

FEYNMAN LECTURES ON COMPUTATION

Study Guide

Feynman Lectures On Computation

Feynman Lectures on Computation: Exploring the Intersection of Physics and Computer

Science

feynman lectures on computation offer a fascinating glimpse into the mind of one of

the 20th century's most brilliant physicists, Richard Feynman, as he investigates the fundamental principles behind computation. These lectures bridge the gap between physics and computer science, illustrating how the laws of nature influence our understanding of computation and how computational concepts can be viewed through the lens of physical systems. If you are curious about the theoretical underpinnings of computers, quantum mechanics, and information theory, the Feynman Lectures on Computation provide a rich and engaging resource.

Who Was Richard Feynman and Why Are His Lectures Important?

Richard Feynman was a Nobel Prize-winning physicist renowned for his ability to distill complex scientific ideas into clear, intuitive explanations. While he is primarily known for his contributions to quantum electrodynamics and particle physics, his curiosity extended deeply into the realm of computation. In the 1980s, Feynman delivered a series of lectures that addressed some of the most profound questions about how computation

works and how it relates to physical laws.

These lectures are significant because they challenge traditional views of computing machines by incorporating principles from thermodynamics, quantum physics, and information theory. His approach has influenced the development of quantum computing and inspired generations of scientists and engineers to think differently about what computation fundamentally is.

Key Themes in the Feynman Lectures on Computation

The Feynman Lectures on Computation are a treasure trove of ideas, and several key themes run throughout the material. Understanding these themes helps contextualize why these lectures remain relevant in today's rapidly evolving tech landscape.

Computation as a Physical Process

One of the central ideas Feynman explores is that computation is not just an abstract mathematical process but a physical one. Every computational operation happens within a physical system, whether it's a silicon chip or a quantum particle. This perspective forces us to consider the laws of physics—such as energy conservation and entropy—when designing computational devices.

For example, Feynman discusses the thermodynamic cost of computation, highlighting Landauer's principle, which states that erasing information has a minimum energy cost. This insight is crucial for understanding the limits of classical computers and motivates the search for more efficient computation methods.

Reversibility and Quantum Computing

Feynman's fascination with quantum mechanics led him to explore reversible computation, where computational steps can be undone without loss of information. This is not just a theoretical curiosity; reversible computation is a necessary condition for quantum computing, which promises exponential speed-ups for certain problems.

In his lectures, Feynman introduces the idea that quantum computers operate fundamentally differently from classical ones, manipulating quantum bits (qubits) that can exist in superpositions of states. This concept laid the groundwork for the development of quantum algorithms and the field of quantum information science.

Limits of Computation and Universality

Another important topic covered in the lectures is the concept of universality—the idea that a single computational model can simulate any other. Feynman discusses the Turing machine, the classical model of computation, and extends this to quantum machines, showing how quantum computers could theoretically simulate any physical system.

He also touches on the limitations of computation, such as undecidability and the halting problem, reminding us that no matter how powerful our machines become, some problems remain inherently unsolvable.

How the Feynman Lectures Influenced Modern Computing

The insights from the Feynman Lectures on Computation have had a profound impact on various fields:

Inspiration for Quantum Computing

Feynman is often credited as one of the pioneers of quantum computing. His lectures helped clarify why classical computers struggle to simulate quantum systems efficiently and proposed that quantum computers could overcome these challenges. This vision has catalyzed decades of research into quantum algorithms, error correction, and hardware development.

Advancing Information Theory and Thermodynamics

By linking computation with thermodynamics, Feynman's work encouraged scientists to rethink energy efficiency in computing. Today, as we push toward smaller and more powerful devices, understanding these physical limits helps engineers design hardware that balances performance with energy consumption.

Educational Impact

The Feynman Lectures on Computation remain popular teaching resources. Their conversational tone and engaging explanations make complex ideas accessible to students and professionals alike. They encourage critical thinking about the foundational principles underlying technology we often take for granted.

Exploring the Content of the Feynman Lectures on Computation

The lectures cover a broad range of topics, each connecting physical laws with computational theory. Here are some highlights that showcase the depth and breadth of Feynman's exploration:

Classical Computation and Logic Gates

Feynman starts with the basics, examining how classical computers operate using logic

gates?AND, OR, NOT?and how these can be combined to perform complex operations.

