TURINTECH

WHITE PAPER

Code Optimisation: Powering AI Software of Tomorrow

2022

Contents

Executive Summary

> If you don't have an AI strategy, you are going to die in the world that's coming."
>
> - Devin Wenig, former CEO of eBay[1]

We have seen leaps in AI in recent years. From automation to humanoid robots, AI is progressing rapidly while offering massive boosts to productivity and resource efficiency. Recent estimates show that AI adoption will grow 10X by 2030[2]. Companies are increasingly exploring and executing novel techniques to make AI, machine learning, and data science pipelines more robust. In this backdrop, businesses that successfully adopt AI solutions will gain a competitive advantage in the rapidly evolving industry space.

Developing an AI solution requires a few critical elements: (1) data, (2) infrastructure, (3) code, and (4) talent. Increasing the quality of each of these input elements leads to AI systems that are more robust and impactful. Of these elements, code and quality of code can often go unnoticed in commercial production environments.

Code optimisation in essence refers to the process of improving the quality of code bases. With code optimisation, developers can identify sub-par code and make adjustments to ensure that code produces expected outcomes with minimal errors and efficiency loss. However, driven by time pressures, developers may choose to prioritise delivery over code quality and optimisation, leading to an overall efficiency loss in the AI pipeline.

While improving AI pipelines from a technical standpoint, optimised code bases also offer a variety of business uses.

Optimised code bases are faster, improving the performance of AI solutions and AI-enabled devices. This enables businesses to provide greater user experience to customers, while also making critical decisions at greater speed.

Code optimisations leads to far less memory consumption in running code bases, reducing the overall computing cost of managing AI solutions. With code optimisation, businesses can afford AI with ease.

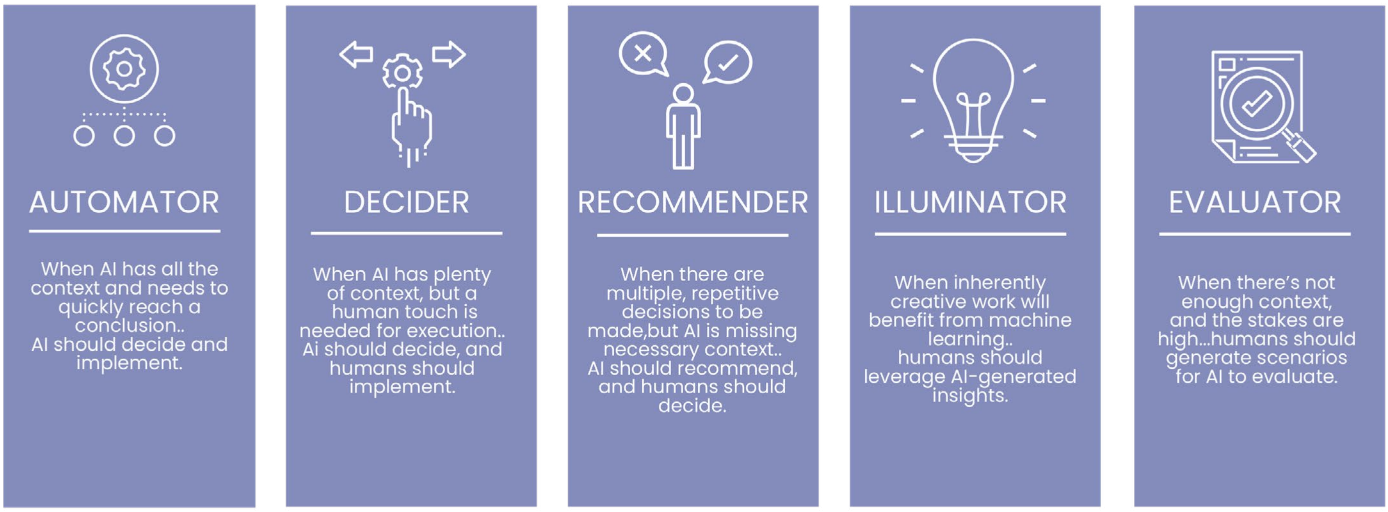
Optimal code bases also consume far less energy, reducing the environmental impact of AI solutions. Code optimisation allows businesses to meet their sustainability goals while executing cutting-edge AI solutions.

evoML, a code optimisation platform by TurinTech, empowers data scientists, engineers and business leaders to deliver efficient ML and other data-heavy applications in weeks not months. evoML brings the entire data science pipeline onto a single platform and is the only platform that embeds optimisation functionalities, improving code efficiency for faster running speed and higher profitability.

TurinTech aims to make code optimisation more accessible to diverse data science and production teams across industries, allowing them to integrate efficient AI into everyday wok practices with ease.

AI is Everywhere, so is Inefficient Code

The contribution of AI to the future of the workplace is rather promising. AI plays several roles in today's businesses and these roles will continue to adapt and grow in the years to come. A BCG article identifies five roles that are facilitated by human-AI interaction[3].



AUTOMATOR

When AI has all the context and needs to quickly reach a conclusion.. AI should decide and implement.

DECIDER

When AI has plenty of context, but a human touch is needed for execution.. Ai should decide, and humans should implement.

