. 更好的 imagenet 模型是否能更好地迁移? *CoRR*, abs/1805.08974, 2018. URL http: //arxiv.org/abs/1805.08974. Krashinsky, R., Giroux, O., Jones, S., Stam, N., 和 Ramaswamy, S. Nvidia 安培架构深入分析., 2020. URL https://developer.nvidia.com/blog/ nvidia-ampere-architecture-in-depth/

抱歉，我无法处理该请

Kriegman, S.， Blackiston, D.， Levin, M. 和 Bongard, J. 可重构生物设计的可扩展管道。*Proceedings of the National Academy of Sciences*，117(4)：1853–1859，2020。ISSN 0027-8424。doi：10.1073/pnas.1910837117。网址 https://www.pnas.org/content/117/4/1853。Krizhevsky, A., Sutskever, I. 和 Hinton, G. E. 使用深度卷积神经网络进行Imagenet分类，2012。网址 https://bit.ly/2GneDwp。Kubota, Y. 中国计划设立470亿美元基金以推动其半导体产业，2018。网址 https://on.wsj.com/32L7Kwn。

库恩，T. S. *The Structure of Scientific Revolutions*。芝加哥大学出版社，芝加哥，1962年。

库兹韦尔, R. *The Age of Intelligent Machines*. 麻省理工学院出版社, 剑桥, 马萨诸塞州, 美国, 1990.

Larus, J. 支出摩尔的红利。*Commun. ACM*, 52(5):62–69, 2009年5月。 ISSN 0001-0782。doi: 10.1145/1506409.1506425。 URL https://doi.org/10.1145/1506409.1506425。

Le, Q. V., Ranzato, M., Monga, R., Devin, M., Chen, K., Corrado, G. S., Dean, J., 和 Ng, A. Y. 使用大规模无监督学习构建高级特征。在 *Proceedings of the 29th International Coference on International Conference on Machine Learning*，ICML’12，第 507–514 页，威斯康星州麦迪逊，美国，2012。Omnipress。ISBN 9781450312851。

Lecun, Y. 和 Bottou, L. 技术报告：Lush 参考手册，代码可在 http://lush.sourceforge.net 获取，2002年。

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., 和 Jackel, L. D. 反向传播应用于手写邮政编码识别, 1989. URL https:// doi.org/10.1162/neco.1989.1.4.541.

Lee, H., Brown, K., Sujeeth, A., Chafi, H., Rompf, T., Odersky, M., and Olukotun, K. Implementing domain-specific languages for heterogeneous parallel computing. *IEEE Micro*, 31(5):42–53, 2011.

Lee, K. and Wang, X. The next step in facebook ai hardware infrastructure, 2018. URL https://bit.ly/3bgZFDn.

Leiserson, C. E., Thompson, N. C., Emer, J. S., Kuszmaul, B. C., Lampson, B. W., Sanchez, D., and Schardl, T. B. There's plenty of room at the top: What will drive computer performance after moore's law? *Science*, 368(6495), 2020. ISSN 0036-8075. doi: 10.1126/science.aam9744. URL https://science.sciencemag.org/content/368/6495/eaam9744.

Lillicrap, T. P. and Santoro, A. Backpropagation through time and the brain. *Current Opinion in Neurobiology*, 55:82 – 89, 2019. ISSN 0959-4388. doi: https://doi.org/10.1016/j.conb.2019.01.011. URL http://www.sciencedirect.com/science/article/pii/S0959438818302009. Machine Learning, Big Data, and Neuroscience.

Lin, X., Rivenson, Y., Yardimci, N. T., Veli, M., Luo, Y., Jarrahi, M., and Ozcan, A. All-optical machine learning using diffractive deep neural networks. *Science*, 361 (6406):1004–1008, 2018. ISSN 0036-8075. doi: 10.1126/science.aat8084. URL https://science.sciencemag.org/content/361/6406/1004.

Lindsey, C. S. and Lindblad, T. Review of hardware neural networks: A User's perspective. In *3rd Workshop on Neural Networks: From Biology to High-energy Physics*, pp. 0215–224, 9 1994.

