

RB_FINPRO_21: Research Project

Pricing & Cost Efficiency of Deep Learning as a Service

Not confidential

John Rateb

Student ID: 19-621-861

Supervisor Name: Prof. Dr. Ivo Blohm

Date of Submission: 2 July 2021

Word Count 17,121

Declaration of Authorship

RB_FINPRO_21 - Final Project (Research/Business Project)

Student Information:

	Student's First & Last Name	Matriculation #
Student #1	John Rateb	19-621-861
Student #2 (if working in group of 2)		
Student #3 (if working in group of 3)		
Student #4 (if working in group of 4)		

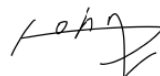
Word Count (excl. cover page, acknowledgements, reference list, appendices): _____ 17,121 _____

Please read and tick the respective boxes to confirm the following:

• I/We confirm that I/we have familiarised myself/ourselves with the project guidelines.	✓
• I/We confirm that I/we have familiarised myself/ourselves with and adhere to the rules and regulations as stipulated in the Student Handbook; in particular sections 6.0 - Code of Conduct, 7.0 - Academic Standards, and 8.0 - Academic Regulations.	✓
• I/We confirm that the project submitted is my/our own unaided work except where otherwise indicated.	✓
• I/We have not copied inappropriately from the work of others.	✓
• I/We confirm that my/our work as submitted was neither partially nor fully submitted to another institution, programme, or course.	✓
• I/We confirm that I/we have received and reviewed the APA referencing guidelines.	✓
• I/We confirm that all information, data, graphs, charts, or anything else which was retrieved from any external sources are referenced and quoted appropriately in line with the APA standards of referencing.	✓
• In the report, I/we have acknowledged any assistance which I/we have received, as appropriate.	✓
• I/We did not use any other person/organisation to have the work prepared for me/us, either partially or fully.	✓
• I/We am/are aware that the project can be examined for the use of unauthorised aid and that it will be submitted to turnitin.com to check for plagiarism and inappropriate referencing. Any inappropriate use of external sources will be penalised in line with the MBA's regulations. For the comparison of my/our work with existing sources, I/we agree that it will be entered into a database where it will also remain after examination to enable comparison with future theses submitted.	✓
• Where the project was completed by more than one student: We confirm that all group members contributed significantly to the outcome of the research and the generation of the project report.	N/A

Date: _ 2 July 2021 _____

Signature(s):



Acknowledgments

I would like to thank Prof. Dr. Ivo Blohm for his continued support and guidance throughout this research. I am also indebted to the MBA faculty and staff who opened my eyes to what is possible and to make better sense of the complex business reality we live in.

John Rateb

Abstract

This research studies the relationship between cloud service provider (CSP) pricing strategies and the cost efficiency of training deep learning models. Secondary research is first used to identify commonly used CSPs by data scientists. The two CSPs are then selected to represent an incumbent and a CSP focused on deep learning services. Pricing & value strategy of each CSP is then analyzed, and pricing models used to offer deep learning services are identified. In the second step, two deep learning case studies are defined in the domains of computer vision and natural language processing. Experiments are conducted to observe neural network training performance and Cost Efficiency of GPU instances offered by the selected CSPs for both case studies. Finally, the research contributes an experimental procedure for data scientists to elect the most cost efficient GPU instance for any deep learning use case.

Table of Contents

1. Introduction.....	3
2. Literature Review	6
2.1. Value-based pricing	6
2.2. Pricing of cloud services.....	7
2.3. Deep learning.....	8
2.3.1. Rise of deep learning	8
2.3.2. Cost and efficiency of deep learning	10
3. Research Methodology/Design.....	11
3.1. Pricing models of CSP deep learning services	12
3.1.1. Taxonomy of pricing strategies.....	12
3.1.2. Overview of cloud service providers (CSPs)	14
3.1.3. Deep learning scope of CSP services.....	16
3.1.4. Choices of CSPs for pricing model analysis	17
3.2. Performance and Cost Efficiency of CSP GPU instances.....	18
3.2.1. Metrics and data.....	18
3.2.2. Case study design	20
3.2.3. Performance measurement.....	24
3.3. Procedure design for cost-efficient instance choice	27
4. Research Findings.....	28
4.1. Pricing model analysis.....	28
4.1.1. Azure pricing models.....	28
4.1.2. Paperspace pricing models.....	32
4.1.3. Insights from comparative analysis	33
4.2. Performance and Cost Efficiency analysis	34
4.2.1. Azure performance results	34
4.2.2. Paperspace performance results	35
4.2.3. Cost Efficiency.....	36
4.2.4. Discussion	37
4.3. Procedure for cost-optimal instance choice	38
5. Conclusion	40
6. Recommendations	41
References	43
Bibliography	49
Appendices.....	56

List of Illustrations

Figure 1. Peer-reviewed AI publications, 2000–2019.	8
Figure 2. Private investment in funded AI companies, 2015–2020.	9
Figure 3. Graphical illustration of a deep neural network composed of five layers.	9
Figure 4. Computing power needed to train modern deep learning architectures.	10
Figure 5. A sample convolutional neural network (CNN) with seven layers.	21
Figure 6. Parallel view of the human visual cortex and a CNN.	21
Figure 7. A visualization of recurrent neural networks (RNNs).	22
Figure 8. Pricing strategy classification of Azure and Paperspace deep learning services.	33
Figure 9. Relative performance of multi-GPU to single GPU instances.	35
Figure 10. Relative performance of Paperspace instances to P4000.	36
Figure 11. Cost efficiency of Azure and Paperspace GPU instances.	37
Figure 12. Experimental procedure for instance choice.	38

List of Tables

Table 1. Taxonomy of pricing models.	14
Table 2. Usage rate of CSP deep learning services by data scientists.	17
Table 3. Neural architecture, dataset, and hyperparameters used for the case studies.	24
Table 4. GPU instance characteristics and their hourly pricing.	25
Table 5. Overview of the case study experimental design variables.	26
Table 6. Overview of deep learning services by Azure and Paperspace.	28
Table 7. Summary of the strategies employed by Azure deep learning services.	31
Table 8. Summary of the strategies employed by Paperspace deep learning services.	33
Table 9. Time per Epoch for CNN and RNN cases on Azure instances.	34
Table 10. Time per Epoch for CNN and RNN cases on Paperspace instances.	35

1. Introduction

Artificial Intelligence (AI) has been heralded as one of the biggest facets of the Industry 4.0 revolution (IBM, 2018). It has captured the imagination of film makers, scientists, and investors alike. Deep learning is the branch of AI that touches our lives the most by powering search engines, navigating warehouses, and reducing fraud. Deep learning has quickly moved from the domain of computer science laboratories to servicing businesses around the globe, from startups to Fortune 500 companies.

This accelerated uptake was facilitated by an ecosystem of data scientists, consultants, and industry experts. It was enabled in large by the parallel rise of cloud service providers (CSPs). The cloud has enabled businesses to adopt deep learning without building expensive high performance computing data centers. Cloud players quickly offered myriad deep learning services built atop an array of hardware to enable all sorts of use cases. This made the cloud the only practical choice of any business wanting to implement deep learning without having to spend significant upfront capital investment.

Large cloud incumbents benefit from extensive business experience enabling them to offer deep learning services using a suite of pricing strategies and models to suit different customer needs. The ultimate goal of these pricing models is to maximize the profit of the cloud provider, and they are in many cases difficult to directly compare between cloud providers or between individual services within one cloud provider. Thus, data scientists building deep learning models face the challenge of optimally choosing the right type and size of computing services for their use case. As deep learning models usually rely on massive amounts of data and expensive hardware, they are increasingly becoming a cost driver (The Economist, 2020a). This cost challenge is here to stay —IDC (2020) forecasts that worldwide spending on AI will more than double by the year 2024.

Thus, a suboptimal cloud service choice could unnecessarily add substantial cost to companies that heavily utilize deep learning. This is significant since a large number of startups are entering the deep learning space, and they attract billions of dollars of private investment every year (Stanford University HAI, 2021). The cost challenge is compounded by the lack of commonplace cost prediction tools that can reliably take the use case and hardware specifications into account. The continuous experimentation nature of building deep learning models adds further complexity to the challenge. A deep learning model may demand another optimal service after a data scientist tweaks its parameters or expands the underlying data as is common to achieve better model performance. Also, optimizing only for low cost is not ideal, as businesses expect a certain level of performance when building deep learning models, not wanting to wait for weeks of computation to see how predictive the model is. This study focuses on cost efficiency, a fairer way of comparing the cost of services with variable hardware and performance.

The cost for utilizing cloud services is a function of the pricing and value delivery strategies of CSPs. CSPs offer a myriad of services to cover the needs of customers of different sizes and skill levels. Infrastructure as a Service allows customers to pay for utilized hardware resources by the second. Platform as a Service gives access to sector-specific software applications allowing customers to avoid managing infrastructure. On the other end of the value spectrum, the Function as a Service offer gives less technical customers a way to implement intelligent services like machine translation, billing customers by functional units such as number of characters translated and alleviating them from needing to invest in the skills or time required to build a translation engine themselves.

Linking pricing strategies and value propositions used by CSPs to cost efficiency requires studying the specific pricing models CSPs use for maximizing their profit and gaining market share. Wu et al. (2019) extensively studied and classified the pricing models of CSPs, and concluded that CSPs should continue shifting their pricing strategies from cost-based and market-based to value-based pricing. Convincing customers to be loyal to CSPs requires value co-creation according to Reen et al. (2017), sharing both risk and reward.

As large incumbent CSPs serve many use cases beyond AI, new disruptive CSPs that are focused on deep learning offer innovative services to deliver more value for data scientists in an attempt to gain market share. These innovations call for value-driven pricing models to convince data scientists to use the new specialized CSPs and to differentiate from the incumbents. Large CSPs then are encouraged to also adopt the shift to value models to protect their market share.

Data scientists crave automation in nearly all components of their daily work, according to Wang et al. (2021). One notable exception is making trade-off decisions, which data scientists still do not prefer to automate. When training deep neural networks, choices must be made regarding which CSP and which specific service to use. While data scientists may not want to automate these decisions, they undoubtedly prefer to have metrics to help them make these choices fairly under the cost and time constraints of their work environment.

The business problem motivating this research is two-fold. First, it is challenging for data scientists to optimally choose between CSPs or services for their business use case due to the variety of pricing models and service specifications. Second, the pre-set bundling of computing resources in the cloud can lead to underutilization – and higher cost – depending on the nature of the deep learning model.

Drawing from these challenges, this research aims to answer the following question: what is the impact of the pricing strategies employed by CSPs on the performance and cost efficiency of deep learning business projects?

To answer that research question, the research has three objectives:

1. Identify the pricing strategies and models used by CSPs offering deep learning services. To do that analysis, first we will survey available CSPs. Then CSPs are classified into general purpose CSPs which offer a wide range of cloud services, and special purpose CSPs which focus on deep learning. Two CSPs will be selected—one representing each group. General purpose CSPs are expected to offer a variety of pricing strategies, while special purpose CSPs may offer more innovative value and pricing models to attract data scientists and gain market share.
2. Investigate how does the pricing and features of deep learning cloud computing services impact the performance and cost efficiency of commonly used neural network architectures. To answer this objective, two case studies are introduced covering deep learning applications commonly used in business—image and text classification. The neural network training performance of each case study will be tested on the compute infrastructure (virtual cloud computer) of the two CSPs selected earlier. This research conjectures that different architectures could demand a different cost efficient service.
3. Design a procedure to help support data scientists systematically elect the optimal cloud compute service for their deep learning workload given an objective of cost efficiency. The procedure should be sufficiently quick and relatively cheap to encourage its users to repeat it whenever they face a compute service choice. The experimental nature of

training deep neural networks suggests that changing assumptions about network architecture or the data may require a new service choice to maintain cost efficiency.

2. Literature Review

2.1. Value-based pricing

Customer value creation has become synonymous with competitive advantage. Buts & Goodstein (1996) refer to it as the “emotional bond established between a customer and a producer after the customer has used a salient product or service produced by that supplier and found the product to provide an added value”. They argued that net customer value, defined as the difference between the benefit of the product and the customer sacrifice, is the driver of the bond between a customer and a producer.

Value calculation can be challenging as it relies on customer perception and is considered the sum of extrinsic and intrinsic values gained by the customer. Intrinsic values correspond to core product characteristics that cannot be easily modified such as materials or quantity, while extrinsic values relate to qualitative product attributes such as service quality or brand name (Richardson et al., 1994). Price could be considered an extrinsic value, which is why Shaw (1991) argued that pricing can be considered as a competitive positioning tool, and Tsao et al. (2006) discuss how pricing can be used to signal a brand as a market leader under conditions of information asymmetry.

Priem (2007) provides a more modern definition of value as the benefit experienced by customers or the maximum price they are willing to pay based on their perception of a product or service – also known as “reservation price” (Voelckner, 2006). This view is considered a shift from the “consumer surplus” view (Hinterhuber, 2004). The “consumer surplus” view poses a challenge of defining value dependent on price, which limits the use of value calculation in setting the right price (Marn & Rosiello, 1992).