He emphasizes that these gates are physical entities subject to the laws of physics, which has implications for their efficiency and reliability.

Information and Entropy

The lectures delve into the nature of information itself, relating it to entropy, a fundamental concept in thermodynamics. Feynman explains how information can be measured, manipulated, and transmitted, and discusses the physical cost associated with these processes.

Quantum Mechanics and Computation

Perhaps the most groundbreaking sections deal with quantum theory. Feynman introduces the principles of quantum superposition and entanglement, explaining how these phenomena can be harnessed to perform computations impossible for classical systems. He also discusses the challenges of building quantum computers, including error correction and decoherence.

Practical Considerations and Future Directions

Although theoretical, Feynman's lectures also touch on practical aspects such as how to design reversible computing circuits and the potential architectures for quantum machines. He speculates on future developments, many of which continue to influence ongoing research.

Tips for Getting the Most Out of the Feynman Lectures on

Computation

If you're interested in diving into these lectures, here are a few suggestions to enhance your learning experience:

Familiarize Yourself with Basic Concepts: A solid understanding of classical

1.

computation, basic physics, and linear algebra will make the material more accessible.

Take Notes Actively: Feynman's style is conversational, but the concepts can be

2.

dense. Writing down key points helps reinforce understanding.

Supplement with Modern Resources: Since the lectures were delivered several

3.

decades ago, pairing them with current textbooks or online courses on quantum computing can provide additional context and updates.

Engage in Discussions: Joining forums or study groups focused on computation

4.

and quantum mechanics can deepen your grasp through shared insights and questions.

Where to Find the Feynman Lectures on Computation

Originally presented as a series of lectures, these talks have been compiled into books

and recorded formats accessible online. Some are available through academic institutions, while others can be found on platforms dedicated to science education. Exploring these materials not only enriches your understanding of computation but also offers a window into Feynman's unique approach to science.

The Feynman Lectures on Computation continue to inspire those fascinated by the fundamental nature of computing. By combining physics and computer science, these lectures reveal the profound connections between the physical universe and the abstract world of algorithms and information. Whether you are a student, researcher, or enthusiast, delving into these lectures invites you to rethink what computation really means in our universe.

Question

Answer

What are the Feynman
Lectures on Computation?

The Feynman Lectures on Computation are a series of lectures and writings by physicist Richard P. Feynman that explore the fundamental principles of computation, linking physics and computer science.

Who authored the Feynman
Lectures on Computation?

The Feynman Lectures on Computation were authored by Richard P. Feynman, along with co-authors Anthony J.G. Hey and Robin W. Allen, who helped compile and edit the lectures.

What topics are covered in the Feynman Lectures on Computation?

The lectures cover topics such as classical computation, quantum computing, reversible computation, complexity theory, and the physical limits of computation.

Why are the Feynman Lectures on Computation considered important?

They are important because they provide deep insights into the relationship between physics and computation, emphasizing the physical nature of information and inspiring developments in quantum computing.

Are the Feynman Lectures on Computation suitable for beginners?

While accessible, the lectures assume some background

in physics and computer science, making them suitable for advanced undergraduates, graduate students, or professionals interested in computational theory.

Where can I access the

Feynman Lectures on

Computation?

The lectures are available in book form titled 'Feynman Lectures on Computation' published by Addison-Wesley, and some materials can be found online through academic websites and libraries.

How do the Feynman

Lectures on Computation

relate to quantum

computing?

Feynman's lectures were pioneering in suggesting that quantum mechanical systems could be used to perform computation more efficiently than classical computers, laying groundwork for quantum computing.

What is a key concept

introduced by Feynman in

these lectures?

A key concept is reversible computation, where computational processes can be reversed without loss of information, which has implications for energy efficiency in computing.

Can the Feynman Lectures on Computation help in understanding modern computing technologies?

Yes, they provide foundational knowledge about the principles underlying modern computing technologies, especially in areas like quantum computing, computational complexity, and the physical limits of computers.

Feynman Lectures on Computation: Exploring the Intersection of Physics and Computer Science

feynman lectures on computation represent a seminal series of talks and writings by

the renowned physicist Richard P. Feynman, delving into the fundamental principles underlying computation. These lectures are not merely a technical guide but a profound investigation into how the laws of physics intertwine with the theory and practice of

computing. As interest in quantum computing and theoretical computer science continues to grow, revisiting the insights from Feynman's exploration proves both timely and invaluable.