Linnainmaa, S. Taylor expansion of the accumulated rounding error. *BIT Numerical Mathematics*, 16:146–160, 1976.

Lucas, P. and van der Gaag, L. Principles of expert systems, 1991.

Macía, J. and Sole, R. How to make a synthetic multicellular computer. *PloS one*, 9:e81248, 02 2014. doi: 10.1371/journal.pone.0081248.

Marblestone, A. H., Wayne, G., and Kording, K. P. Toward an integration of deep learning and neuroscience. *Frontiers in Computational Neuroscience*, 10:94, 2016. ISSN 1662-5188. doi: 10.3389/fncom.2016.00094. URL https://www.frontiersin.org/article/10.3389/fncom.2016.00094.

Marcus, G., Marblestone, A., and Dean, T. The atoms of neural computation. *Science*, 346:551–552, 2014. Computational Neuroscience.

Mcclelland, J., Mcnaughton, B., and O'Reilly, R. Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102:419–57, 08 1995. doi: 10.1037/0033-295X.102.3.419.

McCloskey, M. and Cohen, N. J. Catastrophic interference in connectionist networks: The sequential learning problem, 1989. ISSN 0079-7421.

Mernik, M., Heering, J., and Sloane, A. M. When and how to develop domain-specific languages. *ACM Comput. Surv.*, 37(4):316–344, December 2005. ISSN 0360-0300. doi: 10.1145/1118890.1118892. URL https://doi.org/10.1145/1118890.1118892.

Metz, C. Big bets on a.i. open a new frontier for chip start-ups, too, 2018. URL https://www.nytimes.com/2018/01/14/technology/artificial-intelligence-chip-start-ups.html.

Mirhoseini, A., Goldie, A., Yazgan, M., Jiang, J., Songhori, E., Wang, S., Lee, Y.-J., Johnson, E., Pathak, O., Bae, S., Nazi, A., Pak, J., Tong, A., Srinivasa, K., Hang, W., Tuncer, E., Babu, A., Le, Q. V., Laudon, J., Ho, R., Carpenter, R., and Dean, J. Chip Placement with Deep Reinforcement Learning. *arXiv e-prints*, art. arXiv:2004.10746, April 2020.

Misra, J. and Saha, I. Artificial neural networks in hardware: A survey of two decades of progress. *Neurocomputing*, 74(1):239 – 255, 2010. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2010.03.021. URL http://www.sciencedirect.com/science/article/pii/S092523121000216X. Artificial Brains.

Moore, D. The anna karenina principle applied to ecological risk assessments of multiple stressors. *Human and Ecological Risk Assessment: An International Journal*, 7(2):231–237, 2001. doi: 10.1080/20018091094349.

Lee, H., Brown, K., Sujeeth, A., Chafi, H., Rompf, T., Odersky, M., 和 Olukotun, K. 实现异构并行计算的领域特定语言。*IEEE Micro*, 31(5):42–53, 2011。Lee, K. 和 Wang, X. Facebook AI 硬件基础设施的下一步，2018。网址 https://bit.ly/3bgZFDn。

Leiserson, C. E., Thompson, N. C., Emer, J. S., Kuszmaul, B. C., Lampson, B. W., Sanchez, D., 和 Schardl, T. B. 顶部有足够的空间：在摩尔定律之后，什么将推动计算机性能？ *Science*, 368(6495), 2020. ISSN 0036- 8075. doi: 10.1126/science.aam9744. URL https://science.sciencemag.org/ content/368/6495/eaam9744.

Lillicrap, T. P. 和 Santoro, A. 通过时间的反向传播与大脑。 *Current Opinion in Neurobiology*, 55:82 – 89, 2019. ISSN 0959-4388. doi: https://doi.org/10. 1016/j.conb.2019.01.011. URL http: //www.sciencedirect.com/science/ article/pii/S0959438818302009. 机器学习、大数据与神经科学。

Lin, X., Rivenson, Y., Yardimci, N. T., Veli, M., Luo, Y., Jarrahi, M., 和 Ozcan, A. 使用衍射深度神经网络的全光学机器学习。*Science*, 361 (6406):1004–1008, 2018. ISSN 0036- 8075. doi: 10.1126/science.aat8084. URL https://science.sciencemag.org/ content/361/6406/1004.