Value-based pricing has proven to increase profitability. Hinterhuber (2008) provides a survey of early evidence. Toni et al. (2017) found that Brazilian companies in the metal industry using value-based pricing enabled better profitability. Liozu (2017) surveyed 144 organizations from different industries around the world and showed the intrinsic and extrinsic product differentiators from competition are predictive of business unit profitability.

Profitable price setting based on the value perceived by customers – value-based pricing – requires balancing three dimensions: cost for the firm (to ensure profitability), economic value for customers (the cost of using an alternative by customer segments), and competitive situation (product and customer segment differentiation as well as competitors’ reaction to price changes). Firms can extract more profit from several angles, yet pricing is considered the most important profitability driver as compared to reducing costs, or increasing sales volumes (Marn & Rosiello, 1992).

A modern form of value-based pricing that predicts success by service firms is value co-creation, or solution selling, where service providers price their services in part by the outcomes gained by their customers. This approach builds longer and more profitable relationships by reducing information asymmetry between service providers and customers and sharing both gains and risks (Sharma & Iyer, 2011).

Despite the benefits of value-based pricing, adoption has been slow due to implementation challenges. Liozu et al. (2012) found that 40% of executives in US industrial companies do not conceptually understand value-based pricing or how to apply it, resulting in a lack of adoption. Hinterhuber (2008) identified additional barriers including difficulty calculating and communicating value, resistance from the sales force, and lack of senior management support in companies around the world.

2.2. Pricing of cloud services

Cloud computing offers a way for companies to move several components of their IT infrastructure to a third party Cloud Service Provider (CSP). This changes how companies invest in their own private infrastructure, and how big their IT teams have to be to build and maintain such infrastructure.

Cloud adoption is frequently claimed to be an important profitability lever for large companies. McKinsey (Forrest, 2021) estimates there is \$1 trillion USD dollars of available profitability for Fortune 500 companies by 2030. This value creation is driven by rejuvenation (via cost and risk reduction through digitization) and innovation (from access to new growth markets, faster product development, and elasticity to customer demand). Marston et al., (2011) analyze the many benefits cloud computing brings to companies. It reduces the investment required by new market entrants by not having to purchase hardware or recruit a large IT team, it provides flexibility by allowing companies to effortlessly scale up or down their computing resources according to the level of demand, and offers the latest technologies like deep learning and satellite communication (AWS, 2021) in an easy to consume way, thus accelerating innovation.

However, Klems et al. (2008) argue that there is no systematic way to calculate the customer value gained from cloud adoption. Also, Jäätmaa (2010) finds that cloud adoption does not markedly reduce the cost of IT services. CSPs have made significant gains in the process; the Economist reports that companies spent \$230 billion USD in 2019 on the cloud (The Economist, 2020b). Cloud adoption has become so commonplace that the latest trend according to the Economist is to endorse multi-cloud computing services giving customers the option to have the best attributes of each cloud with seamless application and data interoperability.

With the large scale adoption of the cloud and their variety of service offers, pricing of CSPs is becoming more important. Wu et al. (2019) surveys and classifies pricing models employed by CSPs based on value propositions, pricing strategies, and methods of value delivery. They conclude that over time, CSPs are moving from cost-based to value-based pricing strategies. Reen et al. (2017) make the case for a value-based pricing strategy and argue that IT service providers (like CSPs) can benefit from value co-creation with their customers, yet they must first undergo a significant internal shift to implement value-based pricing across all parts of the organization.

This co-creation is important because cloud customers must also undergo significant internal transformation. Ahokangas et al. (2014) highlight that this transformation can induce changes to the business model of cloud customers. Value co-creation can have negative consequences; it can introduce ambiguity and conflict between the roles of the CSP and the customer (Chowdhury et al., 2016). A recent McKinsey survey (Dertouzos et al., 2020) supports this, highlighting that 30% of the cloud budget by companies is wasted – particularly due to change management.

CSP pricing has attracted several researchers to analyze and propose profit-maximizing pricing models. This research followed the evolution of cloud service delivery models. Jin et al. (2014) propose a pricing scheme for Infrastructure as a Service (IaaS) to optimize profit for both CSPs and their customers. Platform as a Service (PaaS) models emerged as a way to abstract development and running of applications (Buyya et al., 2009). Software as a Service (SaaS) is now the largest cloud offering and licensing model (Saltan & Smolander, 2021). Most recently, Feature as a Service (FaaS) – or serverless computing – is gaining traction and offers substantial cost savings through innovative pricing models (Villamizar et al., 2017), and has room to grow for yet more applications (Lynn et al., 2017).

2.3. Deep learning

2.3.1. Rise of deep learning

AI is a field that is both heavily researched and invested in. The AI Index report (Stanford University HAI, 2021) shows that private investment in AI companies reached \$40 billion USD in 2020. AI related research papers accounted for 3.8% of all peer-reviewed publications in 2019 after a steep growth over the preceding five years according to the same AI Index report. Figures 1, and 2 show the proportion of AI publications and amount of private investment in AI companies, respectively.

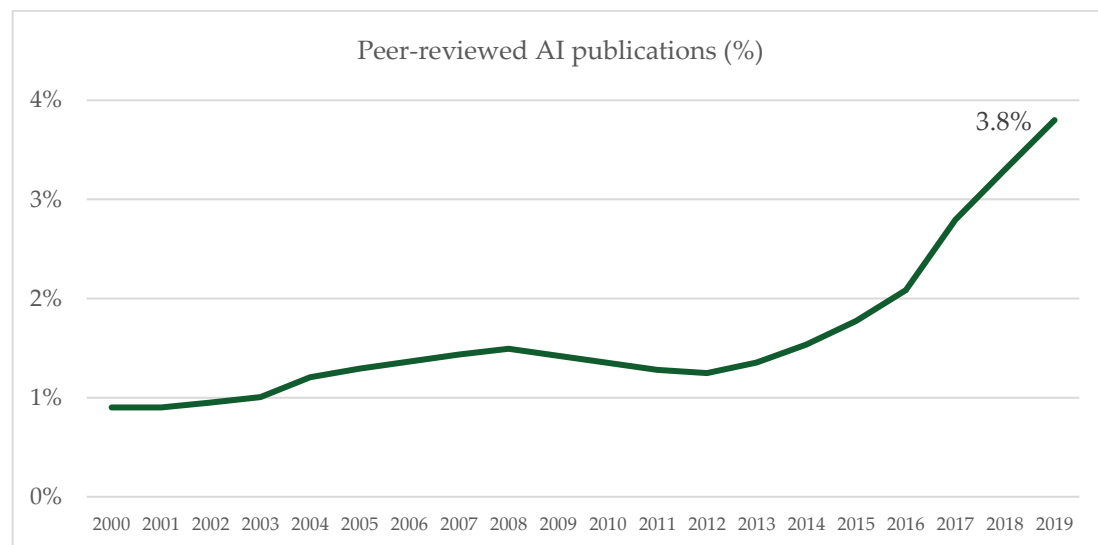


Figure 1. Peer-reviewed AI publications (% of total), 2000–2019. Source: 2021 AI Index Report.

From an industry point of view, AI has been adopted in several sectors. Weber & Schütte (2019) survey the adoption of AI across the value chain in retail, for example in assortment planning, recommender systems in e-commerce, and chatbots. This is supported by the large datasets readily available at retailers, yet they find some big retailers still lag behind in AI adoption. In agriculture, Barbedo (2018) provides a survey of the use of AI in identifying plant pathologies, but highlights that the lack of availability and sharing of data is slowing adoption. Ozbayoglu et al. (2020) survey myriad AI applications in finance, including algorithmic trading, fraud detection, crisis forecasting, derivative pricing, and financial text mining.

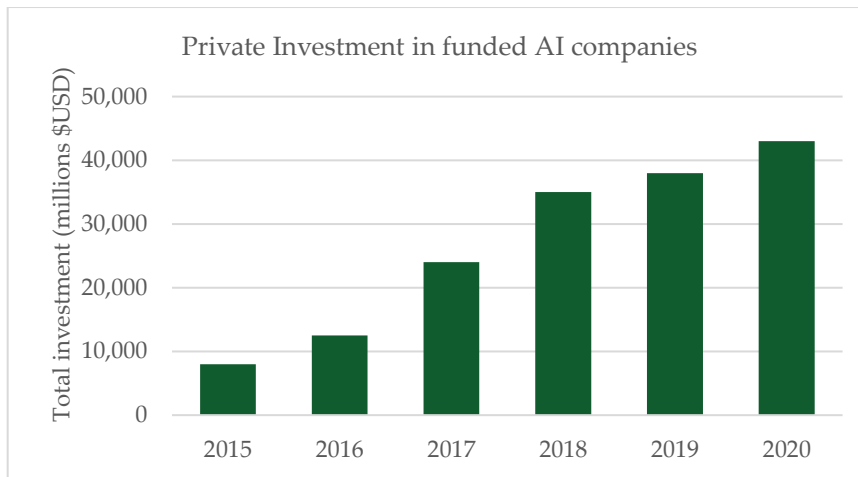


Figure 2. Private investment in funded AI companies, 2015–2020. Source: 2021 AI Index Report.

AI is comprised of a variety of fields, one of which is machine learning. Machine learning trains algorithms based on data, rather than needing to directly code the intelligence logic. Machine learning encompasses many algorithms that are designed to work on different types of problems and volumes of data. Deep learning is a subclass of machine learning designed to execute complex tasks through the use of neural networks (LeCun et al., 2015).

Neural networks are a class of machine learning algorithms designed after the structure of neurons in the brain, and they can learn from complex data representations such as images and text. Neural networks can be as deep or shallow as the data scientist desires. The foundations of neural networks were laid back in 1949 by Donald Hebb, yet the computational requirements for training deep networks were too advanced for the hardware available at the time (Wang & Raj, 2017). Deeper networks can capture ever more complex information but require a commensurate amount of data for training. The process of training deep neural networks is coined “deep learning”. Figure 3 illustrates a five-layer deep neural network (IBM, 2018).

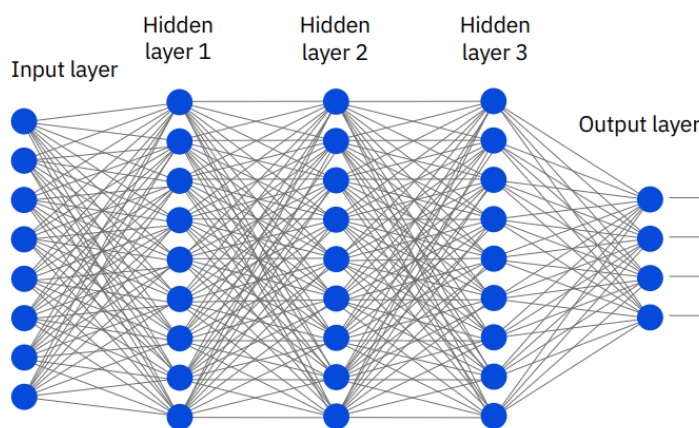


Figure 3. Graphical illustration of a deep neural network composed of five layers. Source: IBM (2018).

GPUs (Graphical Processing Units) are computer hardware that is designed to render graphics and are used extensively in gaming applications due to their ability to run several parallel matrix calculations. The same capabilities that make GPUs ideal for gaming also align with the mathematical models behind training neural networks. Nickolls & Dally (2010) describe how GPUs and hybrid CPU + GPU architectures can be used for massive parallel computing applications, which speed up the training process of deep learning models. Bergstra et al. (2011) introduced a software framework (Theano) that can train deep neural

networks more than four times faster on GPUs instead of CPUs, and ushered in the new era of deep learning. Quickly after, research into distributing deep learning training on multiple GPUs picked up pace. Having multiple GPUs allows neural networks to run parts of the training calculations on all available GPUs in parallel. This parallelization comes at an efficiency price. Cui et al. (2016) who created a software to enable neural network training on multiple GPUs showed a 81% parallelization efficiency when using 16 GPUs.

2.3.2. Cost and efficiency of deep learning

The rise of deep learning combined with the shortage and increased prices of GPUs, particularly due to their use in crypto coin mining (The Economist, 2021a), has pushed firms to use deep learning services on the cloud. Analysis by Amodei and Hernandez (2018) shows that modern deep learning network architectures require exponentially more computing power, doubling every three to four months in the modern era. Figure 4 illustrates the exponential increase of complexity in the modern era, while highlighting significant innovations in deep learning architectures. This accelerated increase in complexity pushes many AI startups to rent their GPU computing power from the cloud, with costs sometimes reaching 25% of their revenue according to The Economist (2020a).