Understanding the Feynman Lectures on Computation

Richard Feynman, celebrated for his groundbreaking work in quantum mechanics and particle physics, ventured into the realm of computation during the 1980s. His lectures, compiled in the influential book **Feynman Lectures on Computation**, articulate a vision that computing is not just an abstract mathematical process but a physical phenomenon constrained and enabled by the laws of nature.

Unlike traditional computer science texts focused solely on algorithms or hardware design, Feynman's approach integrates physics—classical and quantum—to explain what computation fundamentally entails. This perspective has been especially relevant in the context of quantum computing, where classical computational paradigms are challenged by quantum mechanics.

The Core Themes of the Lectures

At the heart of the Feynman lectures on computation are several critical themes:

Physical Limits of Computation: Feynman examines how thermodynamics,

1. information theory, and quantum physics impose fundamental limits on how efficiently and reliably computers can operate.

Reversible Computing: A concept championed by Feynman, reversible

computing

2.

explores computational processes that can run backward without loss of information, thereby reducing energy dissipation according to Landauer's principle.

Quantum Computation: Feynman was among the pioneers who suggested that

3.

quantum mechanical systems could be harnessed to perform computations beyond the capability of classical machines.

Information as a Physical Entity: The lectures emphasize that information is not

4.

an abstract concept divorced from reality but a physical quantity, subject to the same laws that govern matter and energy.

Impact on Modern Computational Theory and Practice

Feynman's insights have had a lasting impact on both theoretical computer science and emerging technologies. The resurgence of interest in quantum computing owes much to the foundational ideas presented in these lectures. His argument that simulating quantum systems efficiently may require quantum computers has shaped research agendas worldwide.

Moreover, the exploration of reversible computing continues to influence attempts at designing ultra-low-power computational devices. While still largely experimental, the principles laid out in the lectures provide a roadmap for reducing energy consumption in

computing, an ever-important consideration as data centers and processors become more power-hungry.

Comparing Feynman Lectures with Other Foundational Works

When positioned alongside other classic texts like Alan Turing's foundational papers or Claude Shannon's information theory, the Feynman lectures offer a unique fusion. While Turing defined computability and Shannon quantified information transmission, Feynman contextualized these concepts within physical reality. This multidisciplinary approach broadens the scope of computation beyond abstract machines and signals to encompass the physical universe itself.

Key Features and Contributions

The *Feynman Lectures on Computation* stand out for several distinctive features:

Clarity and Accessibility: Despite dealing with complex interdisciplinary

1. concepts, Feynman's characteristic clarity makes the material approachable to a wide audience ranging from physicists to computer scientists.

Interdisciplinary Approach: The lectures bridge gaps between physics, computer

2. science, and information theory, encouraging holistic understanding.

Forward-Looking Vision: Feynman anticipated future challenges and

3.

opportunities in computing, particularly quantum computing and energy-efficient computation, well before they became mainstream research areas.

Strengths and Limitations

The lectures' strengths lie in their innovative perspective and foundational depth. They challenge readers to rethink computation at a fundamental level, making them invaluable for researchers interested in the theoretical underpinnings of computing technology.

However, some limitations are noteworthy:

The book's age means that certain aspects lack coverage of recent advances in

1.

quantum algorithms or hardware implementations.

Its conceptual density and interdisciplinary nature may present a steep learning

2.

curve for those without a background in physics.

Practical applications or step-by-step guides to computation are limited, as the

3.

focus remains theoretical and philosophical.

Legacy and Continuing Relevance

More than three decades since their publication, the *Feynman Lectures on Computation* continue to inspire. Universities include them in advanced curricula focused on quantum information science and computational physics. Researchers cite Feynman's work when

discussing the physical limits of computation or developing new quantum algorithms.

The lectures also serve as a historical lens, showing how visionary thinking can presage entire new fields. The notion that computation is intrinsically tied to physical law has now become a cornerstone in designing next-generation technologies.

While the landscape of computation has evolved rapidly, the fundamental questions posed by Feynman—about what computation truly means, how it can be realized physically, and what its limits might be—remain as relevant as ever. His work encourages ongoing inquiry and innovation at the intersection of physics and computing, a domain that will undoubtedly shape the technological future.