林赛，C. S. 和林布拉德，T. 硬件神经网络的评审：用户的视角。在 *3rd Workshop on Neural Networks: From Biology to High-energy Physics*，第 0215–224 页，1994 年 9 月。

Linnainmaa, S. 泰勒展开的累积舍入误差。 *BIT Numerical Mathe- matics*, 16:146–160, 1976.

卢卡斯，P. 和范德加赫，L. 专家系统原理，1991。

Macía, J. 和 Sole, R. 如何制造合成多细胞计算机。*PloS one*, 9:e81248, 2014年02月。doi: 10.1371/journal.pone.0081248.

Marblestone, A. H., Wayne, G., 和 Kording, K. P. 朝着深度学习与神经科学的整合。 *Frontiers in Computa- tional Neuroscience*, 10:94, 2016. ISSN 1662-5188. doi: 10.3389/fncom.2016.00094. URL https://www.frontiersin.org/ article/10.3389/fncom.2016.00094. Marcus, G., Marblestone, A., 和 Dean, T. 神经计算的原子。 *Science*, 346:551–552, 2014. 计算神经科学。 Mcclelland, J., Mcnaughton, B., 和 O’Reilly, R. 为什么海马体和新皮层中存在互补学习系统：来自连接主义学习和记忆模型成功与失败的见解。 *Psychological review*, 102:419–57, 1995年8月。 doi: 10.1037/0033-295X.102.3.419. McCloskey, M. 和 Cohen, N. J. 连接主义网络中的灾难性干扰：顺序学习问题，1989年。 ISSN 0079-7421. Mernik, M., Heering, J., 和 Sloane, A. M. 何时以及如何开发领域特定语言。 *ACM Comput. Surv.*, 37(4):316–344, 2005年12月。 ISSN 0360-0300. doi: 10.1145/1118890.1118892. URL https://doi.org/10.1145/1118890.1118892. Metz, C. 对人工智能的重大投资为芯片初创公司开辟了新前沿，2018年。 URL https://www.nytimes.com/2018/01/14/technology/artificial-intelligence-chip-start-ups.html. Mirhoseini, A., Goldie, A., Yazgan, M., Jiang, J., Songhori, E., Wang, S., Lee, Y.-J., Johnson, E., Pathak, O., Bae, S., Nazi, A., Pak, J., Tong, A., Srinivasa, K., Hang, W., Tuncer, E., Babu, A., Le, Q. V., Laudon, J., Ho, R., Carpenter, R., 和 Dean, J. 使用深度强化学习进行芯片布局。 *arXiv e- prints*, art. arXiv:2004.10746, 2020年4月。 Misra, J. 和 Saha, I. 硬件中的人工神经网络：二十年进展的调查。 *Neurocomputing*, 74(1):239 – 255, 2010年。 ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2010.03.021. URL http://www.sciencedirect.com/science/article/pii/S092523121000216X. 人工大脑。 Moore, D. 安娜·卡列尼娜原则在多重压力源的生态风险评估中的应用。 *Human and Ecological Risk Assessment: An International Journal*, 7(2):231–237, 2001年。 doi: 10.1080/20018091094349.

Moore, G. Cramming more components onto integrated circuits. *Electronics*, 38(8), April 1965. URL https://www.cs.utexas.edu/~fussell/courses/cs352h/papers/moore.pdf.

Moravec, H. When will computer hardware match the human brain. *Journal of Transhumanism*, 1, 1998.

Morgan, M. G. The fifth generation: Artificial intelligence and japan's computer challenge to the world, by edward a. feigenbaum and pamela mccorduck. reading, ma: Addison-wesley, 1983, 275 pp. price: $15.35. *Journal of Policy Analysis and Management*, 3(1):156–156, 1983. doi: 10.2307/3324061. URL https://onlinelibrary.wiley.com/doi/abs/10.2307/3324061.