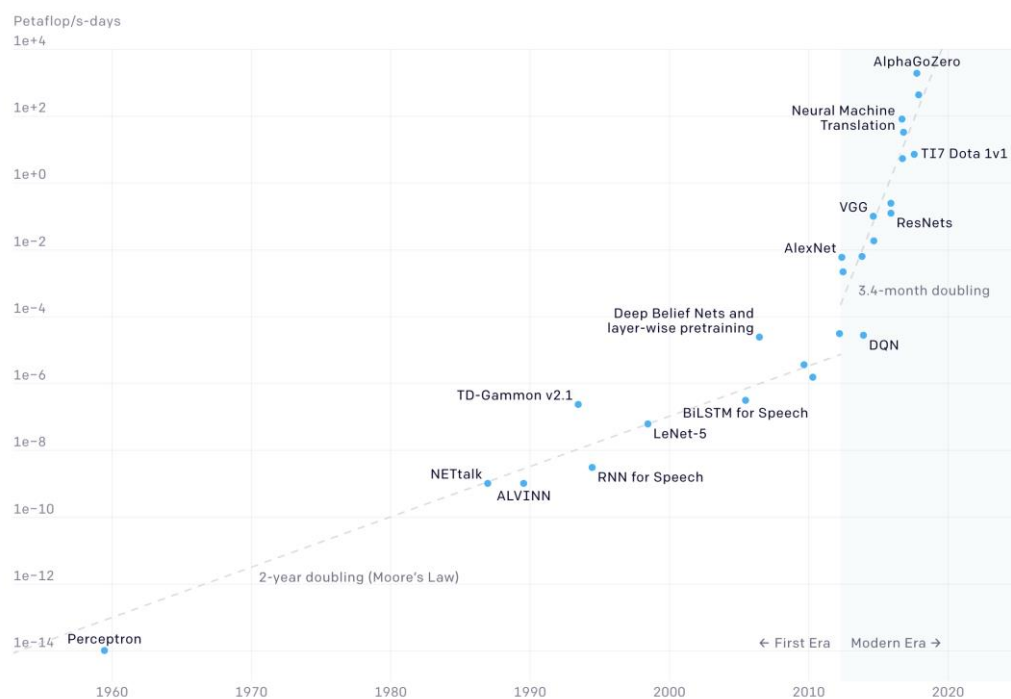


Figure 4. The exponential increase in computing power needed to train modern deep learning architectures in the modern era. Logarithmic Y-axis shows the amount of calculations required to train a network per training example. Source: (Amodei & Hernandez, 2018).

The rise in deep learning computational complexity has triggered a wave of research to quantitatively express complexity and study the efficiency of deep learning from three perspectives: energy efficiency, time to accuracy, and cost efficiency.

He et al. (2016) express computational complexity in billion FLOPs (multiply or add operations) when comparing neural networks for image recognition. They used FLOPs to justify proposing new lighter network architecture that can achieve the same accuracy as a more complex model. Since calculation speed can vary depending on the GPU, Ma et al.

(2018) suggest using a more direct metric – like speed to allow practical use. This research also uses a time-based metric to assess the training performance.

From the perspective of energy efficiency, Strubell et al. (2019) study the energy consumption and environmental footprint of training natural language models. Li et al. (2016) compare the energy demand of CPUs versus GPUs when training neural networks for image and video applications under different hardware and training conditions. On the other side, deep learning was used to improve the energy efficiency of cloud data centers (Lazic et al., 2018) and hybrid electric vehicles (Qi et al., 2017).

Time to accuracy is another important perspective for analyzing deep learning models since accuracy refers how predictive a model is when fed previously unseen data during its training. Kovalev et al. (2016) studies how different software frameworks perform in terms of prediction accuracy and the time they need to reach it under different conditions. Coleman et al. (2017) introduced DAWNBench, a deep learning benchmark and competition where researchers around the world can submit new time to accuracy and training cost submissions for specific datasets. The results vary widely; for example, training ImageNet¹ – a popular standard image classification dataset – to 93% accuracy has ranged from more than 13 days in 2017 to less than three minutes in 2020, costing anywhere from \$7.4 to \$2323.4 USD, with each submission using different neural architecture, software, hardware, and CSP. MLPerf training benchmark (Mattson et al., 2019) specifies different quality metrics beyond accuracy to cover more deep learning use cases than DAWNBench, such as image segmentation, language translation, and recommender systems. MLPerf provides less degrees of freedom in benchmarking by fixing the neural architecture and limiting the modifiable network hyperparameters used to tune the network behavior.

Research on cost efficiency of training neural networks is limited and insufficiently cited relative to the previous two perspectives. One recent paper by Malta et al. (2019) compares the runtime and cost of GPU instances in one CSP across a range of hyperparameters, finding that more performant and expensive GPU hardware could bring cost benefits. Yeung et al. (2020) and Justus et al. (2018) attempt to model the training time given GPU specifications and type of neural architecture layers as parameters in an effort to predict cost. However their models do not extend to recent GPU hardware and miss recent neural network layer types, limiting their predictive power in modern deep learning workloads.

This research was primarily inspired by the mixture of (1) wide cost gap observed by benchmarks, (2) increasing cost of training neural networks in business, (3) limited CSP GPU cost benchmarking, and (4) poor predictability of deep learning workloads.

3. Research Methodology/Design

Real life deep learning applications exhibit a large variety in their data, scale, and underlying algorithms. This research employs a case study strategy to illustrate how the pricing and GPU instance characteristics of selected CSPs impact the cost efficiency of training two deep learning use cases in the computer vision and natural language domains.

Following the suggestion of Yin (1994), a case study strategy is advisable to account for the interdisciplinary complexity of pricing and deep learning applications, particularly since the topic of pricing of deep learning services is not well studied. This case study approach was

¹ ImageNet Training. <https://dawn.cs.stanford.edu/benchmark/ImageNet/train.html>

used by Reen (2017) to research the application of value-based pricing in IT firms, and by Hinterhuber (2004) to illustrate pricing decision-making in new product launches.

Case study strategy is also widely utilized in deep learning research due to the unique nature of each use case. Carniero (2018) used computer vision case studies to study the performance of Google's notebook service, and Kim (2020) used a case study approach to look at the use of deep learning in predicting financial risk-taking. Kovalev et al. (2016) use a digit classification case study to compare the performance of different network depths and software frameworks. Malta et al. (2019) also use two computer vision case studies to illustrate performance gaps between GPUs.

A three-step approach is used to study the relationship between CSP pricing models and cost efficiency of deep learning applications. First, secondary research reveals commonly used CSPs by data scientists as well as the value drivers and pricing models employed by a sample of two CSPs. The two CSPs are chosen to represent a large incumbent and a specialized new entrant. Second, two deep learning case studies are used to observe the training performance and cost efficiency of single and multi GPU instances offered by the selected CSPs under different neural architecture conditions. Third, the research concludes by contributing a procedure for data scientists to elect the most cost efficient GPU instance for their particular deep learning use case to alleviate the challenge of performance unpredictability when training neural networks.

3.1. Pricing models of CSP deep learning services

This section will discuss the pricing and value framework used to analyze the pricing models of CSP deep learning services. It starts with describing the pricing model taxonomy and goes on to classify CSPs into general and special purpose. Finally, two CSPs are chosen to further analyze their pricing models.

3.1.1. Taxonomy of pricing strategies

3.1.1.1. Framework for pricing classification

Wu et al. (2019) conducted a thorough analysis of cloud pricing strategies. They observed three main strategies: value-based pricing, market-based pricing, and cost-based pricing. Each pricing strategy encompasses several possible models depending on how customer value is created. A pricing model is descriptive of the profit maximization formula the CSP uses to calculate its prices. A total of 60 unique pricing models that can be used in tandem in CSP product offers were identified and grouped. Table 1 shows the taxonomy and a selection of the pricing models.

A pricing strategy is the top level of abstraction regarding how a CSP generates profit while co-creating value for its customers. These are driven by market supply and demand dynamics.

- Value-based pricing is driven by customer demand, and is considered a subjective view of pricing as it is driven by customer perception of the value (Baur et al., 2014) delivered by the cloud product. It embeds both intrinsic (e.g. the instance hardware), and extrinsic (e.g. service level agreements and protection from network attacks) value drivers (Wu et al., 2018). Example models include per user, outcome-based, and feature-based pricing.

- Cost-based pricing is driven by resource supply, and is an objective way to calculate pricing, easy to explain, and used by a large majority of companies (Raju & Zhang, 2010). It is directly derived from the cost drivers and helps decision makers decide on pricing new products with a target profitability in mind, and an unknown expected demand (Noble & Gruca, 1999). Example models include cost-plus, and FTE-based pricing.
- Market-based pricing is driven by the balance of supply and demand. It is commonly used in “spot instances” – instances not in use at CSPs that are auctioned off at steep discount with the caveat that they can be shut off any point (Ben-Yehuda et al., 2013). It is calculated based on customers’ price sensitivity and competitive intensity. Example models include freemium, promotional, and English auctions pricing.

Within each pricing strategy, pricing models can be further grouped by the type of value into three categories of models. Each category represents the value proposition of the CSP service.

1. “Good to Have” value, where a CSP works on providing the most essential services to create a good customer experience, thus accelerating cloud adoption, and cementing the CSP market share. For example, it can be applied by reducing the barrier of trying cloud services by offering free service tiers, by making customer cost predictable through pricing models that are driven by customer resource utilization, or by offering faster customer support response time at a higher service price.
2. “Good to Do” value, which creates a “win-win” situation by reinforcing the customer’s value proposition. For example, pricing can be set based on a customer outcome such as guaranteed performance level or user traffic volume, or by offering the pay-per-hour expert resources to support cloud onboarding and building customized architecture for the customer.
3. “Good to Be” value, which is not only dependent on customer value, but is designed to stimulate customer demand. This would then encourage customers to shift workload to the cloud as it claims more customer revenue. This strategy includes models such as peak and off-peak pricing which provides discounts when customers are not gaining high value. Another example is the discriminatory pricing model which differentiates pricing between customer segments depending on the generated value or willingness to pay, a popular model with student licenses.

		Cloud Pricing Strategies		
		Value-Based Pricing (Demand driven)	Market-Based Pricing (Market interaction)	Cost-Based Pricing (Supply driven)
Value Proposition	Good to Have	Service based models, e.g. on-demand and tier-based pricing.	Free upfront and pay later, e.g. freemium and razor-and-blades pricing.	Expenditure based, e.g. cost-plus and percentage pricing.
	Good to Do	Experience based, e.g. outcome and customer care-based pricing.	Auction & Online based, e.g. spot and auction pricing.	Resource based, e.g. volume and FTE-based pricing.
	Good to Be	Customer value based, e.g. psychological and feature-based pricing.	Retail based, e.g. product mix and discriminatory pricing.	Utility based, e.g. dynamic and reserved-based pricing.

Table 1. Taxonomy of pricing models adapted from Wu et al. (2019). Each of the three pricing strategies encompasses three categories of pricing models based on what type of value is delivered to the customer. “Good to have” pricing models are designed to consolidate the CSP market share by redefining service value to customers. “Good to do” models focus on creating value on the customer side, increasing their willingness to pay. “Good to be” models support more demand on the customer side, encouraging them to move workload to the cloud.

3.1.1.2. Classification of CSP pricing models

Secondary research is used to collect the list of deep learning services from two CSPs, one incumbent offering a full selection of cloud services, and the other a specialist focused on deep learning. After listing deep learning services, the features, deliverables, and pricing data are collected for each.

A three-step approach is used for pricing model classification. (1) Each service, even under the same CSP, could have a different value proposition. Analysis of service features and deliverables yields information to identify the value proposition of each service based on the type of value delivered. (2) The language used in CSP pricing tables and service deliverables help identify which specific pricing models are used, noting that each service could employ multiple models to cater to different customer segments. (3) The last step is to deduce which pricing strategy or strategies is exercised by the CSP based on the identified pricing models using Wu et al. (2019) framework.

3.1.2. Overview of CSPs

CSPs operate in a strong competitive market. This leads the large providers to continuously offer similar services as their competitors to avoid customer attrition and grow their market share. Market competitiveness gradually forces competing CSPs to offer their commodity services, such as instances, at similar prices particularly when the instance configurations are similar. To continue maximizing profit, CSPs then seek to add differentiating value to their services justifying the various pricing models previously described.

Furthermore, specialized CSPs have emerged in the last decade focusing on the machine learning space offering primarily GPU instances, which are specialized software services with the hope of differentiating themselves from large CSPs. Specialized CSPs operate in a

narrower market than large CSPs and focus on delivering value to data scientists, thus they may employ different pricing strategies and models.

Several firms exist that offer deep learning software services without the hardware infrastructure, instead deploying their software on the hardware infrastructure of a general purpose CSP. These firms offering only software are excluded from this study scope.

To study the pricing models of deep learning services, this research identifies two classes of CSPs, *general purpose CSPs* which offer a wide scope of infrastructure services (including deep learning) and *special purpose CSPs* which offer specialized hardware and software services targeted at deep learning use cases.