Richard Feynman, computation theory, quantum computing, computer science, algorithms, computational physics, information theory, Turing machines, quantum mechanics, complexity theory

deep learning adaptive computation and

Deep Learning Adaptive Computation And Machine Learning

Deep Learning Adaptive Computation and Machine Learning: Unlocking Smarter AI Systems

deep learning adaptive computation and machine learning are revolutionizing the way

artificial intelligence systems learn and process information. While the phrase might sound technical, at its core, it represents a dynamic approach to AI that allows models to

adjust their computational effort based on the complexity of the task at hand. This concept is reshaping machine learning by making algorithms more efficient, scalable, and capable of handling real-world challenges with greater intelligence.

In this article, we'll explore the fascinating world of deep learning adaptive computation and machine learning, unraveling what it means, how it works, and why it matters for the future of AI. We'll also dive into related topics such as neural networks, resource optimization, and the latest advances in adaptive learning techniques.

Understanding Deep Learning Adaptive Computation and Machine Learning

At its essence, deep learning adaptive computation refers to the ability of a neural network or machine learning system to dynamically adjust the amount of computational resources it uses when making predictions or decisions. Unlike traditional models that apply the same computational effort uniformly across all inputs, adaptive models tailor their processing based on the difficulty or uncertainty of each input example.

Machine learning, a truncated term often linked to machine learning, encompasses these adaptive strategies to improve learning performance. This approach is closely tied to the idea of conditional computation, where parts of a neural network are selectively activated to save time and energy without sacrificing accuracy.

Why Adaptive Computation Matters

The motivation behind adaptive computation in deep learning stems from several

challenges:

Efficiency: Large deep learning models can be computationally expensive, especially on limited-resource devices like smartphones or embedded systems.

Scalability: As models grow in size, processing every example with equal intensity is not practical.

Real-time Performance: Applications like autonomous vehicles or speech recognition require timely responses, making adaptive computation invaluable.

Energy Consumption: Reducing unnecessary computation helps in building environmentally friendly AI systems.

By enabling models to decide how much effort to spend per task, adaptive computation optimizes resource usage and accelerates inference times.

Key Techniques in Deep Learning Adaptive Computation

Several methodologies have emerged to implement adaptive computation within deep learning frameworks. Let's break down the most prominent ones.

1. Early Exiting Networks

Early exiting allows a neural network to produce an output at intermediate layers if the prediction confidence is sufficiently high. Instead of always passing data through all layers, the model can exit early, saving computation time.

For example, in image classification, if the network is confident after processing the first few layers that the image is a cat, it can avoid further costly computations.

2. Dynamic Neural Networks

Dynamic neural networks adjust their architecture or parameters on-the-fly based on the input. This can involve skipping layers, dynamically changing the number of neurons activated, or routing information through different sub-networks.

These networks are highly flexible and can adapt their complexity to the requirements of each input.

3. Reinforcement Learning for Computation Control

Some adaptive models use reinforcement learning to decide how much computation to allocate. The system learns policies that balance accuracy with computational cost, optimizing for overall performance.

This approach is powerful for tasks where the computational budget is limited and must be intelligently managed.

Applications of Adaptive Computation in Machine Learning

The integration of deep learning adaptive computation and machine learning has opened new avenues in several domains:

Natural Language Processing (NLP)

Adaptive computation helps NLP models process sentences of varying complexity more efficiently. For instance, simple sentences might require fewer layers of processing than complex ones, enabling faster translation or sentiment analysis with reduced resource usage.

Computer Vision

In computer vision, adaptive models can focus computational resources on intricate parts of an image while spending less effort on simpler regions. This selective processing enhances both speed and accuracy in object detection and recognition tasks.

Edge Computing and IoT Devices

Resource constraints on edge devices make adaptive computation essential. Machine learning models that adjust their complexity dynamically can deliver smart functionalities like voice assistants or real-time analytics without draining battery life or requiring cloud processing.

Challenges and Considerations in Adaptive Deep Learning

While adaptive computation offers many benefits, it also introduces complexities:

****Training Difficulty:**** Designing and training models that can effectively decide when and how to adapt computation is non-trivial and often requires sophisticated algorithms.

****Model Interpretability:**** Adaptive behaviors can complicate understanding how decisions are made within the network.

****Hardware Compatibility:**** Not all hardware architectures support dynamic computation efficiently, potentially limiting deployment options.