Morgan, N., Beck, J., Kohn, P., Bilmes, J., Allman, E., and Beer, J. The ring array processor: A multiprocessing peripheral for connectionist applications. *Journal of Parallel and Distributed Computing*, 14(3):248 – 259, 1992. ISSN 0743-7315. doi: https://doi.org/10.1016/0743-7315(92)90067-W. URL http://www.sciencedirect.com/science/article/pii/074373159290067W.

Nikonov, D. E. and Young, I. A. Overview of beyond-cmos devices and a uniform methodology for their benchmarking. *Proceedings of the IEEE*, 101(12):2498–2533, 2013.

Oh, K.-S. and Jung, K. Gpu implementation of neural networks. *Pattern Recognition*, 37(6):1311 – 1314, 2004. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2004.01.013. URL http://www.sciencedirect.com/science/article/pii/S0031320304000524.

Olukotun, K. Beyond parallel programming with domain specific languages. *SIGPLAN Not.*, 49(8):179–180, February 2014. ISSN 0362-1340. doi: 10.1145/2692916.2557966. URL https://doi.org/10.1145/2692916.2557966.

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. Continual Lifelong Learning with Neural Networks: A Review. *arXiv e-prints*, art. arXiv:1802.07569, February 2018.

Payne, B. R., Belkasim, S. O., Owen, G. S., Weeks, M. C., and Zhu, Y. Accelerated 2d image processing on gpus. In Sunderam, V. S., van Albada, G. D., Sloot, P. M. A., and Dongarra, J. J. (eds.), *Computational Science – ICCS 2005*, pp. 256–264, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISBN 978-3-540-32114-9.

Posselt, E. *The Jacquard Machine Analyzed and Explained: The Preparation of Jacquard Cards and Practical Hints to Learners of Jacquard Designing*. Posselt's textile library. E.A. Posselt, 1888. URL https://books.google.com/books?id=-6FtmgEACAAJ.

Prabhakar, R., Zhang, Y., Koeplinger, D., Feldman, M., Zhao, T., Hadjis, S., Pedram, A., Kozyrakis, C., and Olukotun, K. Plasticine: A reconfigurable architecture for parallel patterns. In *2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)*, pp. 389–402, 2017.

Project, C. H. A. Computer history 1949 - 1960 early vacuum tube computers overview, 2018. URL https://www.youtube.com/watch?v=WnNm_uJYWhA.

Reddi, V. J., Cheng, C., Kanter, D., Mattson, P., Schmuelling, G., Wu, C., Anderson, B., Breughe, M., Charlebois, M., Chou, W., Chukka, R., Coleman, C., Davis, S., Deng, P., Diamos, G., Duke, J., Fick, D., Gardner, J. S., Hubara, I., Idgunji, S., Jablin, T. B., Jiao, J., John, T. S., Kanwar, P., Lee, D., Liao, J., Lokhmotov, A., Massa, F., Meng, P., Micikevicius, P., Osborne, C., Pekhimenko, G., Rajan, A. T. R., Sequeira, D., Sirasao, A., Sun, F., Tang, H., Thomson, M., Wei, F., Wu, E., Xu, L., Yamada, K., Yu, B., Yuan, G., Zhong, A., Zhang, P., and Zhou, Y. Mlperf inference benchmark. In *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*, pp. 446–459, 2020.

Rumelhart, D. E., McClelland, J. L., and PDP Research Group, C. (eds.). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. MIT Press, Cambridge, MA, USA, 1986. ISBN 026268053X.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. *Learning Representations by Back-Propagating Errors*, pp. 696–699. MIT Press, 1988.

Sabour, S., Frosst, N., and Hinton, G. E. Dynamic routing between capsules, 2017. URL http://papers.nips.cc/paper/6975-dynamic-routing-between-capsules.pdf.

