

SUMMARY

There are an introduction and key points at the beginning of the article.

First computers were very specialized. Software was written exclusively for certain model of computer. The general-purpose computer era crystallized in 1969, when Gordon Moore wrote his mind in one magazine – he said that the number of transistors on an integrated circuit could be doubled every two years. Prediction held and motivated a remarkable decline in the cost of transforming energy into information over the next 50 years. 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 computing power.

There is one essential principle: “a deficiency in any one of a number of factors dooms an endeavor to failure” (Anna Karenina principle). Most computer science breakthroughs follow the Anna Karenina principle. hardware lottery—avoiding possible points of failure in downstream hardware and software choices.

Charles Babbage was a computer pioneer who designed a machine that could be programmed, but first such machines were build much later – in 20th century.

GLOSSARY

- software – программное обеспечение
- hardware- аппаратное обеспечение
- siloed – разрозненный
- heterogeneous – неоднородный
- thread – поток
- compatible – совместимый
- general-purpose – общего назначения
- integrated circuit – интегральная схема
- emphasis – акцент
- myriad – мириады
- deviate – отклониться
- formidable – значительный, внушительный
- cease – прекратить
- deficiency – недостаток
- endeavor – попытка
- to doom – обречь
- surreptitiously – тайком
- fabricating – изготовление
- precision – точность
- to lay down – закладывать (основы)

TRANSLATION

История говорит нам, что научный прогресс несовершенен. Интеллектуальные традиции и доступный инструментарий могут оттолкнуть ученых от одних идей и склонить их к другим. Это добавляет шума на рынок идей и часто означает инерцию в распознавании перспективных направлений исследований. В этой статье утверждается, что в области исследований искусственного интеллекта (ИИ) именно инструменты сыграли непропорционально большую роль в принятии решения о том, какие идеи успешны, а какие нет.

Далее следует документ с изложением позиции и исторический обзор. Я ввожу термин «аппаратная лотерея» для описания случая, когда исследовательская идея выигрывает, потому что она совместима с имеющимся программным и аппаратным обеспечением, а не потому, что эта идея превосходит альтернативные направления исследований. Выбор программного и аппаратного обеспечения часто играл решающую роль в определении победителей и проигравших в ранней истории информатики.

Эти уроки особенно важны по мере того, как мы вступаем в новую эру более тесного сотрудничества между исследовательскими сообществами, занимающимися аппаратным и программным обеспечением, и машинным обучением. После десятилетий рассмотрения аппаратного обеспечения, программного обеспечения и алгоритма как отдельных вариантов, катализаторы более тесного сотрудничества включают изменение экономики аппаратного обеспечения, гонку «чем больше, тем лучше» в размерах архитектур глубокого обучения и головокружительные требования развертывания машинного обучения к пограничным устройствам.

Более тесное сотрудничество сосредоточено на волне аппаратного обеспечения нового поколения, ориентированного на предметную область, которое оптимизируется для коммерческих вариантов использования глубоких нейронных сетей. В то время как специализация предметной области обеспечивает значительное повышение эффективности для основных исследований, ориентированных на глубокие нейронные сети, она, возможно, делает еще более дорогостоящим отклонение от проторенного пути исследовательских идей. Все более фрагментированный аппаратный ландшафт означает, что выгоды от прогресса в области вычислений будут все более неравномерными. Хотя у глубоких нейронных сетей есть явные варианты коммерческого использования, есть ранние признаки того, что путь к следующему прорыву в области ИИ может потребовать совершенно другой комбинации алгоритма, аппаратного и программного обеспечения.

Эта статья начинается с признания важного парадокса: исследователи машинного обучения в основном игнорируют аппаратные средства, несмотря на ту роль, которую они играют в определении успеха идей. Разрозненная эволюция аппаратного, программного и алгоритмического обеспечения сыграла решающую роль в ранних аппаратных и программных лотереях. В этой статье рассматриваются последствия этой разрозненной эволюции на примерах первых аппаратных и программных лотерей. И хотя сегодняшний аппаратный ландшафт становится все более неоднородным, я утверждаю, что аппаратная лотерея никуда не делась, и разрыв между победителями и проигравшими будет увеличиваться. После изложения этих аргументов статья завершается мыслями о том, как избежать будущих аппаратных лотерей.

Отдельные племена

Для создателей первых компьютеров программой была машина. Из-за стоимости электроники и отсутствия универсального программного обеспечения ранние машины могли выполнять лишь одну функцию; не предполагалось, что они будут перепрофилированы для выполнения новой задачи. «Разностная машина» Чарльза Бэббиджа (1817 г.) предназначалась исключительно для

вычисления полиномиальных функций. Harvard Mark I IBM (1944 г.) был программируемым калькулятором. Перцептронная машина Розенблатта (1958 г.) вычисляла ступенчатую однослойную сеть. Даже ткацкий станок Жаккарда (1804 г.), которую часто считают одной из первых программируемых машин, на практике повторное выполнение потоков было настолько дорогим, что обычно оно выполнялось один раз для поддержки предварительно фиксированного набора полей ввода.

В начале 1960-х совместная специализация аппаратного и программного обеспечения двигалась вертикально. IBM была пионером в создании наборов инструкций, переносимых между компьютерами. Растущий бизнес может установить небольшой компьютер IBM 360 и не быть вынужденным заново изучать все при переходе на более крупную машину 360. Конкуренты Burroughs, Cray и Honeywell разработали свои собственные системы, совместимые со своими машинами, но не с другими производителями. Программы можно было переносить между разными машинами одного и того же производителя, но не на конкурирующие машины. Сама конструкция оставалась разрозненной, а аппаратное и программное обеспечение разрабатывались совместно внутри компании.