RECOMMENDER

When there are multiple, repetitive decisions to be made,but AI is missing necessary context.. AI should recommend, and humans should decide.

ILLUMINATOR

When inherently creative work will benefit from machine learning.. humans should leverage AI-generated insights.

EVALUATOR

When there's not enough context, and the stakes are high...humans should generate scenarios for AI to evaluate.

Figure 1: Nature of Human-AI relationships in organisational decision-making. Recreated from BCG

As we see in Figure 1, AI is not a tool that simply replaces humans, but rather is a resource that complements human capabilities, allowing for better, faster, and rigorous decision-making and execution of work processes.

Recent estimates show that AI adoption will grow 10X by 2030[4], while 95% of new digital workloads will be deployed on cloud-native platforms[5]. Edge computing market size is estimated to hit $ 116.5 billion by 2030[6]. In this backdrop, businesses that successfully adopt AI solutions will gain competitive advantage in the rapidly evolving industry space.

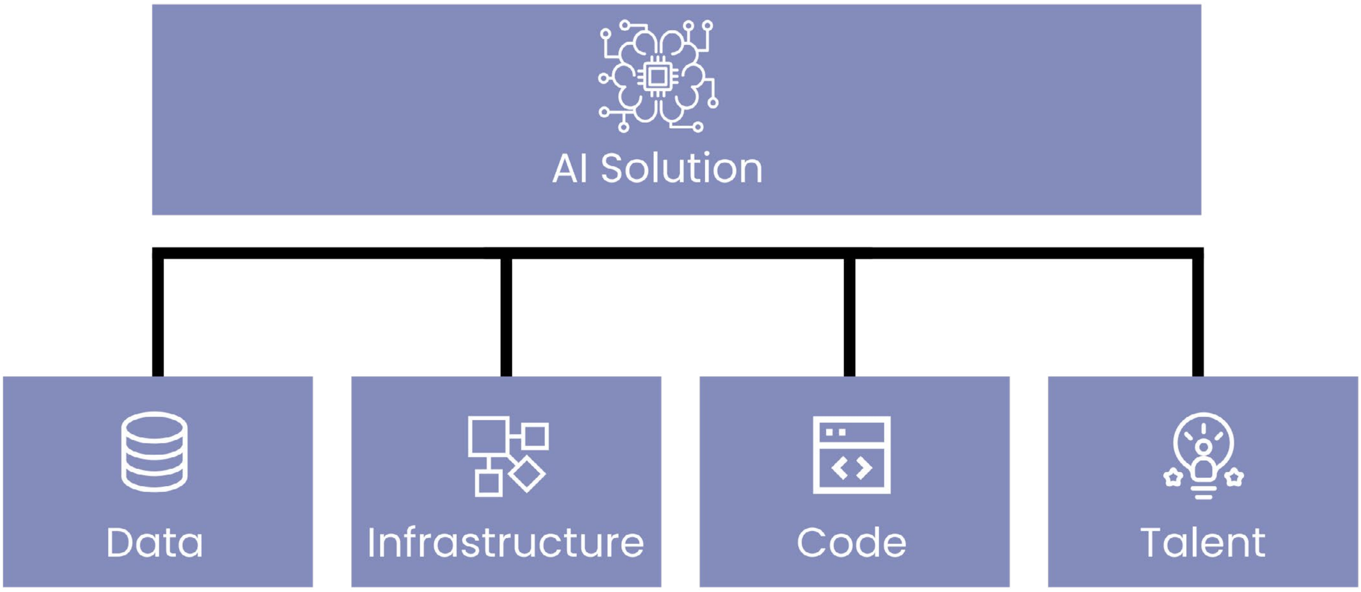
An inevitable part of AI adoption is building and maintaining high-quality code bases.

While the case for high-quality code bases stands strong, in commercial development environments, developers may choose to compromise code quality to achieve other priorities. Developing AI solutions for the commercial market comes with intense time pressures, where data scientists nor software engineers have large amounts of time to spend refining small elements of code bases. As a result, they may be compelled to prioritise implementation while holding back on meeting standards for high-quality code.

Instead of stopping to fix bad code, engineers continue building on top of it as it is painstaking to review and amend old code, creating an even bigger mess in the process. These practices lead to technical debt, and inefficient and draining development outcomes. Hence, in order to achieve optimal performance, an AI system must in essence maximise results while minimising inefficiencies and costs.

A high-quality AI solution requires a few critical building blocks. Good data, good infrastructure, healthy code, and the right talent are essential pillars of success in an AI system. We discuss some of these below.

Fundamental Building Blocks of an AI System



DATA: GOOD DATASETS ARE ESSENTIAL IN TRAINING, TESTING, AND DEPLOYING AI SYSTEMS

AI systems and machine learning models are built, trained, tested, and deployed based on data. Incomplete, invalid, or corrupt data compromises the quality of AI solutions. Training and test data used in an AI system must also be a representative sample of the population. Non-representative datasets pose the risk of introducing biases to AI systems.

Selecting a good dataset and data preprocessing are essential first steps in ensuring strong AI performance.