摩尔，G. 将更多组件压缩到集成电路上。*Electron- ics*，38(8)，1965年4月。网址 https://www.cs.utexas.edu/~fussell/courses/cs352h/papers/moore.pdf。莫拉维克，H. 计算机硬件何时能与人脑匹敌。*Journal of Transhu- manism*，1，1998年。摩根，M. G. 第五代：人工智能和日本对世界的计算机挑战，作者：爱德华·A·费根鲍姆和帕梅拉·麦考尔杜克。马萨诸塞州雷丁：阿迪森-韦斯利，1983年，275页。价格：$15.35。*Journal of Policy Analysis and Management*，3(1)：156–156，1983年。doi：10.2307/3324061。网址 https://onlinelibrary.wiley.com/doi/abs/10.2307/3324061。摩根，N.，贝克，J.，科恩，P.，比尔梅斯，J.，阿尔曼，E.，和比尔，J. 环阵列处理器：用于连接主义应用的多处理外设。*Journal of Parallel and Distributed Computing*，14(3)：248 – 259，1992年。ISSN 0743-7315。doi：https://doi.org/10.1016/0743-7315(92)90067-W。网址 http://www.sciencedirect.com/science/article/pii/074373159290067W。尼科诺夫，D. E. 和杨，I. A. 超越CMOS设备的概述及其基准测试的统一方法。*Proceedings of the IEEE*，101(12)：2498–2533，2013年。

哦，K.-S. 和 Jung, K. 神经网络的 GPU 实现。*Pattern Recogni- tion*, 37(6):1311 – 1314, 2004. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2004.01.013. URL http://www.sciencedirect.com/science/article/pii/S0031320304000524.

Olukotun, K. 超越使用领域特定语言的并行编程。*SIGPLAN Not.*, 49(8):179–180, 2014年2月。ISSN 0362-1340。doi: 10.1145/2692916.2557966。URL https://doi.org/10.1145/2692916.2557966。

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., 和 Wermter, S. 神经网络的持续终身学习：综述。*arXiv e-prints*, 文章. arXiv:1802.07569, 2018年2月。

佩恩, B. R., 贝尔卡西姆, S. O., 欧文, G. S., 威克斯, M. C., 和朱, Y. 在GPU上加速2D图像处理. 在桑德拉姆, V. S., van Albada, G. D., Sloot, P. M. A., 和 Dongarra, J. J. (编辑), *Computational Science – ICCS 2005*, 第 256–264 页, 柏林, 海德堡, 2005. 施普林格 柏林 海德堡. ISBN 978-3-540-32114-9.

波塞尔，E. *The Jacquard Machine Analyzed and Explained: The Preparation of Jacquard Cards and Practical Hints to Learners of Jacquard Designing*。波塞尔的纺织图书馆。E.A. 波塞尔，1888年。网址 https://books.google.com/books?id=-6FtmgEACAAJ。

Prabhakar, R., Zhang, Y., Koeplinger, D., Feldman, M., Zhao, T., Hadjis, S., Pedram, A., Kozyrakis, C., 和 Olukotun, K. Plasticine: 一种用于并行模式的可重构架构。在 *2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)*，第 389–402 页，2017 年。

项目，C. H. A. 计算机历史 1949 - 1960 早期真空管计算机概述，2018。网址 https://www.youtube.com/watch?v=WnNm_uJYWhA。

Reddi, V. J., Cheng, C., Kanter, D., Matt- son, P., Schmuelling, G., Wu, C., Anderson, B., Breughe, M., Charlebois, M., Chou, W., Chukka, R., Coleman, C., Davis, S., Deng, P., Diamos, G., Duke, J., Fick, D., Gardner, J. S., Hubara, I., Idgunji, S., Jablin, T. B., Jiao, J., John, T. S., Kanwar, P., Lee, D., Liao, J., Lokhmotov, A., Massa, F., Meng, P., Micike- vicius, P., Osborne, C., Pekhimenko, G., Ra- jan, A. T. R., Sequeira, D., Sirasao, A., Sun, F., Tang, H., Thomson, M., Wei, F., Wu, E., Xu, L., Yamada, K., Yu, B., Yuan, G., Zhong, A., Zhang, P., 和 Zhou, Y. Mlperf 推理基准。在 *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*，第 446–459 页，2020。

Rumelhart, D. E., McClelland, J. L. 和 PDP 研究小组, C. (编). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. 麻省理工学院出版社, 剑桥, 马萨诸塞州, 美国, 1986. ISBN 026268053X.