Сегодня, в отличие от специализации, необходимой в самом начале развития вычислительной техники, исследователи машинного обучения склонны рассматривать аппаратное обеспечение, программное обеспечение и алгоритмы как три отдельных варианта. Во многом это связано с периодом в истории компьютерных наук, который радикально изменил тип производимого оборудования и побудил сообщества исследователей аппаратного, программного обеспечения и машинного обучения развиваться изолированно.

Эпоха компьютеров общего назначения началась в 1969 году, когда молодой инженер по имени Гордан Мур написал в журнале Electronics статью под названием «Втиснуть больше компонентов на печатные платы». В ней Мур предсказал, что количество транзисторов в интегральной схеме может удваиваться каждые два года. Статья и последующие действия изначально были мотивированы простым желанием — Мур думал, что это продаст больше фишек. Однако предсказание подтвердилось и привело к значительному снижению стоимости преобразования энергии в информацию в течение следующих 50 лет.

Закон Мура в сочетании с масштабированием Деннарда позволил увеличить производительность микропроцессора в три раза с 1980 по 2010 год. Предсказуемый рост вычислительной мощности и объема памяти каждые два года означал, что проектирование оборудования стало менее рискованным. Зачем экспериментировать с более специализированными конструкциями аппаратного обеспечения за неопределенное вознаграждение, когда закон Мура позволяет производителям чипов фиксировать предсказуемую норму прибыли? Даже для задач, требующих более высокой производительности, преимущества перехода на специализированное оборудование могут быть быстро вытеснены аппаратным обеспечением общего назначения следующего поколения с постоянно растущей вычислительной мощностью.

Акцент сместился на универсальные процессоры, которые могли решать множество разных задач. Несколько попыток отклониться от курса и создать специализированные суперкомпьютеры для исследований были финансово неустойчивыми и недолговечными. Несколько очень узких задач, таких как овладение шахматами, были исключением из этого правила, потому что престиж и видимость победы над противником-человеком привлекали корпоративное спонсорство.

До недавнего времени сохранялось отношение к выбору оборудования, программного обеспечения и алгоритма как к независимым. Изучение новых типов аппаратного обеспечения дорого обходится как с точки зрения времени, так и требуемых капиталовложений. Производство чипа следующего поколения обычно стоит от 30 до 80 миллионов долларов, а на разработку уходит от двух до трех лет. Эти колоссальные барьеры для входа создали культуру исследования

аппаратного обеспечения, которая может показаться странной или, возможно, даже медленной для среднего исследователя машинного обучения. В то время как количество публикаций по машинному обучению за последние 30 лет росло в геометрической прогрессии, количество публикаций по оборудованию оставалось довольно стабильным. Для компании, производящей оборудование, утечка интеллектуальной собственности может решить или разрушить ее выживание. Это привело к гораздо более хорошо охраняемой исследовательской культуре.

В отсутствие какого-либо рычага, с помощью которого можно было бы повлиять на разработку аппаратного обеспечения, исследователи машинного обучения стали рационально относиться к аппаратному обеспечению как к невозвратным издержкам, которые нужно обойти, а не к чему-то изменчивому, что можно изменить. Однако то, что мы абстрагировали аппаратное обеспечение, не означает, что оно перестало существовать. Ранняя история информатики говорит нам, что было много аппаратных лотерей, в которых выбор аппаратного и программного обеспечения определял, какие идеи преуспеют, а какие потерпят неудачу.

Аппаратная лотерея

Первое предложение толстовской «Анны Карениной» гласит: «Все счастливые семьи похожи друг на друга, каждая несчастная семья несчастлива по-своему». Толстой говорит, что много различных вещей необходимы для семьи чтобы быть счастливой - финансовая стабильность, здоровое потомство. Однако достаточно отсутствия одного из этих аспектов, чтобы семья была несчастлива. Это стало известно как принцип Анны Карениной: «Недостаток любого из множества факторов обрекает начинание на провал».

Несмотря на то, что мы предпочитаем верить, что алгоритмы успешны или терпят неудачу по отдельности, история говорит нам, что большинство прорывов в области информатики следуют принципу Анны Карениной. Успешные прорывы часто отличаются от неудач тем, что они выигрывают от тайного совпадения нескольких критериев. Для исследований ИИ это часто зависит от победы в том, что мы назвали аппаратной лотереей – избежание возможных точек отказа при выборе аппаратного и программного обеспечения.

Ранним примером аппаратной лотереи является аналитическая машина (1837 г.). Чарльз Бэббидж был пионером в области компьютерных технологий, который разработал машину, которую можно было запрограммировать, по крайней мере теоретически, для решения любых типов вычислений. Его аналитическая машина так и не была построена, отчасти потому, что ему было трудно изготавливать детали с нужной точностью. Электромагнитные технологии, необходимые для фактического создания теоретических основ, заложенных Бэббиджем, появились только во время Второй мировой войны. В первой половине 20-го века электронные вакуумные лампы широко использовались для радиосвязи и радаров. Во время Второй мировой войны эти электронные лампы были перепрофилированы, чтобы обеспечить вычислительную мощность, необходимую для взлома немецкого кода энигмы.

Как отмечалось в телешоу «Кремниевая долина», часто «быть слишком рано — это то же самое, что ошибаться». Когда Бэббидж скончался в 1871 году, между его идеями и современными компьютерами не было непрерывной связи. Концепция хранимой программы, изменяемого кода, памяти и условного ветвления была заново открыта столетие спустя, потому что существовали нужные инструменты, чтобы эмпирически показать, что эта идея работает.