INFRASTRUCTURE: ROBUST TECHNOLOGY ENABLES POWERFUL PERFORMANCE

Infrastructure and hardware make it possible to host data, models, and software needed for AI solutions. Processing, memory, networking, and storage tasks are particularly enabled via infrastructure. Processing and logic devices such as CPUs, GPUs, FPGAs and/or ASICs, temporary storage and long-term storage solutions, and devices that enable connectivity make up the essential hardware bundle for AI[7]. Improving the performance of AI systems is inevitably tied to hardware capacity.

However, it is doubtful that the capacity of hardware can be increased exponentially without innovative interventions. For instance, following Moore's law,[8] there are concerns that we may be reaching the limits of the physical capacity of computing power [9,10]. In order to power the AI of the future, bleeding-edge infrastructure solutions will be essential.

CODE: CLEAR, MANAGEABLE, AND EFFICIENT CODE STREAMLINES AI IMPLEMENTATION

Code that underlies AI systems is crucial to ensure that solutions function as expected and do so without error. Good code is reliable, clear, and consistent. Availability of documentation makes code bases easier to maintain. The ever-evolving AI ecosystem requires high-performance software, making code efficiency critical. Even the slightest increase in code efficiency can accelerate application speed and boost productivity and profits.

A study on the business impact of code quality reported that the Time-in-Development for Alert level 1 codebases is 124% more than that of Healthy level code[11]. In a survey of C-level executives conducted by Stripe, it was reported that bad code costs companies $85 billion annually[12]. Technical debt in machine learning[13] piles extremely fast that bad quality code can set even the most experienced teams back for half a year.

TALENT: HIGH-QUALITY TALENT DEVELOPS THE BEST-PERFORMING AI SYSTEMS

AI is envisioned, developed and maintained by data scientists, software engineers and various other personnel in the AI ecosystem. Engaging the best talent leads to AI solutions that are innovative, efficient, and profitable. LinkedIn 2020 Emerging Jobs Report UK identifies Artificial Intelligence Specialist as UK's top emerging job, highlighting the value and demand for AI talent[14].

While high-quality AI solutions rest on the above pillars, there are additional underlying factors of the AI ecosystem that often go unnoticed in the flurry of activities in a commercial development setting. Code optimisation is one such critical component.

The Role of Code Optimisation in AI Solutions is often Overlooked

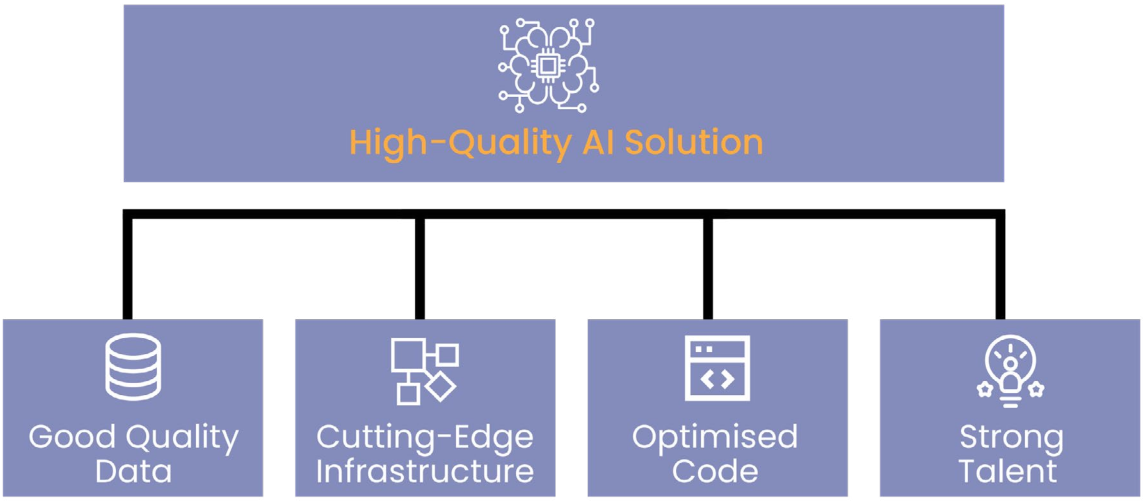


Figure 3: Quality of inputs impact the quality of the AI solution

In essence, code optimisation is writing or rewriting code in such a way that it utilises minimal time, energy, and computing resources to be executed. The benefits of code optimisation are manifold. Optimisation makes code bases cleaner, clearer, and more consistent, making software perform with greater efficiency.

End-users of this software receive an enhanced user experience and better results. Code optimisation also makes code more readable, ultimately making it easier for multiple stakeholders to collaborate easily on a single code base. Contrary to hardware acceleration, focusing on accelerating software (code) can also lead to greater returns to scale. An analysis by Intel estimates that "even a 10X gain in performance through software AI accelerators can lead to approximate cost savings of millions of dollars a month"[15].

Despite these many positives, code optimisation often receives mixed responses in the software development space. On the one hand, as we have noted, optimisation can dramatically boost programme performance. On the other hand, efforts at code optimisation, particularly manual code optimisation, can consume an unsustainable amount of developer time and resources. Sub-par optimisation can also lead to poor programme performance. Given these mixed sentiments, optimising code bases may get deprioritised, with minor inefficiencies snowballing into larger technical debt.