Kaggle is one of the largest data science communities offering competitions and educational resources. Kaggle surveys thousands of data scientists each year about the “State of Data Science and Machine Learning”, including the usage² of both general purpose and special purpose (named “hosted notebook products”) CSPs, and openly provides the survey response data. While some general purpose CSPs (e.g. AWS) are popular in the data science community, Kaggle (2020) survey helps to provide a comprehensive list of both general and special purpose CSPs and is used as a secondary data source to report the usage of each CSP.

3.1.2.1. General purpose CSPs

Cloud computing emerged as a way to rent out excess computing resources in data centers at large companies. Virtualization technology enabled this revenue-generating activity through emulating isolated compute units, despite not being physically separated (Marston et al., 2011). This isolation is necessary to allow multiple cloud customers to securely use compute resources without interfering with one another. Another concept cloud computing brought is multitenancy, where one software application can be used to service multiple customers.

Virtualization innovations ushered new business models. In 2006, AWS became the Infrastructure as a Service (IaaS) cloud provider by offering its Simple Storage Service (S3) with on-demand usage driven pricing (AWS, 2016). Shortly after, they introduced their virtual machine rental cloud service – Elastic Compute Cloud (EC2). By 2008, Google followed AWS, and entered the cloud market with its Google App Engine offering hosting services on its data centers. Microsoft followed suit with its Azure cloud service in 2010 using similar pricing models as AWS and Google.

Maturation of virtualization software and the growth in demand for cloud computing prompted several other firms to enter the cloud market such as IBM, Oracle, and Rackspace. AWS, Google Cloud Platform (GCP), and Microsoft Azure are still the largest CSPs with a combined market share of 55% according to the cloud market share report by Canalys (Canalys, 2020).

3.1.2.2. Special purpose CSPs

Programming notebooks are an essential component in the modern data scientist toolset. Notebooks are web-based tools that can run offline on the data scientist machine or on the cloud. These tools were inspired by Mathematica notebooks, popular with mathematicians,

² Questions 10 and 25 of the 2020 survey indicate the usage of specialized CSPs in the survey, and large CSPs, respectively.

and physicists. Now, notebooks allow data scientists to combine their code, the analysis results, as well as associated documentation and graphics in one document. This makes notebooks a portable research tool and encourage research collaboration. Google saw the opportunity and was the first CSP to offer a specialized notebook environment built atop its GPU compute resources as a service, called Google Colab (Carneiro, 2018). Google Colab can be used for free (Google, 2021) as a way to acquire loyal users and get them familiar with Google cloud services.

A new generation of CSPs emerged that offers services targeted at data scientists with a focus on GPU instances and associated software. Notebooks are a common theme of these specialist CSPs. Pérez & Granger (2007) designed and launched Jupyter as a free notebook software in the hope of democratizing data science and it gained popularity in the data science community. Paperspace, the first specialized CSP, launched its Gradient notebook service based on Jupyter (Marcel, 2018) with a novel tiered subscription pricing model that provides gradual access to more powerful GPU instances. Other niche players followed suit with new pricing models such as Floydhub (Floydhub, 2021) and Jarvislabs (Jarvislabs, 2021). These special purpose CSPs provide a contrasting view to the large general purpose CSPs that enjoy multiple sources of profitability from their cloud services.

3.1.3. Deep learning scope of CSP services

Incumbent general purpose CSPs began to launch services targeted at data scientists on top of their commodity compute services. This was driven by the increasing adoption of deep learning by companies, combined with the exponential increase in compute resources needed to train neural networks, and the proliferation of competitor CSPs – both large and small. These services cover a wide spectrum of machine learning models, not just deep learning. Examples of these include SageMaker by AWS and Azure Cognitive Services by Microsoft. The features of these services include data science notebooks and management tools meant to reduce the administration burden on data scientists to train, monitor, and deploy machine learning models to live production environments.

Table 2 shows an overview of general and special purpose CSPs surveyed by Kaggle, the deep learning services they offer, and their usage rate by data scientists. Usage rate data are based on a 2020 survey except for Floydhub where 2019 data is used. Google Colab Notebooks have a particularly high usage rate since they are offered for free, albeit with limitations on what software can be installed and the neural network training duration.

While the scope of deep learning software services by CSPs varies widely, each CSP offers some version of GPU instances – each with bundled hardware configuration – for data scientists to run their computation on. AWS has the highest usage of instances (14.0%), followed by Google Cloud (11.4%), and Microsoft Azure (8.5%). The common availability of GPU instances provides a way of comparing cost efficiency between CSPs when training neural networks on their infrastructure.

CSP	Services	Usage rate
Amazon Web Services (AWS)	Amazon EC2, Accelerated Computing instances	14.0%
	Amazon Sagemaker Studio	2.5%
	Amazon EMR Notebooks	1.2%
Microsoft Azure	Azure VM instances	8.5%
	Azure Notebooks	4.3%
Google Cloud	Colab Notebooks	31.6%
	Google Cloud Platform (GCP)	11.4%
IBM Cloud	IBM Watson Studio	4.2%
	IBM Cloud Servers	2.2%
Oracle Cloud	Oracle GPU Virtual Machines	1.4%
Paperspace	Gradient Notebooks	0.9%
Floydhub ³	Floydhub Deep Learning Platform	0.5%

Table 2. Usage rate of CSP deep learning services by data scientists. Source: Kaggle (2020).

3.1.4. Choices of CSPs for pricing model analysis

In this step, two CSPs are chosen to investigate the pricing strategies and models employed when offering deep learning services. To observe differences between general and specialized purpose CSPs, one CSP is chosen from each group. The remainder of this research will focus on analyzing the pricing models and cost efficiency of GPU instances on Microsoft Azure (henceforth called Azure) and Paperspace.

While Azure is the third largest general purpose CSP, it is chosen for analysis for four reasons. (1) The author is familiar with its infrastructure and programming environment, aiding the experimentation process. (2) Azure has a large variety of GPU instances on offer, including modern GPUs and instance with multiple GPUs. (3) It offers a number of deep learning services, each delivering a unique value model. (4) It has data centers in Switzerland, making it particularly relevant to Swiss customers with strict regulatory requirements.

Azure GPU instances can be set up on different operating systems. Instance pricing varies depending on which operating system is being used due to license cost. Since operating system license pricing is beyond the scope of this research, all experiments use only Ubuntu Linux instances, being the operating system of choice for most deep learning workloads. Also, other operating systems are not guaranteed to support all the necessary software to train neural networks.

Paperspace is chosen to represent special purpose CSPs for three reasons. (1) It the most used special purpose CSP according to Kaggle. (2) It offers a variety of GPU hardware in its instances. (3) It is a mature business having offered GPU compute services for five years (Paperspace, 2016).

Pricing strategies and models used by Azure and Paperspace are collected through secondary research of their pricing and the features and deliverables of their deep learning services. All information is collected from the services' public web pages, and the information is analyzed according to the pricing framework by Wu et al. (2019) to determine which models are used.

³ Floydhub usage data is based on Kaggle 2019 survey.

3.2. Performance and Cost Efficiency of CSP GPU instances

Shretha et al. (2019) explain that the primary shortcomings of training deep learning models include having to make assumptions and optimizing hyperparameters that are used to tweak the model behavior. Each time data scientists make such assumptions they must train the neural network and observe the training behavior and predictive accuracy. Network training can be very time-intensive, sometimes taking hours or days. After training, a neural network can be overfitted so that it does not work well with new data outside the training dataset, requiring the data scientist to restart training with different assumptions or hyperparameters, and ultimately consuming more time.

Aho et al. (2020) highlight the iterative nature of data science projects. Data scientists go through several versions of their models from a minimal viable product to production quality. Software developers, like many other professionals, loathe interruptions. Meyer et al. (2014) report that developers perceive to be most productive when they do not switch between tasks. Since data scientists also develop software (albeit for the purpose of modeling data) it is safe to assume they also prefer to avoid task switching. If data scientists have to wait a long time for one model to train, then task switching is inevitable. Quick experimentation – shifting between model versions and instances – is thus an important value driver for data scientists. It allows them to iteratively tweak their deep learning model to achieve the desired predictive accuracy.

Thus, reducing the time per experimental iteration is valuable to data scientists as they can find the optimal solution efficiently and deliver a neural network that delivers business value in a reasonable timeframe. However, from the business point of view, training time is not the only factor. Faster training requires using relatively expensive GPU clusters. A balance between both time and cost is key—thus this research focuses on measuring cost efficiency as a decision driver for GPU instance choice.

3.2.1. Metrics and data

3.2.1.1. Metrics

Real time measurements are preferred by many researchers in measuring the performance of neural network training. DAWNBench (Coleman, 2019) defines performance as Time-to-Accuracy, the number of epochs needed for the neural network to reach a prespecified predictive score multiplied by the time-per-epoch. Ma et al. (2018) are also proponents for using time-based metrics like speed (e.g. trained data points per second) instead of an abstract one like FLOPs.

The training process of the neural network first calculates the number of batches per training step (known as epoch) by dividing the training dataset over the batch size and running the training algorithm for a number of batches. This process is repeated for a number of epochs chosen by the data scientist. After training, the data scientist observes the model accuracy then decides whether to continue or restart training. Every time the training is resumed or restarted, the data scientist may change the batch size (or other hyperparameters) or the entire neural network architecture until the desired predictive accuracy is achieved or resources are exhausted. After each training experiment with a preset number of epochs, the data scientist gains information about the network behavior and decides how to adapt the architecture and hyperparameter assumptions for better predictive performance.

One of the hyperparameters that must be decided upon before training is the learning rate, the rate at which the neural network updates its internal parameters with each pass on one batch of data. Learning rates are challenging to decide upon for the data scientist as smaller learning rates tend to take many more epochs for the network to achieve the desired accuracy, while too large learning rates make the network overfit.

Another critical hyperparameter is the batch size which indicates the number of the training data points that are bundled to do one training step for the neural network. Analysis of the winning entries in the DAWNBench competition shows ever increasing batch sizes as a key characteristic of winning entries (Coleman, 2019). Smith et al. (2018) also experimented with both hyperparameters, finding that increasing batch sizes can lead to more accuracy as compared to optimizing learning rates. Batch size is limited by the amount of GPU memory available on the instance, meaning that when a data scientist decides to change the batch size, a different instance may be cost optimal. Data scientists must experiment both with learning rates and batch sizes to achieve the desired accuracy which rapidly increases the amount of possible hyperparameter combinations, and further confirms the importance of quick experimentation.

This research focuses on training Time per Epoch as a primary value driver as the faster the epoch is trained, the faster the data scientist gains knowledge about how to tweak the network architecture and its hyperparameters. Accordingly, Time per Epoch will be used as the performance metric to measure for the case study. It is also used for calculating the instance Cost Efficiency to help data scientists make sense of choosing between instances. Due to potential background processes running on the instance, three epochs are run, and the average Time per Epoch recorded.

Time per Epoch: The amount of time (in seconds) needed to train the neural network for one epoch (a single iteration over the full training dataset).

Cost Efficiency for an instance is calculated based on the approach Roloff et al. (2012) took to calculate cost efficiency of different CSPs in the case of High Performance Computing. First, the Time per Epoch performance results are normalized, and expressed in percent since the absolute performance data can vary greatly. Second, the Cost Efficiency metric is calculated by scaling the normalized performance by the hourly price of the instance (in this case, calculated in US dollars). This approach for Cost Efficiency calculation is designed to have a fairer comparison between instances with variable performance will be useful when choosing between instances from different CSPs.

Performance is normalized compared to the highest performing instance – with the shortest Time per Epoch. This allows data scientists to trade off performance loss with cost efficiency when choosing a slower instance.

Cost Efficiency (Dollar/hour) = Normalized average Time per Epoch (%) × Hourly cost (Dollar/hour)

3.2.1.2. Data collection

A primary research approach is used for collecting performance data, through directly running two deep learning cases on each instance in scope. The data needed to calculate Cost Efficiency is the hourly price of using an instance and the amount of time a given neural network takes to train for one epoch. The scope of the pricing data collection excludes free

services and enterprise-targeted services without public pricing information as they do not relate to choosing between instances on a cost basis.

Deep learning projects require GPU instances for their workloads. CSPs offer many types of instances, each serving a specific need, with GPU instances only accounting for a portion of them. Special purpose CSPs like Paperspace offer primarily GPU instances. Azure offers a vast array of instance types serving different applications. Even though deep learning can be done on CPU instances, albeit at a much slower pace, data collection for pricing information is focused only on GPU instances that are designed for deep learning applications.

On Azure, there are multiple price points per instance, including (1) “Pay as you go” which shows pricing to use the instance per hour and bills the customer by the second, (2) “1 or 3 year reserved” which entails long-term commitment to using the instance and is discounted compared to “Pay as you go”, and (3) “Spot” pricing which fluctuates over time based on balance of supply and demand and with the caveat that the instance can be stopped at any point by Azure with a 30 seconds eviction notice (Microsoft, 2021b). Since this research aims to maximize the data scientist flexibility, only “Pay as you go” instance pricing is used for cost efficiency calculation. Paperspace instance pricing is by the hour – and is billed by the second. Since instance pricing is provided on a per-hour basis by both CSPs, yet billing is done per second, the cost of training one epoch is calculated by prorating the instance hourly price based on the training duration in seconds.