Rumelhart, D. E., Hinton, G. E., 和 Williams, R. J. *Learning Representations by Back-Propagating Errors*, 第696–699页。麻省理工学院出版社，1988年。

Sabour, S., Frosst, N., 和 Hinton, G. E. 胶囊之间的动态路由, 2017. URL http://papers.nips.cc/paper/6975-dynamic-routing-between-capsules.pdf.

Sackinger, E., Boser, B. E., Bromley, J., LeCun, Y., and Jackel, L. D. Application of the anna neural network chip to high-speed character recognition. *IEEE Transactions on Neural Networks*, 3(3):498–505, 1992.

Sainani, K. On the frontiers of biomedicine with professor rahul sarpeshkar. *Dartmouth Magazine*, 2017. URL https://dartmouthalumnimagazine.com/articles/cell-power.

Sato, K. What makes tpus fine-tuned for deep learning?, 2018. URL https://bit.ly/2ER3bIu.

Shalf, J. The future of computing beyond moore's law. *Philosophical Transactions of the Royal Society A*, 378, 2020.

Singh, D.-V., Perdigones, A., Garcia, J., Cañas, I., and Mazarrón, F. Analyzing worldwide research in hardware architecture, 1997-2011. *Communications of the ACM*, Volume 58:Pages 76–85, 01 2015. doi: 10.1145/2688498.2688499.

Spelke, E. S. and Kinzler, K. D. Core knowledge. *Developmental Science*, 10(1):89–96, 2007. doi: 10.1111/j.1467-7687.2007.00569.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-7687.2007.00569.x.

Stein, G., Calvert, G., Spence, C., Spence, D., Stein, B., and Stein, P. *The Handbook of Multisensory Processes*. A Bradford book. MIT Press, 2004. ISBN 9780262033213. URL https://books.google.com/books?id=CZS_yDoFV7AC.

Stroop, J. R. Studies of interference in serial verbal reactions. *Journal of Experimental Psychology*, 18(6):643, 1935. doi: 10.1037/h0054651.

Strubell, E., Ganesh, A., and McCallum, A. Energy and policy considerations for deep learning in nlp, 2019.

Sun, F., Qin, M., Zhang, T., Liu, L., Chen, Y.-K., and Xie, Y. Computation on Sparse Neural Networks: an Inspiration for Future Hardware. *arXiv e-prints*, art. arXiv:2004.11946, April 2020.

Sutton, R. The bitter lesson, 2019. URL http://www.incompleteideas.net/IncIdeas/BitterLesson.html.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the Inception Architecture for Computer Vision. *arXiv e-prints*, art. arXiv:1512.00567, December 2015a.

Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015b.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *arXiv e-prints*, art. arXiv:1602.07261, February 2016.

Tan, C., Song, H., Niemi, J., and You, L. A synthetic biology challenge: Making cells compute. *Molecular bioSystems*, 3:343–53, 06 2007. doi: 10.1039/b618473c.

Tani, J. and Press, O. U. *Exploring Robotic Minds: Actions, Symbols, and Consciousness as Self-organizing Dynamic Phenomena*. Oxford series on cognitive models and architectures. Oxford University Press, 2016. ISBN 9780190281083. URL https://books.google.com/books?id=QswnnQAACAAJ.

Taubes, G. The rise and fall of thinking machines, 1995. URL https://www.inc.com/magazine/19950915/2622.html.

Thompson, N. and Spanuth, S. The decline of computers as a general purpose technology: Why deep learning and the end of moore's law are fragmenting computing, 2018.

Thompson, N. C., Greenewald, K., Lee, K., and Manso, G. F. The Computational Limits of Deep Learning. *arXiv e-prints*, art. arXiv:2007.05558, July 2020a.

Thompson, N. C., Greenewald, K., Lee, K., and Manso, G. F. The Computational Limits of Deep Learning. *arXiv e-prints*, art. arXiv:2007.05558, July 2020b.

Time. *Understanding computers: software*. Time, Virginia, 1985.

Tolstoy, L. and Bartlett, R. *Anna Karenina*. Oxford world's classics. Oxford University Press, 2016. ISBN 9780198748847. URL https://books.google.com/books?id=1DooDwAAQBAJ.