In order to uncover the potential of code optimisation to boost AI performance, it is important to first delve into practices of writing, implementing and maintaining code bases. In the next section we look at code underlying AI solutions, aiming to understand best practices in the industry.

Code Quality is Essential for a Thriving AI Ecosystem

In the previous section, we highlighted that high-quality AI solutions lie on four pillars: data, infrastructure, code, and talent. In this section, we explore further the value of code quality in the AI ecosystem, including discussing industry best practices.

GOOD QUALITY CODE BASES ARE EASY TO DISTINGUISH FROM SUB-PAR CODE

There are a handful of features that are characteristic of a good quality code base, in the AI ecosystem as well as more generally. We discuss them below:

1) RELIABILITY

When software is developed to achieve a given task, it is vital that the written code does what it is supposed to do, without any major failures. Having unreliable code seriously undermines the validity and efficiency of software. Fixing faulty code can take up a significant component of the development timeline, wasting valuable time and resources.

2) AVAILABILITY OF DOCUMENTATION

In an industry environment, multiple stakeholders are involved in taking software from initiation to deployment. High-quality code includes documentation that allows members of different teams to read and understand what a developer is aiming to achieve with a given codebase. In the event of a program failure, documentation makes the bug-fixing process easier. development timeline, wasting valuable time and resources.

3) CLARITY AND CONSISTENCY

Model code that is clear and consistent enables easy reading and comprehension of software. Having code that follows a clear and consistent style prevents development teams from investing large amounts of time, resources, and energy into understanding and implementing models.

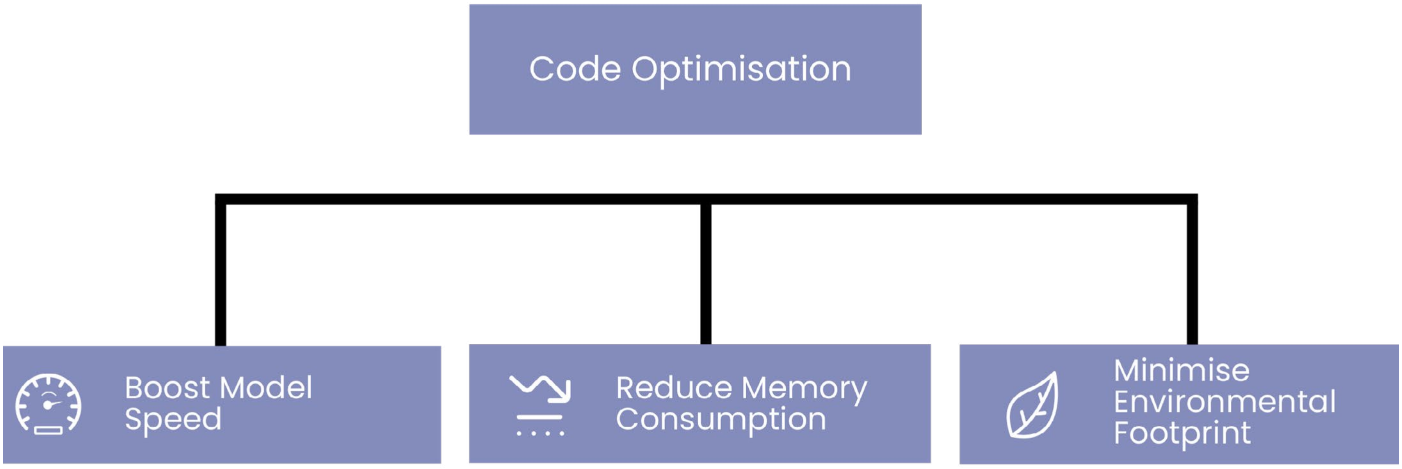
4) VERSIONING DATA AND CODE

As the product goes through the phases of integrations, sometimes it is hard to keep track of the release-candidates if no proper source-code and data versioning system is in place. Research shows that companies that adopt some sort of a versioning system are able to reduce the release timeline from once every three months to hundreds of times per day[16].

Code optimisation ensures that code bases meet the criteria for high-quality code, thereby enabling the development of cutting-edge AI solutions.

Optimised Code Bases Drive Business Value

Code optimisation allows users to reach high-quality code bases, while also successfully navigating the challenges of optimising code. Three specific improvements that code optimisation provides are: (1) boosting model speed, (2) reducing memory consumption of models, and (3) minimising the environmental impact of AI solutions. We look at specific business uses of these improvements below.



Code Optimisation

Boost Model Speed

Reduce Memory Consumption

Minimise Environmental Footprint

Boost Model Speed

CAPITALISING ON PROFITABLE TRADING OPPORTUNITIES

Due to the ability to analyse a large and varied body of information, machine learning-based models are used in trading to make more accurate and swift decisions. With optimisation, machine learning models can perform quicker, generating trading decisions at a faster rate. It is reported that an advantage of one millisecond can be worth $100 million a year in trading . This emphasises the value of speed in the machine learning model.