While Azure has pricing pages for different geographical regions, Paperspace pricing is only based in US Dollars. For comparability, pricing analysis for Azure is thus limited to US Dollars based on the “East US” region pricing. All prices were collected during May 2021, and a sample of the pricing tables can be seen in the Appendix.

3.2.2. Case study design

Neural networks are used in a large variety of business domains, and different neural network types – neural architectures – serve specific domains. Neural architectures represent the way the individual neurons are arranged, making the network more suitable to tackle a certain task.

According to Shretha et al. (2019) and Alom et al. (2019), two architectures are prevalent and extensively researched: convolutional neural networks (CNNs) and recurrent neural networks (RNNs). These two architectures are used heavily in the business environment to work with images, natural language, and time series data. CNNs and RNNs are also the two most used architectures by data scientists among deep learning algorithms, with 29% and 17% of data scientists, respectively, according to Kaggle’s data science survey (Kaggle, 2020). Accordingly this study focuses on CNNs and RNNs as the basis for designing the case studies detailed below.

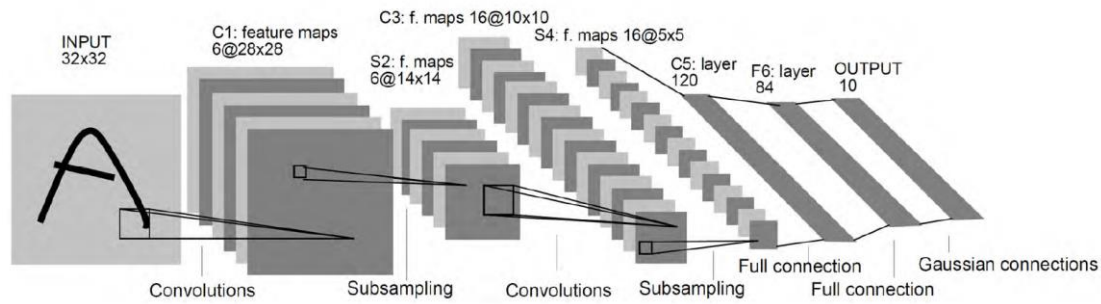


Figure 5. A sample convolutional neural network (CNN) with seven layers. Each layer extracts more nascent features from the images. Source: Shrestha & Mahmood (2019)

CNNs are neural network architectures designed to mimic the human visual system. Their design allows them to extract features from 2D and 3D images or videos by detecting different patterns, textures, and other visual abstractions. After extracting myriad visual features, CNNs usually conclude with a classifier to make a prediction based on the image data, making them suitable for image segmentation and classification tasks. Figure 5 illustrates the various operations CNNs do to extract visual features (Shrestha & Mahmood, 2019). CNNs are employed by social media platforms to tag faces (Trigueros, 2018) and objects (Gong, 2014) in images, by drones navigating their surroundings (Kim & Chen, 2015), and even for identifying positive COVID-19 patients from chest X-ray images (Wang et al., 2020). Figure 6 from Qin et al. (2018) illustrates the conceptual similarities between the human visual cortex and a CNN.

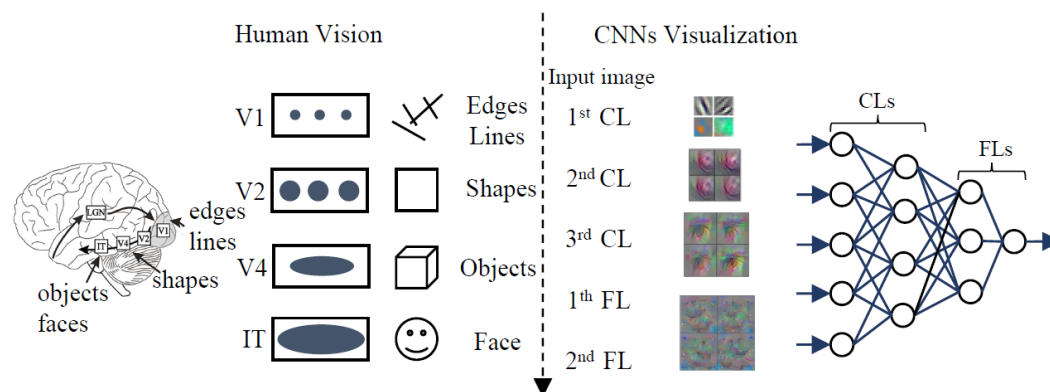


Figure 6. Simplified view of how the human visual cortex processes visual information. In parallel, a CNN is shown with the image features extracted in each layer. Source: Qin et al., (2018)

RNNs are designed to have a sense of time, maintaining memory or historical information as it processes sequential information, and using that memory to make better predictions. This is particularly useful to maintain context in natural language processing (NLP) of long pieces of text or for maintaining dialogue in question answering chatbots. RNNs have been used in challenging tasks like machine translation (Zhou, 2016), for building intelligent chatbots to conduct customer service (Xu, 2017), and for predicting BitCoin price based on sentiment analysis (Pant et al., 2018). Figure 7 illustrates different ways RNNs encode memory into the neural network layers (Karpathy, 2015).

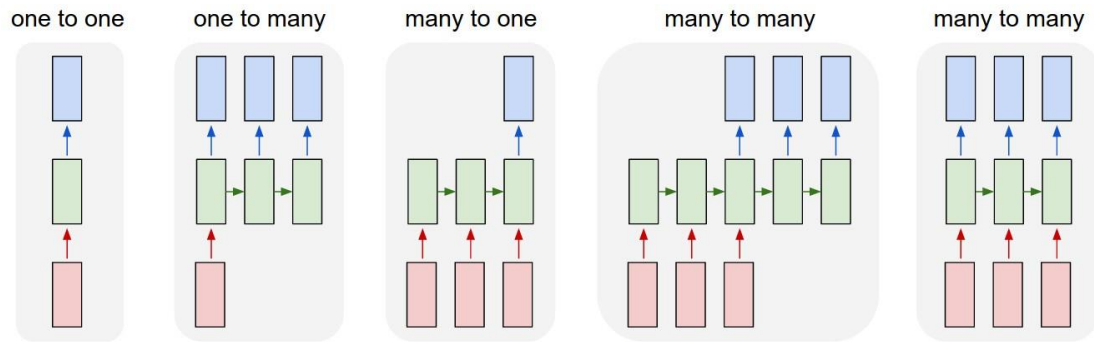


Figure 7. A visualization of how recurrent neural networks (RNNs) vary from traditional neural networks. Each color represents a network layer. RNNs show interconnectedness within each layer, unlike traditional networks moving only forward (leftmost box). This allows RNNs to preserve sequence information critical for NLP tasks. Source: Karpathy (2015).

As CNNs and RNNs are used widely and have different underlying architectures, it is plausible that the choice of a neural network architecture results in a different optimal choice of a GPU instance in terms of cost efficiency. To investigate that possibility, the instance cost efficiency is measured in two deep learning use cases: an image classification case study by a CNN, and a sentiment classification case study based on text by a RNN.

The process of using a deep neural network for any task requires making many decisions including how to process the data, deciding on the neural architecture, tuning the hyperparameters, and choosing the training algorithm. This makes a case study approach ideal to design a scenario that is relevant to real world use cases.

Designing a case study to test the time needed to train a neural network for one epoch minimally requires defining four elements: (1) the dataset on which the neural network is trained, (2) the neural network architecture and its specific variant, (3) the network hyperparameters, and (4) the cloud instance which provides the hardware for the training. The two case studies below describe choices for the first three elements. Multiple instances from Azure and Paperspace are iterated on for the fourth element to measure the Time per Epoch and Cost Efficiency for each.

Each case study is complemented by source code used to train the neural network on the dataset and reports back the running duration per epoch. The full source code for this research is inspired by publicly available software libraries and tutorials, and is published on a GitHub repository⁴. Section 3.2.3.2 describes how the code is used to measure the Time per Epoch of training. The code for running both case studies is based on the FastAI software library (Howard & Gugger, 2020), and its tutorials for image classification, sentiment analysis, and distributed training. Distributed training code is necessary to utilize the full capacity of instances that include multiple GPUs and works by training multiple batches of data in parallel, thus speeding up the Time per Epoch.

3.2.2.1. Case study 1: Image classification by CNN

This case study simulates a common image problem faced by many businesses – image classification. Image classification is the process of automatically choosing to label an image from a set of labels. It can be used as a cost-saving measure, for example by detecting whether a product is defective or not in manufacturing optical quality control (Weimer et al., 2016) by

⁴ Code repository can be found at https://github.com/jrateb/pricing_dlaas_thesis

taking pictures on the assembly line and sending the result to another system to take action in case a defect is found. Image classification can also be used by online retailers to help shoppers conduct a visual search and find a matching item in a retailer's catalogue from a user submitted photo, which can increase the retailer sales revenue (Bell & Bala, 2015).

Dataset & preprocessing

ImageNet (Deng, 2009) is the standard public dataset for image classification algorithms. New neural network architectures boast about their performance by publishing new records of ImageNet classification accuracy (defined as proportion of test images withheld from the training algorithm that are correctly classified). Since ImageNet is very large and requires a long training time and high cost, this case study will use a smaller dataset called Imagenette (Howard, 2019). Imagenette is composed of a subset of ten classes of ImageNet. This makes Imagenette a more realistic representation of an image classification task within a business context. The dataset includes 9,469 images for training and 3,925 images for validation. A separate validation dataset is key to ensure the neural network is not overfitting on the training data.

Imagenette data is processed before passing to the CNN for training. A randomized transformation is applied to each image, including flipping, rotating, warping, and changing lighting conditions. In addition, each image is randomly cropped with at least 75% of the original image remaining. These transformations help make the CNN more resilient to image changes and allow it to be more predictive when faced with previously unseen images that are photographed in less than ideal conditions (Zhang, 2015). Finally, since image sizes may vary, each image is resized to 224×224 pixels to fit the input layer of the neural network.

Neural architecture

The case study uses a well-studied CNN variant, ResNet50. ResNet50 is a type of residual neural network introduced by Microsoft researchers (He et al., 2016). Residual neural networks are designed to allow for deeper architectures with less computational complexity which helps them extract more patterns from images and achieve high predictive classification accuracy. ResNet50 is chosen as it is well benchmarked by researchers (Mattson et al., 2019 and Yamazaki, 2019) for image classification tasks and its architecture code is readily available in open source.

Hyperparameters

As the case code is run on GPU instances of various power, a batch size of 64 images is chosen as it fits the amount of memory in the smallest instance in the experiment scope. A learning rate of 0.001 is used as it falls in the optimal learning rate range as shown in the Imagenette classification tutorial (FastAI, 2021c).

3.2.2.2. Case study 2: Sentiment classification by RNN

The second case study focuses on text classification, a natural language task that is commonplace in business. RNNs are particularly useful for text classification, as they maintain a historical and contextual information in the text as the network processes the text sequentially (Zhou, 2015), thus simulating having both short- and long-term memory. Text classification is used in automating customer service via chatbots (Nuruzzaman & Hussain, 2018), detecting spam content in e-mails and online product reviews (Ren & Ji, 2017), and in stock prediction based on financial news (Vargas, 2017).

Specifically, the case utilizes a RNN to perform a sentiment classification task, which is the process of rating a piece of opinion text as positive or negative. Machine learning algorithms have been long used for sentiment classification, with movie reviews being a popular dataset choice for researchers since movie reviews and associated ratings data are available online (Pang, 2002).

Dataset & preprocessing

Following the lead of text classification researchers, the case will use the IMDB movie review dataset (Maas, 2011). The dataset includes 25,000 polarized movie reviews for training, and another 25,000 for predictive validation. Each review is labeled either positive or negative. The dataset is also well documented in a text classification tutorial by FastAI, our software library of choice (FastAI, 2021b). The dataset was not processed any further before training.

Neural architecture

A variant of RNN referred to as AWD-LSTM (Merity, 2017) is an ideal choice to use for sentiment classification. AWD-LSTM architecture is designed to reduce overfitting in text classification tasks, allowing it to work better with previously unseen review text. AWD-LSTM is also included in a benchmarking study by Klyuchnikov et al. (2020), and is proven to outperform other RNN architectures in financial sentiment analysis (Araci, 2019).

Hyperparameters

Similar to case study 1, a batch size of 64 pieces of text was chosen based on the available memory in the smallest instance, also with a learning rate of 0.001.

Table 3 shows a summary of the specifications of both case studies.

Case study design summary		
	Image classification	Sentiment classification
Neural architecture	CNN (ResNet50)	RNN (AWD-LSTM)
Dataset	Imagenette (subset of ImageNet)	IMDB movie reviews
Training records	9,469 images	25,000 reviews
Validation records	3,925 images	25,000 reviews
Batch size	64	64
Learning rate	0.001	0.001

Table 3. Neural architecture, dataset, and hyperparameters used for the case studies.