****Balancing Accuracy and Efficiency:**** Finding the right trade-off remains a nuanced problem and depends heavily on the application.

Tips for Implementing Adaptive Computation in Your Models

If you're interested in integrating adaptive computation techniques into your machine

learning projects, consider these pointers:

Start Small: Experiment with early exiting in simple models before moving to more

1.

complex dynamic architectures.

Use Confidence Metrics: Implement reliable measures of prediction confidence to

2.

guide adaptive decisions.

Leverage Transfer Learning: Use pretrained models as a base to reduce training

3.

time when adding adaptive components.

Monitor Resource Usage: Track computational savings and energy consumption

4.

to evaluate the benefits of adaptation.

Explore Open-Source Tools: Frameworks like PyTorch and TensorFlow offer

5.

flexible APIs to build dynamic neural networks.

The Future Landscape of Deep Learning Adaptive Computation and Machine Learning

As AI continues to permeate everyday life, the demand for smarter, faster, and more efficient systems will only grow. Deep learning adaptive computation and machine learning are poised to play a central role in shaping this future. Researchers are actively exploring novel algorithms that combine adaptive computation with other cutting-edge developments like self-supervised learning and neural architecture search.

Moreover, the ongoing advancements in specialized hardware such as AI accelerators and neuromorphic chips will enhance the practical deployment of adaptive models, making them ubiquitous across industries ranging from healthcare to autonomous systems.

The journey toward truly intelligent and resource-aware machine learning models is well underway, and adaptive computation stands as a cornerstone of this transformation.

Whether you're a researcher, developer, or enthusiast, understanding and embracing these concepts will be key to harnessing the full potential of AI technologies.

Question

Answer

What is adaptive

computation in deep

learning?

Adaptive computation in deep learning refers to methods

where the computational resources or the structure of the

model dynamically adjust based on the input data or the

task complexity, improving efficiency and performance.

How does adaptive
computation improve
machine learning models?

Adaptive computation allows machine learning models to
allocate more resources to complex inputs while using
fewer resources for simpler ones, leading to faster
inference times and reduced energy consumption without
sacrificing accuracy.

What are some common
techniques used in adaptive
computation for deep
learning?

Common techniques include dynamic neural networks,
early exiting, adaptive depth and width networks,
conditional computation, and neural architecture search
that tailor the model's complexity to the input.

How is machine learning
integrated with adaptive
computation frameworks?

Machine learning models can be designed to learn policies

or gating mechanisms that decide which parts of the network to activate or skip during inference, enabling adaptive computation strategies.

What are the challenges in implementing adaptive computation in deep learning?

Challenges include designing effective gating mechanisms, maintaining model accuracy while reducing computation, handling hardware constraints, and ensuring training stability for dynamic architectures.

Can adaptive computation techniques help in deploying deep learning models on edge devices?

Yes, adaptive computation can significantly reduce the computational load and energy usage of deep learning models, making them more suitable for deployment on resource-constrained edge devices.

What role does reinforcement learning play

in adaptive computation for

machine learning?

Reinforcement learning can be used to train controllers or

policies that decide how to adaptively allocate

computation resources in neural networks, optimizing

performance and efficiency based on feedback.

Deep Learning Adaptive Computation and Machine Learning: Navigating the Future of

Intelligent Systems

deep learning adaptive computation and machine learning have become pivotal

concepts in the evolving landscape of artificial intelligence (AI). As the demand for more

efficient, scalable, and intelligent systems grows, the intersection of deep learning and

adaptive computation methods is reshaping how machines learn and perform complex

tasks. This article delves into the mechanics, applications, and implications of integrating

adaptive computation with deep learning, offering a comprehensive review tailored for

professionals and enthusiasts alike.

Understanding Deep Learning and Adaptive Computation

At its core, deep learning is a subset of machine learning that employs multi-layered

neural networks to model complex patterns in data. Traditional deep learning models

often rely on fixed computational graphs and static architectures, which, while powerful,

can be resource-intensive and inflexible when facing dynamic or evolving data

environments.

Adaptive computation introduces a paradigm shift by enabling models to modify their computational pathways based on input complexity or contextual demands. This means that instead of processing every input with the same amount of computation, the system can allocate resources more efficiently, potentially reducing latency and energy consumption without sacrificing accuracy.