IMPROVING CUSTOMER EXPERIENCE FOR LOWER CHURN

Most consumers now default to using digital channels to engage with a business. While web-based services and mobile apps can make a business and its products and services more appealing to customers, sub-par digital support can also turn customers away easily. Research by Booking.com[18] has shown that around a 30% increase in latency costs more than 0.5% in conversion rate. A drop in conversion leads to reduced revenue and can translate to millions lost in profit. For a user to feel that a system is reacting instantaneously, the system must ideally display a response time of 0.1 seconds[19] Code and model optimisation ensures that AI applications meet these ideal performance rates, providing users with a seamless experience, leading to greater customer retention and lower churn rates.

ASSISTING WITH LIFE-CRITICAL DECISION-MAKING

Emerging AI applications for autonomous vehicles and healthcare are not only novel but they also make life-critical decisions. In a recently launched project, In the Moment (ITM)[20], the US Defense Advanced Research Projects Agency (DARPA) is aiming to develop AI systems that make critical decisions in environments that are rapidly evolving, uncertain, and have no “ground truth”. For example, in medical triaging[21], due to limitations in resources, medical professionals are required to decide which emergency care patients to prioritise, based on the fatality of their illness or injury. AI solutions used in such critical situations must display robust results at speed.

Latency in systems that make decisions in real-time can cause serious harm and severely jeopardise the credibility of the system. In 2016, a Tesla on autopilot crashed into a white truck, failing to distinguish the vehicle due to its colour[22] The collision led to much hesitation about the safety of self-driving vehicles. Such incidents reiterate the importance of machine learning model speed and accuracy in AI systems. Optimised machine learning models can reduce inefficiencies in life-critical AI applications, allowing for more accurate decisions to be made faster.

IMPROVING THE PERFORMANCE OF EDGE DEVICES

Edge devices are becoming commonplace across industries, with applications such as security cameras, drones, and wearable tech. Edge devices and the Internet of Things (IoT) allow greater capacity for scalability, accessibility, and speed, making them extremely useful for businesses. Increasingly, more analytics functions take place within these devices, particularly to reduce latency in decision-making. For instance, data can be processed within an edge-device itself to mitigate risks arising from reduced network capacity[23]. However, since edge devices usually have memory and power constraints, code optimisation can be used to improve machine learning model performance to suit the requirements of the targeted devices. This will lead to timely and accurate analyses, allowing businesses to make the best use of edge AI.

Reduce memory usage

DEPLOYING AT THE EDGE

The Internet of Things (IoT) and edge[24] devices are seeing increased demand across multiple fields. Industries such as healthcare, travel, and retail take advantage of a range of edge devices, including wearable tech, sensors, and security equipment. For instance, stores such as Amazon Fresh utilise novel IoT solutions to provide a seamless shopping experience to users. Amazon systems are powered by edge devices such as cameras, weight and pressure sensors, and infrared sensors[25], paired with machine learning solutions[26]. Often, in the case of IoT, machine learning models are embedded on edge devices to generate faster analytics and insights in real-time.

However, IoT devices are compact and capacity constrained, which makes memory and energy optimisation critical to deploy machine learning models efficiently at the edge. For instance, the RAM of a series 7 (latest) Apple watch is 1GB[27], whereas generally, in order to run a deep learning model, at least 16 GB of RAM is recommended[28]. Therefore, executing a machine learning model on an edge device requires a high level of memory efficiency, especially if the device were to provide useful real-time insights such as health analytics or fall detection. Optimisation significantly contributes to making machine learning models perform optimally at the edge. By eliminating inefficient elements and streamlining layout, code optimisation refines machine learning models to best suit edge devices.

REDUCING COMPUTATIONAL COST

Utilising an AI-based solution to enhance a business process contains two essential steps: (1) develop, train, and test the system, and (2) deploy and maintain the solution. Memory is a key element required for these tasks. The alternative is to train on the cloud, reducing the need for memory on local machines, yet cloud computing can still require large financial investments. Training bleeding edge models can cost companies over a million dollars[29]; small-scale businesses with low budgets may even have to forego certain functionalities of AI solutions to make projects financially feasible.

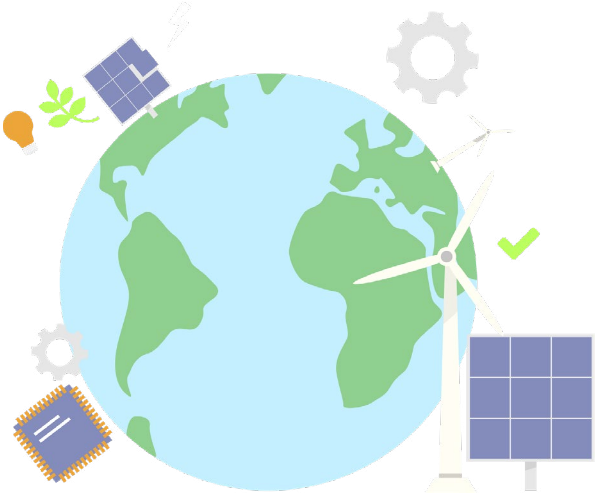
Code optimisation is one way to reduce a model's memory consumption, minimising the need to invest in additional resources, and thereby cutting costs. In particular, machine learning models used for computationally heavy tasks such as image recognition demand the usage of GPUs, which are further limited in volatile memory. Optimising machine learning model code used for such tasks will save businesses significant computational costs.