3.2.3. Performance measurement

3.2.3.1. Experiment instances

GPU instances on Azure and Paperspace are used to train each of the two neural networks. Azure and Paperspace offer different specifications (hardware configuration bundles) of their GPU instances. Hardware configuration bundles are designed by CSPs to serve different audience needs. This is why it is plausible to expect variable instance performance for each neural network, given the underlying architectures. For that reason this research considers only the GPU instances from each provider that are targeted for deep learning applications. Other GPU instances on offer from Azure that cater to graphical and gaming applications are thus excluded from this study.

Azure provides instances with both single and multiple GPUs, while Paperspace only offers single GPU instances at the time this research is conducted. Since instances with multiple

GPUs can provide faster training, thus speeding up the experimentation cycle of data scientists, they are included in the study scope.

Table 4 lists the pricing and hardware characteristics of the GPU instances from Azure and Paperspace (Paperspace, 2021) on which neural network training time is measured. For Azure, both the NC (Microsoft, 2021d), and the NCv3 (Microsoft, 2021e) series are tested. For simplicity, only the hardware characteristics most relevant to training deep learning models are described. For Paperspace, disk space is not tied to the instance, but rather to the subscription type. A \$24 USD/month professional subscription was used to run the experiments which provided 1 terabyte (TB) of disk space.

GPU instance pricing and hardware characteristics					
Instance Type	Price/hr (\$USD)	GPU(s)	GPU memory (GB)	Memory (GB)	Disk space (GB)
Azure					
NC6	\$0.90	1 x K80	12	56	340
NC12	\$1.80	2 x K80	24	112	680
NC24	\$3.60	4 x K80	48	224	1440
NC6s v3	\$3.06	1 x V100	112	16	736
NC12s v3	\$6.12	2 x V100	224	32	1474
NC24s v3	\$12.24	4 x V100	448	64	2948
Paperspace					
P4000	\$0.51	1 x P4000	8	30	1024
P5000	\$0.78	1 x P5000	16	30	1024
P6000	\$1.10	1 x P6000	24	30	1024
V100	\$2.30	1 x V100	16	30	1024

Table 4. GPU instance characteristics (hardware configuration bundles) and their hourly pricing based on Pay as you Go.

3.2.3.2. Time per Epoch measurement

This section outlines the general process to measure the Time per Epoch for each case study per instance. It also describes in a stepwise manner what the code does once it is executed on an instance. For each CSP, a specific setup procedure was followed to prepare for code execution; these procedures are described in the next section. The procedure is repeated for each case study and on each of the 10 instances described above, and is logically identical for both case studies. The following table illustrates the experimental design variables, with a total of 20 distinct experiments.

Case study	Instance
Case 1 (CNN)	NC6
	NC12
	NC24
	NC6s v3
	NC12s v3
	NC24s v3
	P4000
	P5000
	P6000
	V100
Case 2 (RNN)	NC6
	NC12
	NC24
	NC6s v3
	NC12s v3
	NC24s v3
	P4000
	P5000
	P6000
	V100

Table 5. Overview of the case study experimental design variables. Each case study is run on each instance for performance measurement.

- a. A new instance (e.g. NC6) is remotely started, using Ubuntu Linux as the operating system. Then, the necessary software for training deep learning models (e.g. FastAI) is installed if not already included in the instance.
- b. A secure connection to the newly created instance is established, then a script is initiated to pull the training and measurement code from an online repository. The source code is identical on all instances to ensure comparability, with slight modifications for Azure since special code is required to train a network on multiple GPUs.
- c. Measurement code is executed. It encompasses the following steps:
 - i. Case dataset is downloaded from its public source.
 - ii. Dataset is loaded, separating training data from validation data.
 - iii. Data preprocessing logic is implemented. Some portions of data preprocessing actually occur (e.g. image transformations) during the training process, and are thus included in the training time. This is common, and is a realistic simulation of training workloads.
 - iv. Neural network architecture and its hyperparameters are defined as a model.
 - v. Model training is started for three epochs. Each epoch includes both a training round over the entire training dataset and a validation round over the validation dataset. The model automatically reports the time consumed per epoch.
 - vi. Time per Epoch is separately recorded for each of the three epochs.
- d. Instance is stopped. Afterwards, the instance and associated storage are deleted to avoid incurring additional costs.
- e. The procedure is restarted with the next instance.

3.2.3.3. CSP-specific setup procedures

Each CSP has its own process for setting up an account, initiating an instance, running code on it, reporting the results, and cleaning up resources. This section summarizes the steps taken on each CSP to run the Time per Epoch measurement code.

Azure

1. A new user account was created to ensure there are no performance interference from other Azure resources and to support reproducibility. Creating a user account on Azure is free of charge, and billing only starts after resources are used.
2. By default, Azure only makes limited GPU instances available for use. Thus, a customer service request was initiated to gain access to the NC, and NC v3 instances listed above. Access was granted within 24 hours.
3. Azure has a scripting environment to allow using code to create, start, and destroy resources such as GPU instances. Code based on this tutorial (FastAI, 2021a) was used to automatically create and start an instance with the prespecified instance type.
4. A secure connection is established with the newly started instance to push the code necessary to download the dataset and train the neural network of the case at hand.
5. After the code is run, and the time per epoch is recorded, the instance is first stopped then deleted along with its associated networking and storage resources. Even though instances do not incur costs when stopped, persistent storage continues to incur per-usage cost.
6. Steps 3 to 5 were repeated for each case study and instance type.

Paperspace

1. A new user account is created. While Paperspace offers a free subscription, it provides limited access to GPUs, thus the Pro subscription – at a monthly cost of \$24 USD – was selected to allow access to all GPU instances. Pro subscription also includes one terabyte of persistent storage. Unlike Azure, no additional per-use cost is required for the storage. This persistent storage allows quickly switching between GPU instances without the necessity to redownload data.
2. A new code notebook is created. A slightly modified code is used for dataset download and network training compared to Azure. The main difference is the removal of the distributed training portion of the code as Paperspace only offered single GPU instances at the time of running the experiment. The notebook is yet unusable as it is not attached to any computing resources.
3. A GPU instance is then attached to the notebook for the code to run on. Paperspace offers instances with preinstalled software packages. The package based on FastAI was chosen.
4. The notebook code is executed, and the Time per Epoch is recorded. Afterwards, the instance is detached, thus stopping pay-per-use billing.
5. Steps 3 and 4 are repeated with each case study and instance type. On Paperspace, the software environment of the instance is persistent on the subscription storage, making switching between GPU instance faster than Azure where all the necessary software must be reinstalled whenever a new instance is tested.

3.3. Procedure design for cost-efficient instance choice

The goal of this research is to understand how pricing model choices of CSPs impact cost efficiency of training deep learning models. To make the study actionable by data scientists, the findings are used to design a procedure to help them quickly elect an instance for their particular deep learning use case.

Data scientists may demand the cheapest instance per hour, the most cost-efficient (balancing performance with cost), the fastest per epoch, or choose an instance based on other criteria. The procedure introduced here is designed to elect the most cost-efficient instance, although it can be easily modified to use another decision criterion. Since the optimal instance choice depends on the model architecture and batch size, the procedure is designed to be as quick as possible with the intention of being repeated every time an instance needs to be elected.

4. Research Findings

4.1. Pricing model analysis

This section discusses the pricing and value delivery strategies utilized by Azure and Paperspace deep learning services as well as the specific pricing models used for profit maximization. Analysis is based on the publicly available service descriptions and pricing tables. Table 6 shows a brief description of the deep learning services offered by both CSPs for price model analysis. Screenshots of their detailed pricing tables is available in the Appendix.







Azure services		
	Azure Compute Infrastructure	A wide array of virtual machines (instances). It includes related storage, and networking infrastructure.
	Azure Machine Learning	Software services including data science notebooks, team collaboration, data labeling, model development and management tools.
	Azure Cognitive Services	Packaged AI services targeted at specific domain use cases. Includes decision, language, vision, and speech services.
	Azure Bot Service	Speech- or text-based intelligent dialogue service that can be customized to different business scenarios.
Paperspace services		
	Paperspace Core	A small selection of virtual machines infrastructure. Paperspace does not offer other virtualized hardware services.
	Paperspace Gradient	Software services including data science notebooks, team collaboration, model development, and management, as well as interfacing with Core instances.

Table 6. Overview of deep learning services by Azure and Paperspace for pricing analysis.

4.1.1. Azure pricing models

Azure services for deep learning are available under the Azure AI Platform umbrella (Microsoft, 2021a). These include four main services: deep learning infrastructure, Azure Machine Learning, Azure Cognitive Services, and Azure Bot Service.

4.1.1.1. Azure Compute Infrastructure

Virtualized hardware infrastructure is the core offering of CSPs. To train and utilize deep learning applications, several cloud services are needed including GPU-powered virtual machine (compute) instances, storage, networking, identity management, etc. Compute instances are the primary cost drivers and the focus of the pricing model analysis.

Azure GPU-powered instances fall under the N-series (Microsoft, 2021c). These encompass multiple subseries of instances. Each series uses specific hardware, and the instances within the series represent different sizes where each instance has progressively more memory, CPU power, total GPU memory, and storage space. Table 4 earlier (section 3.2.3.1) showed such specifications of the NC and NCv3 series.

Several pricing models are utilized in offering GPU instances, as CSPs strive to target a wide variety of customers using the entire spectrum of pricing strategies. Azure is no different—with offering a hybrid of four pricing models for its instances. **On-demand pricing model** is the first pricing point visible in the instance table; it is used to encourage data scientists to try out different instances, making **Pay as you Go** the predominant value delivery strategy. Azure also offers steep discounts for data scientists making an upfront commitment to use an instance for one or three years, which adds the **reserved pricing model** to the mix.

The instances are offered as a hardware configuration bundle. While data scientists can elect the instance (bundle), they cannot create their own. This **product mix pricing model** is a popular choice by CSPs as they help reduce resource underutilization and increase profits compared to offering the individual resources or allowing custom bundles (Bakos & Brynjolfsson, 1999). On top of on-demand and reserved pricing, Azure also offers the **spot pricing model**, an auction-based approach which sells unused resources at a dynamic—usually heavily discounted—price based on the balance of demand and unused instances (supply).

All the pricing models so far were on an instance basis. Azure also utilizes **price discrimination model** depending on the geographic region on which the data scientist is utilizing the instances. The choice of the region is an important one as all other cloud resources connected to the instance must reside in the same region, and the region choice impacts the network connection speed between Azure and the data scientist.

As Azure is the third largest CSP by market share, it is used by many companies for infrastructure services beyond deep learning. It would be beneficial for Azure to avoid having its customers use multiple CSPs and the offering of GPU instances can be considered a way to have a complete portfolio of infrastructure services. This would protect its market share, and indicates that Azure Compute Infrastructure has a **Good to Have** value proposition.

Despite covering the spectrum of all pricing strategies, three out of five pricing models used by GPU instances follow **market-based pricing strategy**. This is not surprising as all general purpose CSPs offer similar instance configurations pressuring infrastructure pricing to follow market dynamics (Lee, 2019).

4.1.1.2. Azure Machine Learning

Azure Machine Learning (ML) services include a suite of software that helps speed up the data scientist work. This includes code notebooks, model management software to help with training, and launching machine learning models. They also include tools to facilitate enterprise team collaboration and source code management, amongst others.

Azure does not charge its customers for using ML software. However, using the service incurs hardware infrastructure utilization including compute, storage, and security services.

The hardware resources utilized to run the software are charged on a **Pay per Use** value delivery basis.

While using Azure ML, the data scientist must still make choices on which instance to do the neural network training. Without guaranteed outcomes, the value proposition of this service is **Good to Have**.

Also similar to Compute Infrastructure services, Azure ML uses the **on-demand pricing model** for launching the necessary instances to host the code notebooks and manage models. Since the data scientist must provision certain resources decided upon by Azure to make Azure ML work, this indicates the use of a **razor-and-blades pricing model**. Razor-and-blades pricing is where certain consumable products – blades – are services that have to be utilized to get the full benefit from Azure ML service – the razor. The pricing models of Azure ML suggest a hybrid pricing strategy with elements of **market-based** and **value-based pricing**.

4.1.1.3. Azure Cognitive Services

Cognitive Services are a suite of application services allowing customers to directly use pretrained AI services without needing to collect data, set up any data science infrastructure, or directly train any neural networks. All Cognitive Services follow the **Pay per Task** delivery model since they offer and price their services per unit of input, and each input type is designed to be easily translated to customer value.

Cognitive Services encompass four domains, with each domain including multiple services with unique pricing models. The following summarizes their description and pricing.