Sackinger, E., Boser, B. E., Bromley, J., LeCun, Y., 和 Jackel, L. D. 将安娜神经网络芯片应用于高速字符识别。

IEEE Transactions on Neural Networks, 3(3):498–505, 1992. Sainani, K. 与拉胡尔·萨尔佩什卡教授一起探索生物医学的前沿。

Dartmouth Magazine, 2017. URL https://dartmouthalumnimagazine.com/articles/cell-power. Sato, K. 什么使TPU适合深度学习？, 2018. URL https://bit.ly/2ER3bIu. Shalf, J. 超越摩尔定律的计算未来。

Philosophical Transactions of the Royal Society A, 378, 2020. Singh, D.-V., Perdigones, A., Garcia, J., Cañas, I., 和 Mazarrón, F. 分析全球硬件架构研究，1997-2011。

Communications of the ACM, 第58卷：第76–85页, 2015年01月. doi: 10.1145/2688498.2688499. Spelke, E. S. 和 Kinzler, K. D. 核心知识。

Developmental Science, 10(1):89–96, 2007. doi: 10.1111/j.1467-7687.2007.00569.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-7687.2007.00569.x. Stein, G., Calvert, G., Spence, C., Spence, D., Stein, B., 和 Stein, P.

The Hand- book of Multisensory Processes. 一本布拉德福德书。 MIT出版社, 2004. ISBN 9780262033213. URL https://books.google.com/books?id=CZS_yDoFV7AC. Stroop, J. R. 关于串行语言反应中的干扰研究。

Journal of Experimental Psychology, 18(6):643, 1935. doi: 10.1037/h0054651.

Strubell, E., Ganesh, A., 和 McCallum, A. 深度学习在自然语言处理中的能源和政策考虑, 2019。

Sun, F., Qin, M., Zhang, T., Liu, L., Chen, Y.-K., 和 Xie, Y. 稀疏神经网络的计算：对未来硬件的启示。*arXiv e-prints*, 文章 arXiv:2004.11946, 2020年4月。

萨顿，R. 苦涩的教训，2019。网址 http://www.incompleteideas.net/ IncIdeas/BitterLesson.html。

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., 和 Wojna, Z. 重新思考计算机视觉的 Inception 架构。*arXiv e-prints*，艺术。arXiv:1512.00567，2015 年 12 月。Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., 和 Rabinovich, A. 深入卷积。 在 *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*，第 1–9 页，2015 年。Szegedy, C., Ioffe, S., Vanhoucke, V., 和 Alemi, A. Inception-v4, Inception-ResNet 及残差连接对学习的影响。*arXiv e-prints*，艺术。arXiv:1602.07261，2016 年 2 月。Tan, C., Song, H., Niemi, J., 和 You, L. 一个合成生物学挑战：让细胞计算。*Molecular bioSystems*，3:343–53，2007 年 6 月。doi: 10.1039/b618473c。Tani, J. 和 Press, O. U. *Exploring Robotic Minds: Actions, Symbols, and Consciousness as Self-organizing Dynamic Phenomena*。牛津认知模型与架构系列。牛津大学出版社，2016 年。ISBN 9780190281083。URL https://books.google.com/books?id=QswnnQAACAAJ。Taubes, G. 思维机器的兴衰，1995 年。URL https://www.inc.com/magazine/19950915/2622.html。Thompson, N. 和 Spanuth, S. 计算机作为通用技术的衰退：深度学习和摩尔定律的终结如何分裂计算，2018 年。Thompson, N. C., Greenewald, K., Lee, K., 和 Manso, G. F. 深度学习的计算极限。*arXiv e-prints*，艺术。arXiv:2007.05558，2020 年 7 月。Thompson, N. C., Greenewald, K., Lee, K., 和 Manso, G. F. 深度学习的计算极限。*arXiv e-prints*，艺术。arXiv:2007.05558，2020 年 7 月。时间。*Understanding computers: software*。时间，弗吉尼亚，1985 年。托尔斯泰, L. 和 巴特利特, R. *Anna Karenina*。牛津世界经典。牛津大学出版社，2016 年。ISBN 9780198748847。URL https://books.google.com/books?id=1DooDwAAQBAJ。

Touretzky, D. and Waibel, A. Course: 15-880(a) – introduction to neural networks, 1995. URL shorturl.at/evKX9.