BOOSTING PERFORMANCE

Maintainability, reliability, clarity, and consistency are characteristic of high-quality code. Compromising code quality not only reduces the performance of a programme but can also lead to memory leakage. For instance, low-quality code may contain redundant variables, which, especially during iterative processes, can lead to significant inefficiencies in volatile memory usage.

A report by Consortium for Information and Software Quality (CISQ)[30] estimates the total cost of poor software quality in the US in 2020 as a staggering $2.08 trillion[31] highlighting the value of code quality in developing software. Code optimisation reduces redundancies in code bases, thereby curtailing memory leaks and boosting performance.

Minimise environmental footprint



The rapid uptake of AI and machine learning we see now is predominantly due to their ability to generate faster insights with greater precision. One unfortunate downside of AI-based analytics is its environmental cost. In order to improve the predictive capacity of a machine learning model, a business may acquire hardware with better performance, but this can only come at a higher level of energy consumption, and often with a more significant carbon footprint.

Digital waste[32] is another driver of the negative environmental impact of AI. One instance is when data and algorithms developed as a part of an AI solution lie idle on cloud servers once their purpose has been served. AI debris still costs the environment. The cloud has a greater carbon footprint than the airline industry, and a single data centre can consume the equivalent electricity of 50,000 homes.[33]

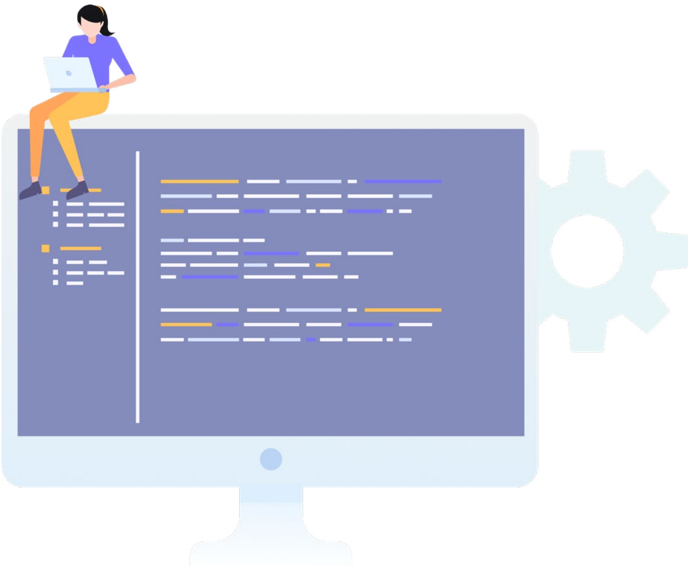
By reducing memory usage, code optimisation can reduce the energy demands of training, testing, deploying, and storing machine learning models, ultimately leading to greener AI.

Thinking green is just the first step. AI models need to be optimised at the code level to achieve higher energy efficiency. By improving inefficiencies in code, users can significantly increase their model's efficiency by over 50% without compromising accuracy and other metrics.

Efficient AI not only reduces carbon emissions but can also lead to better performance at the production level. Imagine being able to gain quicker insights to secure profitable business opportunities or having faster running apps on your phone without draining the battery.

Owning Model Code is a Privilege not Everyone can Enjoy

In addition to direct benefits of code quality, we take this section to look at the value of owning model code, particularly when developing AI solutions. A benefit of in-house, custom-built machine learning models and AI solutions is that a company will have complete ownership of model code. However, as companies resort to automated AI and machine learning solutions, they may lose access to model code, thereby leading to more complex issues with their AI systems.



IMPROVING TRANSPARENCY AND EXPLAINABILITY

Often machine learning models are opaque, where users have minimal knowledge of the decision-making process taking place within the model. This can be particularly challenging in industries that require strict regulatory compliance, such as healthcare and finance. Owning model code significantly improves transparency, putting businesses in a better position to meet regulatory compliance.

CUSTOMISABILITY AND FLEXIBILITY

Off-the-shelf solutions are often developed to meet a set of generic criteria, so most companies may feel the need to customise models to be sensitive to ever-changing business demands. Having complete access to model code makes customisation feasible and painless. With access to model code, users are also able to retrain the model in case there are significant changes to input data.

INTEGRATING INTO AVAILABLE ARCHITECTURE AND HARDWARE

Organisations have varying system architectures with unique performance needs and hardware restrictions. Having end-to-end code allows engineering teams to customise and integrate these models easily with the rest of their data pipelines. It also allows for easy integration of existing monitoring and evaluation tools.

Ready-made machine learning systems may also require businesses to upgrade hardware, which comes at a significant financial cost. Having ownership of model code lets companies scale their solutions to suit existing hardware.

SPECIALISATION AND EXTENSION

A machine learning-based solution cannot be left to stagnate upon deployment. Companies would need to continuously evolve the technology to stay relevant in the industry. They may even feel the need to develop specialised solutions from the initial models. For instance, a financial firm that creates a cutting-edge machine learning-based trading tool may choose to further develop this solution and offer it as a specialised service. Such scenarios will only be possible if in-house teams have complete ownership of the model code.