- (1) Decision support services including anomaly detection and content moderation. Azure prices these services per 1,000 transactions. In some decision support services, like Content Moderator, price per 1,000 transactions is tiered by volume with gradually reduced pricing after 1 million, 5 million transactions, etc.
- (2) Natural language services including sentiment analysis and translation. Prices are based on task and input size with the potential for discounts when working with large volumes of text. An example of pricing by task is where a normal text translation costs \$10 USD per million characters, while text translation of a full document costs \$15 USD per million characters. The higher price for documents is for the added value of preserving the document structure and format.
- (3) Speech services including speech to text, text to speech, and speech translation. For services based on speech, pricing is per hour of audio, while text-based services are priced by million characters. Special services pad the price tag, like added fees for processing particularly long pieces of text to speech.
- (4) Vision services including feature extraction from images and emotion detection in faces. Pricing for extracting image features is per transaction and per feature. For example, identifying landmarks and brands in an image would constitute two transactions. Face and emotion detection services are priced per transaction. All vision services have a tiered pricing with discounts for large volumes.

Both **outcome-based** and **freemium pricing models** are used across Cognitive Services. The outcome in outcome-based pricing is dependent on the type of service and does not require the data scientist to launch any instances or build deep learning models. Even though freemium is based on a market-based pricing strategy, it is pervasive across all Azure services. Elements of cost-based pricing (volume discounts) are utilized, but only in a limited

scope. Outcome-based pricing (**Good to Do** value proposition) is the unique differentiator of Cognitive Services indicating a **value-based pricing strategy**.

4.1.1.4. Azure Bot Service

Bot Service provides text- and speech-based chat bot experience that customers can embed in any digital channel. It builds on top of Cognitive Services to enable intelligent bot functionality.

Pricing Bot Service takes a layered approach with a pricing per 1,000 chat messages as the first layer, also following the **Pay per Task** delivery model. Bot Service automatically provisions one of the Cognitive Services like Language Understanding and Speech, so their pricing is added on top of the per message pricing. Like Cognitive Services, Bot Service also follows a **value-based pricing strategy** with **outcome-based** and **freemium pricing models**.

4.1.1.5. Summary

Azure provides several services to support deep learning needs. Starting with infrastructure, it offers various pricing models following a market-based strategy. Cognitive Services and Bot Service focus primarily on a Pay per Task delivery model utilizing value-based pricing strategy as their prices depend mainly on the input units. Pricing of services builds on each other, with Azure ML relying on infrastructure pricing, and Azure Bot relying on Cognitive Services pricing. With this mixture of services and pricing models, Azure fulfills the needs of several customer types. Experienced data scientists can elect optimal instances for their models. Teams can utilize Azure ML to manage and monitor multiple models. Businesses wanting to implement AI capabilities without investing in training models can directly use Cognitive and Bot services and pay per task. The strategies and models employed by Azure are summarized in table 7.

Service	Pricing strategy	Value proposition	Delivery model	Pricing model(s)
Azure Compute Infrastructure	Predominantly Market-based pricing	Good to Have, Good to Do, and Good to Be	Pay as you Go	On-demand Reserved Product mix Discriminatory Spot
Azure Machine Learning	Mixture of Value-based & Market-based pricing	Good to Have	Pay per Use	On-demand Razor-and-blades
Azure Cognitive Services	Value-based pricing	Good to Do	Pay per Task	Outcome-based Freemium
Azure Bot Service	Value-based pricing	Good to Do	Pay per Task	Outcome-based Freemium

Table 7. Summary of the value propositions, pricing strategies, and models employed by Azure deep learning services. Azure offers pricing and delivery models across the entire value spectrum.

4.1.2. Paperspace pricing models

Paperspace is different from Azure in that it requires a paid monthly subscription for a data scientist to access all the GPU instances on offer. Its pricing is based on the subscription tier and the instance utilization costs are then added on top.

Paperspace offers two services: Gradient and Core. Gradient started as a hosted notebook environment and progressed to encompass several software tools to manage deep learning projects, keep track of experiments, datasets, and models. Gradient is powered by Paperspace Core, which offers a series of virtualized GPU instances.

4.1.2.1. Paperspace Core

Core pricing is based on **Pay as you Go** delivery strategy with hourly utilization pricing. Only one price is available for each instance type without any long-term reserve pricing, and without spot pricing.

Instances available on Core vary only by the GPU hardware (including the GPU card model and its available memory). However, the memory and CPU power are kept constant across all instances in contrast to Azure's instances, which proportionally increases all hardware resources for each instance size.

Given the focus on hourly pricing, Core follows a **value-based pricing strategy** and **Good to Have** value proposition, specifically using an **on-demand pricing model**. Unlike Azure, product mix pricing is not utilized by Paperspace Core since only the GPU specifications vary between different instances.

4.1.2.2. Paperspace Gradient

Gradient pricing is based on multiple tiers of monthly subscription, starting with a Free tier that only offers access to low performance GPUs, and moving on to Developer and Professional tiers. The Professional tier provides access to all GPU instances on offer. On top of GPU instance availability, each tier is differentiated with the persistent storage volume, the number of concurrent notebooks that can be run, and the type of customer support. Persistent storage is a feature allowing data scientists to persist their software configuration and datasets while choosing between GPU instances, significantly speeding up switching between instances.

This indicates that Gradient uses **Pay as you Can** (or feature as a service) delivery strategy by making better features and performance available for higher prices. The use of a **feature-based pricing model** indicates a **Good to Be** value proposition based on a **value-based pricing** strategy.

4.1.2.3. Summary

Paperspace offers instances and software services directly aimed at data science teams. Overall, Paperspace uses a value-based pricing strategy using on-demand and feature-based models which aligns well with a special purpose CSP. Value-based pricing aims to achieve the highest possible price given the subjective value a customer places on the services offered, which in turn maximizes profit—a clear goal for a small scale CSP competing with large incumbents. The strategies and models employed by Paperspace are summarized in table 8.

Service	Pricing strategy	Value proposition	Delivery model	Pricing model(s)
Paperspace Core	Value-based pricing	Good to Have	Pay as you Go	On-demand
Paperspace Gradient	Value-based pricing	Good to Be	Pay as you Can	Feature-based

Table 8. Summary of the value propositions, pricing strategies, and models employed by Paperspace deep learning services. Paperspace focuses its strategy on value-based pricing and stops on the value spectrum just shy of Good to Do (Feature as a service).

4.1.3. Insights from comparative analysis

As Wu et al. (2019) observed in their cloud pricing review, CSPs are shifting towards value-based pricing. This is also evident with the deep learning services by Azure and Paperspace.

Since Azure is a large general purpose CSP, some market-based pricing models still persist. This is expected as it offers several commodity cloud services with similar offers from competing large CSPs. Paperspace uses value-based pricing exclusively as a strategy to maximize profit and cement itself as a CSP for data scientists.

Bundling of hardware configurations in an instance is prevalent in both CSPs. While Azure scales up all hardware resources in an instance as the price increases within a series, Paperspace only increases GPU computational capacity while keeping other resources the same. Azure's approach maximizes revenue for Microsoft but can lead to underutilized resources and reduction in customer value. The Paperspace approach, however, can induce bottlenecks if the computational workload requires faster CPU or more disk space. Figure 8 shows a summary of the pricing and value strategies as well as the pricing models used by both CSPs according to Wu's (2019) taxonomy.

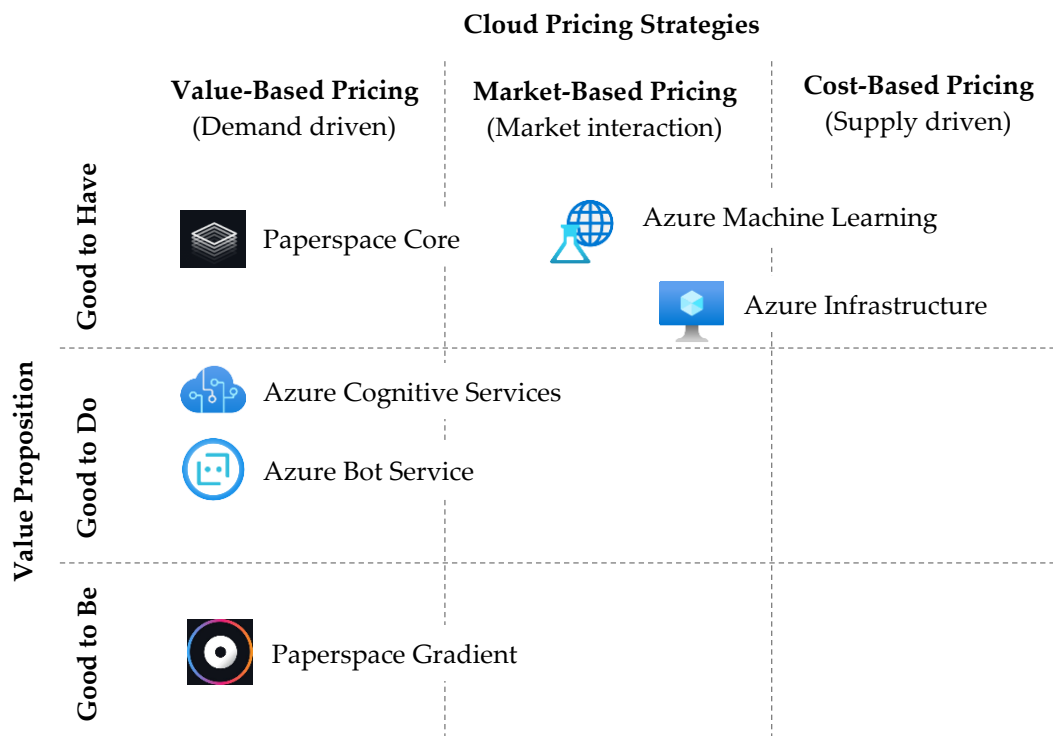


Figure 8. Classification of pricing strategy and value propositions of Azure and Paperspace deep learning services.

4.2. Performance and Cost Efficiency analysis

Training the deep learning models for both case studies was implemented on both Azure and Paperspace instances in scope. This section summarizes the results of Time per Epoch and Cost Efficiency for each instance and case.

4.2.1. Azure performance results

Table 9 shows the average Time per Epoch (in seconds) for Azure instances to train both neural architectures. The time it took to train a CNN on Imagenette dataset for one epoch took between 19.3 and 269.3 seconds, thus a data scientist can gain up to 93% performance improvement by shifting the training from a NC6 instance to a NC24s v3 instance. For training a RNN on IMDB dataset, one epoch took between 43 seconds and 433 seconds, with the NC24s v3 instance delivering 90% performance improvement compared to NC6. The performance gain from using the more modern V100 (NC6s v3) to K80 (NC6) in single GPU instances is similar for CNN and RNN, where V100 Time per Epoch was 17.1% and 19.6% of the K80's, respectively.

Azure Instance (GPUs)	Average Time per Epoch (seconds)	
	CNN (ResNet50), Imagenette	RNN (AWD-LSTM), IMDB
NC6 (1 x K80)	269.3	433.0
NC12 (2 x K80)	160.7	335.0
NC24 (4 x K80)	92.0	236.7
NC6s v3 (1 x V100)	46.0	84.7
NC12s v3 (2 x V100)	30.3	62.7
NC24s v3 (4 x V100)	19.3	43.0

Table 9. Time per Epoch (average of three runs) for CNN and RNN cases on Azure instances. Instance names are accompanied by the amount and type of GPUs that are included.

Figure 9 shows the relative performance of multi GPU instances compared to the single GPU instance in the same series. Since Azure offers instances with multiple GPUs with a matching linear increase in price, we can observe the performance gain from choosing an instance with two or four GPUs. These varied widely between the two cases. Among the NC series, NC12 – with double the price of NC6 – needed 60% of the training time for CNN and 77% for the RNN compared to NC6. The NC24 instance took 34% of the time as a NC6 for training the CNN, and 55% for the RNN.

Looking at the NCv3 series, the performance gains from multiple GPUs are more closely comparable between CNN and RNN. For NC12s v3, the CNN trained in 66% of the time, and the RNN trained in 74% of the time as the NC6s v3. For the NC24s v3, the relative training times were 42% for CNN and 51% for RNN. This indicates that performance gains from multi GPU instances could vary widely depending not only on the neural network architecture and the nature of data, but also on the GPU instance characteristics. This confirms the necessity of experimentation by data scientists to identify the cost optimal instance for their use case.

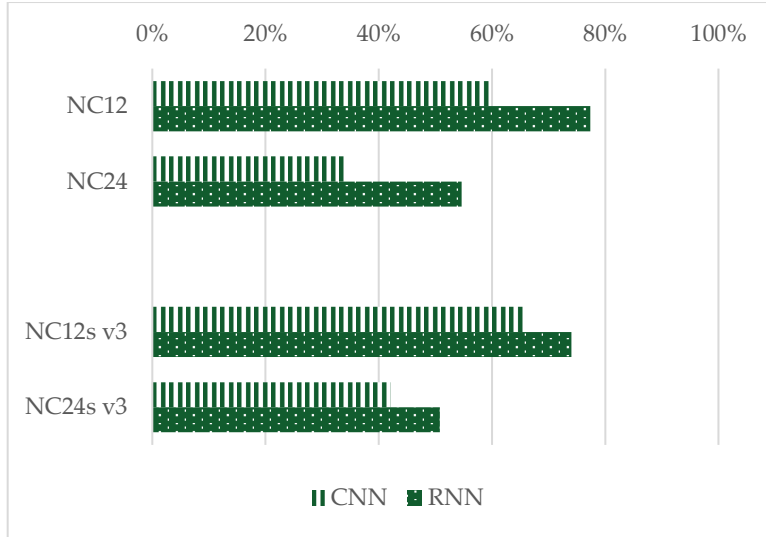


Figure 9. Relative performance of 2 and 4 GPU instances to single GPU instances within the same series. A smaller percent indicates better relative performance. Performance gains from multiple GPUs are higher for CNN compared to RNN architecture.