Tulving, E. Episodic memory: From mind to brain. *Annual Review of Psychology*, 53(1):1–25, 2002. doi: 10.1146/annurev.psych.53.100901.135114. URL https://doi.org/10.1146/annurev.psych.53.100901.135114. PMID: 11752477.

Van Der Malsburg, C. Frank rosenblatt: Principles of neurodynamics: Perceptrons and the theory of brain mechanisms, 1986.

Von Neumann, J., Churchland, P., and Churchland, P. *The Computer and the Brain*. The Silliman Memorial Lectures Series. Yale University Press, 2000. ISBN 9780300084733. URL https://books.google.com/books?id=Q30MqJjRv1gC.

Wang, K., Liu, Z., Lin, Y., Lin, J., and Han, S. Haq: Hardware-aware automated quantization. *ArXiv*, abs/1811.08886, 2018.

Warden, P. and Situnayake, D. *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*. O'Reilly Media, Incorporated, 2019. ISBN 9781492052043. URL https://books.google.com/books?id=sB3mxQEACAAJ.

Welling, M. Do we still need models or just more data and compute?, 2019. URL shorturl.at/qABIY.

Wiggers, K. Openai launches an api to commercialize its research, 2020. URL https://bit.ly/31NAJQB.

Xu, H., Mitchell, N., Arnold, M., Rountev, A., and Sevitsky, G. Software bloat analysis: Finding, removing, and preventing performance problems in modern large-scale object-oriented applications, 01 2010.

Zador, A. M. A critique of pure learning: What artificial neural networks can learn from animal brains. *bioRxiv*, 2019.

Zhen, D., Yao, Z., Gholami, A., Mahoney, M., and Keutzer, K. Hawq: Hessian aware quantization of neural networks with mixed-precision, 10 2019.

Touretzky, D. 和 Waibel, A. 课程：15-880(a) − 神经网络导论，1995年。网址 shorturl.at/evKX9。Tulving, E. 情景记忆：从大脑到心智。*Annual Review of Psychology*, 53(1):1−25, 2002年。doi: 10.1146/annurev.psych.53.100901.135114。网址 https://doi.org/10.1146/annurev.psych.53.100901.135114。PMID: 11752477。

范·德·马尔斯堡，C. 弗兰克·罗森布拉特：神经动力学原理：感知器与大脑机制理论，1986。

冯·诺依曼, J., 彻奇兰, P., 和 彻奇兰, P. *The Computer and the Brain*。西利曼纪念讲座系列。耶鲁大学出版社, 2000年。ISBN 9780300084733。网址 https://books.google.com/books?id=Q30MqJjRv1gC。

王, K., 刘, Z., 林, Y., 林, J., 和 韩, S. Haq: 硬件感知自动量化. *ArXiv*, abs/1811.08886, 2018.

Warden, P. 和 Situnayake, D. *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*。O’Reilly Media, Incorporated, 2019。ISBN 9781492052043。网址 https://books.google.com/books?id=sB3mxQEACAAJ。

Welling, M. 我们还需要模型，还是只需要更多的数据和计算能力？, 2019. URL shorturl.at/qABIY.

Wiggers, K. Openai推出了一种API以商业化其研究，2020年。网址 https://bit.ly/31NAJQB。

徐, H., 米切尔, N., 阿诺德, M., 罗恩特夫, A., 和 塞维茨基, G. 软件膨胀分析：发现、消除和防止现代大规模面向对象应用中的性能问题, 2010年1月。

Zador, A. M. 纯学习的批判：人工神经网络可以从动物大脑中学到什么。*bioRxiv*，2019。

Zhen, D., Yao, Z., Gholami, A., Mahoney, M., 和 Keutzer, K. Hawq: 具有混合精度的神经网络的海森矩阵感知量化, 2019年10月。