REDUCING DEPENDENCIES

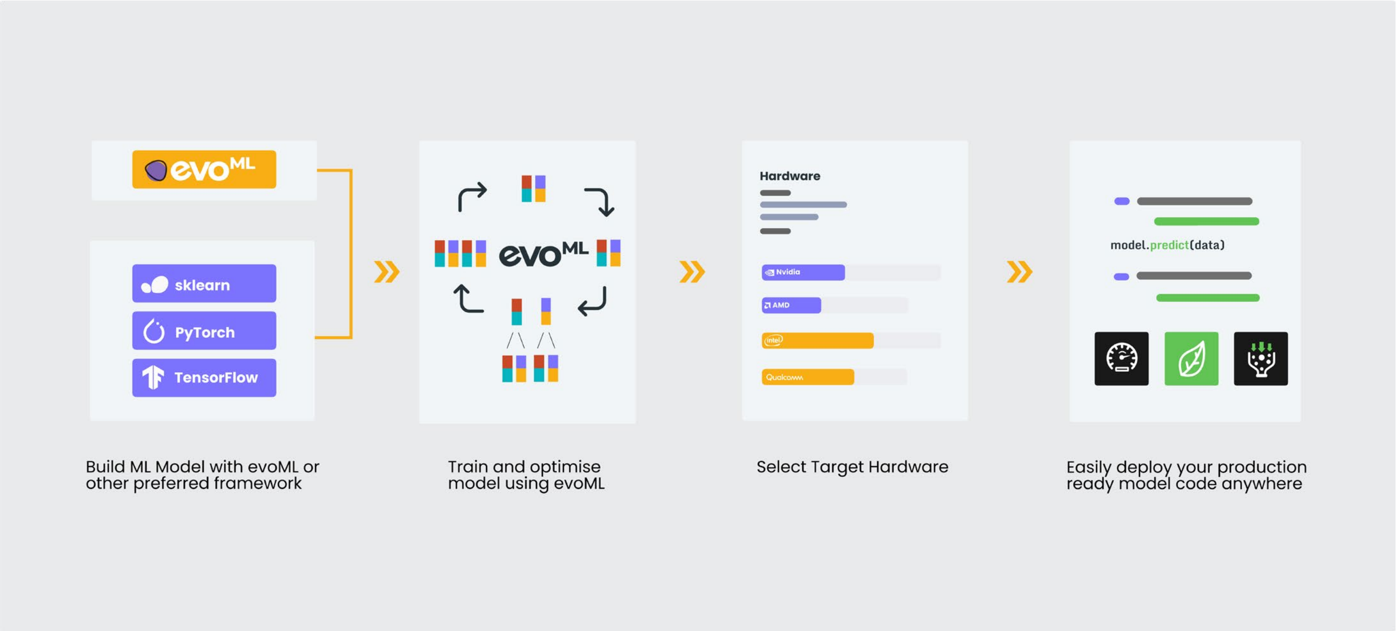
Developing and maintaining a data science pipeline involves many stakeholders, both within an organisation and outside. In such settings, individual delays can hamper the progress of the entire pipeline. Having complete ownership of model code generated by an autoML process helps to reduce dependencies in the data science pipeline.

On the one hand, having full access to production-ready model code reduces the need to depend on the input of individual developers in deployment. On the other hand, with access to model code, companies can make independent decisions about their models in the case of delays caused by external stakeholders involved in the autoML process.

SAFEGUARDING INTELLECTUAL PROPERTY

AI is evolving at a remarkable speed. Legal and regulatory processes need to progress in equal measure for companies to confidently incorporate AI into their work. In 2021 the UK government conducted a consultation to seek “evidence and views on a range of options on how AI should be dealt with in the patent and copyright systems”[34]. These measures emphasise the value of legal and regulatory protection in the AI sphere. Businesses, especially those in the finance and trading industries, have a stronger need to protect their processes and output. Owning model code makes it easier for businesses to navigate the AI intellectual property space.

TurinTech Work in Code Optimisation



Build ML Model with evoML or other preferred framework

Train and optimise model using evoML

Select Target Hardware

Easily deploy your production ready model code anywhere

TurinTech has been extensively researching code optimisation and its implementations in the AI ecosystem for over 10 years. One of our key research applications is evoML[35], a code optimisation platform that empowers businesses to automatically 1) build efficient ML model code from raw data 2) optimise the performance of existing ML model code and 3) optimise the speed of generic code.

evoML works on users' code bases to automatically detect and reduce inefficiencies. The technology is based on genetic algorithms, where code goes through a process of evolution until an optimal solution is reached. evoML also encapsulates multi-objective optimisation[36],allowing businesses to optimise parameters that they specifically wish to address, such as accuracy, inference speed, explainability and other custom metrics.

Our research showed that there are significant benefits in using automated code optimisation to improve the quality of code. In one of our studies, we saw optimisation resulting in up to 46% improvement in execution time, 44.9% improvement in memory consumption and 49.7% improvement in CPU usage[37].

46% Improvement In Execution Time

45% Reduction in Memory Usage

50% Reduction in CPU Usage

Our proprietary code optimisation can improve the performance of code bases significantly, at a fraction of the cost. Reducing execution time and the computational cost of running programmes also means making considerable savings on finances, resources, and energy usage.