The Synergy Between Deep Learning and Adaptive Computation

Integrating adaptive computation into deep learning architectures allows for dynamic adjustment of model depth, width, or processing time. For example, techniques such as adaptive neural networks, conditional computation, and early exit strategies enable models to decide in real-time how much computation is necessary for a given task.

This synergy addresses several challenges inherent in deep learning:

Computational Efficiency: By adaptively scaling the computation, models can

1.
operate faster on simpler inputs and devote more resources to complex ones.

Energy Consumption: Adaptive methods reduce unnecessary computations,

2.
which is critical for deploying AI on edge devices with limited power.

Scalability: Systems can handle a broader range of applications by dynamically

3.
adjusting their processing complexity.

Key Techniques in Adaptive Deep Learning Computation

Several innovative approaches exemplify how adaptive computation is being embedded within deep learning frameworks:

1. Dynamic Neural Networks

Dynamic neural networks adjust their structure during inference. Examples include models that selectively activate neurons or layers depending on the input. This selective activation can be realized through gating mechanisms or attention modules that prioritize relevant features.

2. Early Exit Networks

Early exit architectures allow models to halt computation at intermediate layers when sufficient confidence in the prediction is achieved. This approach reduces latency and computational load, particularly useful in real-time or resource-constrained environments.

3. Neural Architecture Search (NAS) with Adaptivity

NAS automates the design of neural network architectures, and when combined with adaptive computation principles, it can discover architectures that inherently support variable computational pathways tailored for different input complexities.

Applications Driving Innovation in AI

The practical implications of deep learning adaptive computation and machine learning are far-reaching across various sectors:

Edge Computing and IoT Devices

Deploying intelligent models on devices with limited hardware capabilities demands efficient computation. Adaptive deep learning models enable real-time analytics and decision-making on smartphones, sensors, and other IoT devices without constant cloud connectivity.

Natural Language Processing (NLP)

Adaptive computation enhances NLP models by allowing variable-length processing for sentences of differing complexity. For instance, transformer-based architectures can dynamically adjust attention mechanisms or layer utilization, optimizing both speed and accuracy.

Computer Vision

In image recognition and video analysis, adaptive models can focus computational effort on regions of interest, improving performance in tasks such as object detection and scene understanding while conserving resources.

Comparative Overview: Static vs. Adaptive Computation in Deep Learning

Feature	Static Computation	Adaptive Computation
Computational Load	Fixed for all inputs	Variable based on input complexity

| Energy Efficiency | Often higher due to redundant processing | Improved through selective computation |

| Latency | Consistent but potentially slow | Reduced for simpler inputs |

| Model Complexity | Simpler to design and deploy | More complex due to dynamic paths |

| Flexibility | Limited | High, suitable for diverse scenarios |

This comparison highlights why adaptive computation is garnering increased attention, particularly as AI systems become more integrated into everyday technology.

Challenges and Considerations in Implementing Adaptive Computation

While promising, integrating adaptive computation into deep learning models presents several challenges:

Training Complexity: Dynamic architectures require sophisticated training

1.
algorithms to ensure stability and convergence.

Hardware Compatibility: Not all hardware accelerators efficiently support

2.
dynamic computation graphs, potentially limiting deployment.

Predictability and Debugging: Variable computation paths complicate model

3.
interpretability and error analysis.

Addressing these issues requires ongoing research into optimization techniques, hardware-software co-design, and explainable AI methods.

Emerging Trends and Future Directions

The field is witnessing several advancements, including:

Hybrid Models: Combining fixed and adaptive components to balance reliability

1.

and efficiency.

Meta-Learning: Leveraging adaptive computation to enable models that learn how

2.

to learn more effectively.

Energy-Aware AI: Designing models that explicitly optimize for power

3.

consumption alongside accuracy.

These trends suggest that deep learning adaptive computation and machine learning will continue to evolve, driving the next generation of intelligent systems.

The integration of adaptive computation into deep learning frameworks represents a significant leap toward more efficient, flexible, and scalable AI. As research progresses and deployment challenges are addressed, these technologies will increasingly underpin applications across industries, transforming how machines interpret, learn from, and interact with data.

deep learning, adaptive computation, machine learning, neural networks, reinforcement learning, deep reinforcement learning, adaptive algorithms, artificial intelligence, supervised learning, unsupervised learning

Read the full article online: [deep learning adaptive computation and](#)