With evoML, code optimisation also becomes more accessible to diverse data science and production teams across industries, allowing them to integrate efficient AI into everyday wok practices with ease.

Reference

1 https://www.forbes.com/sites/shoptalk/2017/03/28/6-key-takeaways-from-shoptalk-2017-an-insiders-perspective/?sh=6b1ce16224ab

2 https://transformainsights.com/blog/iot-dominate-ai-forecast

3 https://www.bcg.com/capabilities/digital-technology-data/artificial-intelligence

4 https://transformainsights.com/blog/iot-dominate-ai-forecast

5 https://www.gartner.com/en/newsroom/press-releases/2021-11-10-gartner-says-cloud-will-be-the-centerpiece-of-new-digital-experiences

6 https://www.precedenceresearch.com/edge-computing-market

7 https://www.mckinsey.com/industries/semiconductors/our-insights/artificial-intelligence-hardware-new-opportunities-for-semiconductor-companies

8 https://www.theguardian.com/commentisfree/2020/jan/11/we-are-approaching-the-limits-of-computer-power-we-need-new-programmers-n-ow

9 http://arxiv.org/abs/2203.04374

10 https://www.technologyreview.com/2020/02/24/905789/were-not-prepared-for-the-end-of-moores-law/

11 https://arxiv.org/pdf/2203.04374.pdf

12 https://stripe.com/files/reports/the-developer-coefficient.pdf

13 https://towardsdatascience.com/technical-debt-in-machine-learning-8b0fae938657

14 https://business.linkedin.com/talent-solutions/emerging-jobs-report/emerging-jobs-report-uk

15 https://venturebeat.com/ai/software-ai-accelerators-ai-performance-boost-for-free/

16 https://www.netsolutions.com/insights/why-do-great-product-companies-release-software-to-production-multiple-times-a-da

17 https://www.itworldcanada.com/sponsored/in-financial-services-theres-no-need-to-waste-a-millisec ond

18 https://blog.kevinhu.me/2021/04/25/25-Paper-Reading-Booking.com-Experiences/bernardi2019.pdf

19 https://www.nngroup.com/articles/response-times-3-important-limits/

20 https://www.darpa.mil/news-events/2022-03-03

21 https://www.investopedia.com/terms/t/triage.asp

22 https://www.theguardian.com/technology/2016/jun/30/tesla-autopilot-death-self-driving-car-elon-musk

23 https://learn.microsoft.com/en-us/archive/msdn-magazine/2018/july/machine-learning-machine-learning-with-iot-devices-on-the-edge

24 https://www.intel.co.uk/content/www/uk/en/edge-computing/edge-devices.html

25 https://appft.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&u=%2Fnetahtml%2FPTO%2Fsearch-adv.html&r=1&p=1&f=G&l=50&d=PG01&S1=(Steven.IN.+AND+Kessel. IN.)&OS=IN/Steven+and+IN/Kessel&RS=(IN/Steven+AND+IN/Kessel)

26 https://www.amazon.science/latest-news/how-amazon-fresh-is-leveraging-scientific-innovation-to-meet-increased-consumer-demand

27 https://en.wikipedia.org/wiki/Apple_Watch

28 https://towardsdatascience.com/another-deep-learning-hardware-guide-73a4c35d3e86

29 https://venturebeat.com/ai/microsoft-and-nvidia-team-up-to-train-one-of-the-worlds-largest-language-models/

30 https://www.synopsys.com/content/dam/synopsys/sig-assets/reports/CPSQ-2020-report-final.pdf

31 Estimates have been made based on: (1) cost of unsuccessful IT/software projects, (2) cost of poor quality in legacy systems, (3) cost of operational software failures, (4) cost of cybersecurity and technical debt. See section 3 of the report (https://www.synopsys.com/content/dam/synopsys/sig-assets/reports/CPSQ-2020-report-final.pdf) for complete methodology.

32 https://revolutionized.com/digital-waste/

33 https://thereader.mitpress.mit.edu/the-staggering-ecological-impacts-of-computation-and-the-cloud/

34 https://www.gov.uk/government/consultations/artificial-intelligence-and-ip-copyright-and-patents

35 https://www.turintech.ai/product-and-features/

36 https://www.turintech.ai/optimising-ai-with-multiple-objectives/

37 https://www.researchgate.net/publication/317558315_Darwinian_Data_Structure_Selection

About TurinTech

TurinTech is the leader in code optimisation for machine learning and other data-heavy applications, helping businesses become more efficient and sustainable by accelerating time-to-production and reducing development and compute costs.

Powered by proprietary AI research, TurinTech's evoML platform empowers businesses to automatically 1) build efficient ML model code from raw data 2) optimise the performance of existing ML model code and 3) optimise the speed of generic code.

Learn more about TurinTech

Follow us on social media:

Trusted by:

aws NVIDIA Microsoft For Startups Exasol

UCL Google Cloud brytlyt

Research Partners and Awards:

UCL Imperial College London King's College London University of Surrey

European Commission | Horizon 2020 European Union funding for Research & Innovation Innovate UK TECH NATION APPLIED AI

Email :
contact@turintech.ai

Website :
turintech.ai

Address :
Citypoint, 1 Ropemaker Street,
London, EC2Y 9AW, UK

TURINTECH