4.2.2. Paperspace performance results

On Paperspace, training time took between 57.3 and 161.7 seconds for CNN and between 83.3 and 225.0 seconds for RNN. Table 10 shows the average Time per Epoch (in seconds) for Paperspace instances to train both neural architectures.

Paperspace Instance	Average Time per Epoch (seconds)	
	CNN (ResNet50), Imagenette	RNN (AWD-LSTM), IMDB
P4000	161.7	225.0
P5000	127.0	139.3
P6000	92.0	111.7
V100	57.3	83.3

Table 10. Time per Epoch (average of three runs) for CNN and RNN cases on Paperspace instances.

Looking at relative performance compared to P4000 (see figure 10), the P5000 instance took 79% of the training time for CNN, and 62% for RNN. Other instances had roughly similar performance improvements for both architectures.

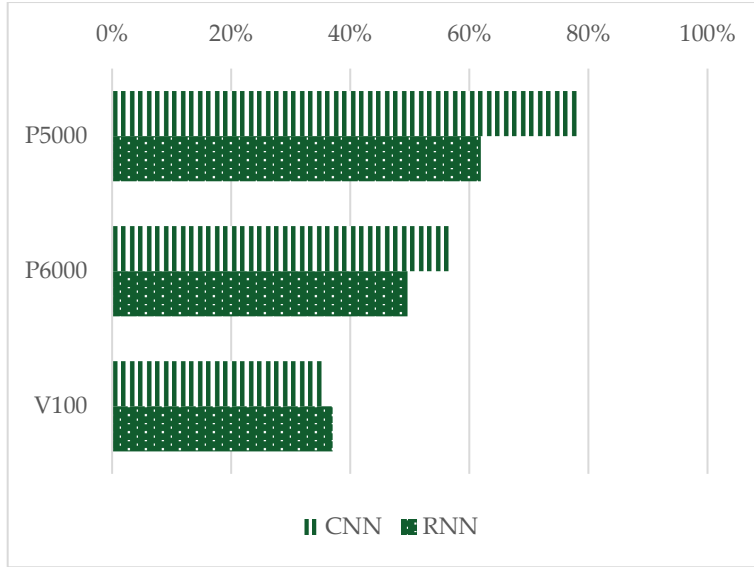


Figure 10. Relative performance of Paperspace instances to P4000. Only the P5000 shows variable performance gain between CNN and RNN architectures.

Both the NC6s v3 instance on Azure and the V100 instance on Paperspace utilize one V100 GPU card. While Paperspace V100 took 25% more time to train one epoch of CNN case as Azure’s NC6s v3, it takes roughly the same amount of time to train one epoch of RNN case. This variation further suggests that experimentation across instances and CSPs is essential to achieve the desired performance.

4.2.3. Cost Efficiency

As this research aims to support data scientists elect between CSPs and individual instances, performance for all ten instances from both CSPs is normalized to the fastest instance, Azure NC24s v3. Cost efficiency for each instance is calculated from the normalized performance as discussed in section 3.2.1.1. Figure 11 shows the Cost Efficiency results for both case studies.

All Paperspace instances show a better Cost Efficiency (lower dollars per training hour) than Azure instances for both case studies. Additionally, Cost Efficiency deteriorates when adding more GPUs in Azure, likely as a result of linear pricing combined with proportionally less increase in performance. Also, when using more GPUs in Azure, Cost Efficiency of RNN converges to that of a CNN on the NC v3 series. However, on the NC series, RNN Cost Efficiency diverges from CNN’s with more GPUs. This variability again highlights the challenge to predict training performance and to elect an instance given cost and performance requirements.

From the ten instances, the P4000 instance was most cost efficient for the CNN case, while the P5000 instance was most cost efficient for the RNN case. Only considering Azure, the NC6s v3 instance is the most cost efficient for both cases.

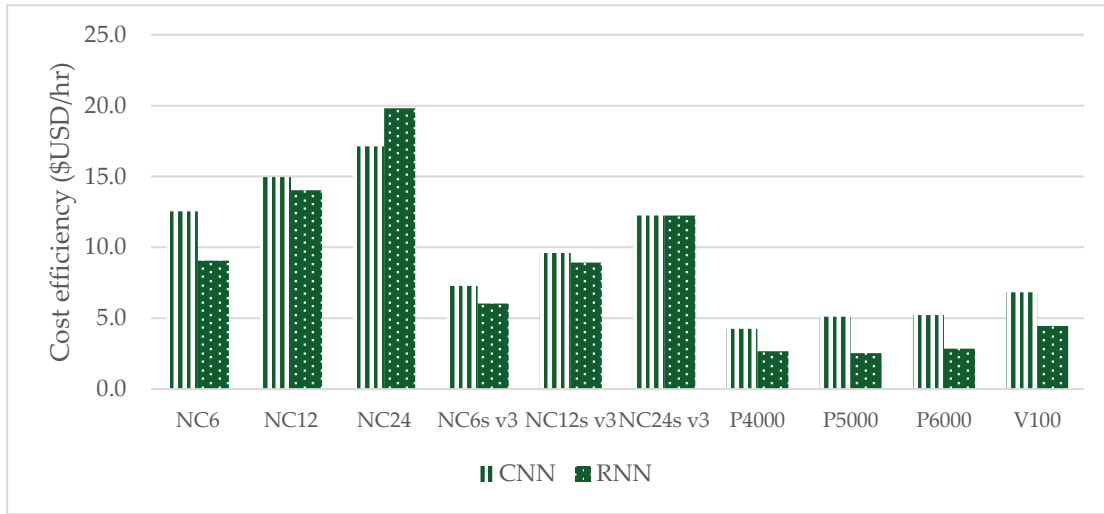


Figure 11. Cost efficiency of Azure and Paperspace GPU instances. Lower figures indicate higher efficiency. Overall, Paperspace instances are more cost efficient than Azure. Each network architecture has a different cost optimal instance: P4000 for CNN, and P5000 for RNN. Most instances show a cost efficiency gap between CNN and RNN architectures.

4.2.4. Discussion

The experiments in this research suggest that it is challenging to predict what is the optimal choice of instance for a given neural network architecture and dataset. Cost Efficiency results show clear variability depending on the network architecture used. Each case has a different cost optimal instance and different efficiency gains when parallelizing over multiple GPUs or from switching between instances.

Instances with multiple GPUs in Azure show the best performance in terms of Time per Epoch compared to single GPU instances. Yet, performance gain when adding GPUs is not linear, and the increase in performance does not match the linear increase in instance price. The result is that Azure instances with multiple GPUs are less cost efficient than their single GPU counterparts.

Variable Cost Efficiency between instances and case studies suggests that the computational power of each instance is not fully utilized during training. Underutilization of an instance (a result of the product mix pricing model) implies less cost on the CSP side and higher profitability. These results are consistent with Bakos & Brynjolfsson (1999), who suggest that bundling of information goods is most profitable when the individual components of the bundle are not correlated. Indeed looking at the NC6s v3 instance from Azure and the V100 instance from Paperspace – each with a single V100 GPU – the Paperspace instance has markedly less memory and disk space and achieves higher cost efficiency than NC6s v3 on both case studies.

This research does not support generalization in the choice of an instance, but instead suggests the necessity of carrying out multiple small scale experiments on different instances to make an informed decision given the performance and cost constraints of the data scientist.

4.3. Procedure for cost-optimal instance choice

Wang et al. (2021) from IBM Research interviewed data scientists to understand the level of automation they prefer throughout the phases of data science projects. Expectedly, data scientists prefer to automate as much of their work as possible. Yet one area – decision optimization – particularly stood out where they prefer to have less automation. Instead, data scientists prefer a human in the loop approach whenever a decision has to be made. Within the scope of training neural networks, decisions include which architecture to use, how to tune its hyperparameters, and which instance to run the training on.

Findings of this study also suggest that automating instance choice is not a straightforward task. Performance of neural network training is dependent on many variables, and approaches to predict performance based on GPU hardware specifications and neural network architecture such as Justus et al. (2018) have had limited success and do not translate well to modern architectures.

Based on this study's findings, a procedure is recommended to quickly identify the optimal instance. The procedure is designed to work with a given set of assumptions on both the neural network and the dataset. Changing an assumption, for example, batch size or the size of data input per training sample, requires repeating the procedure again as a different instance could be cost optimal under the new circumstances.

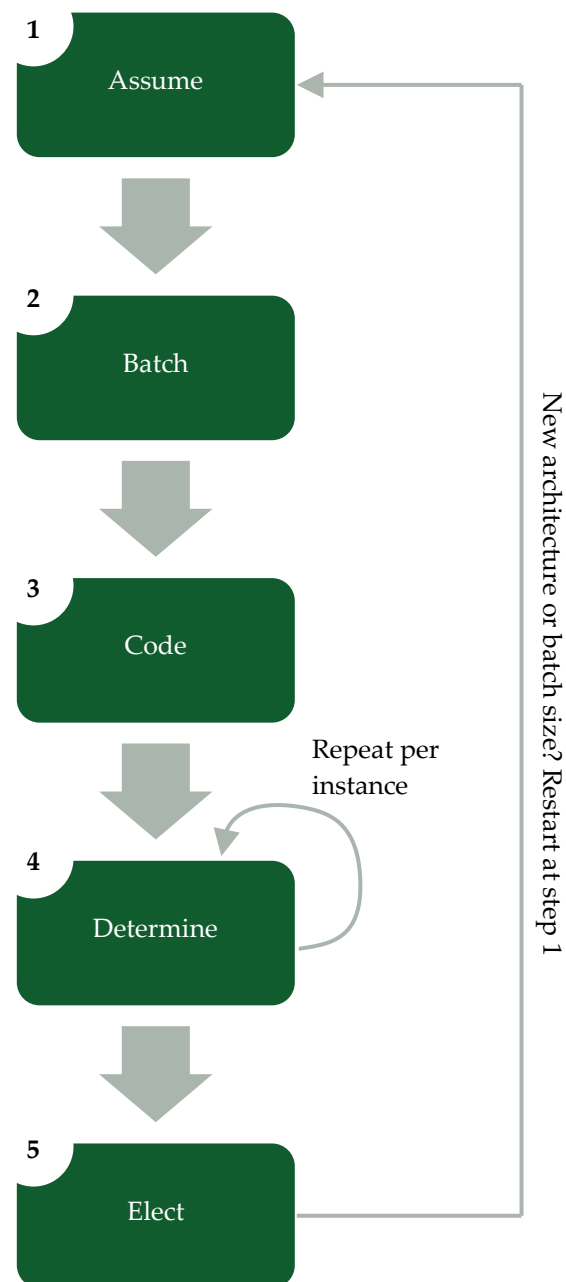


Figure 12. Experimental procedure for instance choice.

The experiment-based procedure is composed of these five steps: (1) setting training assumptions and collecting the list of instances, (2) preparing a reduced size dataset, (3) programming the training code, (4) measuring the performance per instance, and (5) making a cost optimal choice. Figure 12 illustrates the procedure flow.

Step 1 – Assume

- a. Decide which neural network architecture and variant to use for the case at hand. Based on that, determine the needed data input size (e.g. image dimensions).

- b. Specify the batch size, as it is the only hyperparameter that can influence which instances are suitable for training due to memory requirements. Other hyperparameters like learning rate and momentum can be specified as a value or a range. Optimal selection between ranges of hyperparameters is thoroughly covered in deep learning literature and is beyond the scope of this procedure.
- c. Prepare a list of instances under consideration and collect the hourly price per instance. Based on the data input size and the batch size, eliminate instances where GPU memory (per card in case of multi-GPU instances) is smaller than the memory required for one batch of training data.

Step 2 – Batch

- d. Assuming the maximum number of GPUs per instance in the consideration set is equal to g , create a g -batch sized subset of the training data (if batch size is 64, and there is a maximum of four GPUs, the training data subset should include 256 records). The subset serves to quickly measure the Time per Epoch for one batch per GPU – since batches are trained in parallel on multi-GPU instances. The measured time can then be linearly extrapolated to the total dataset size by multiplying the measured time by the expected number of batches divided by g , yielding the total expected Time per Epoch.
- e. Decide on data preprocessing procedures, since these are usually applied during batch training and directly impact Time per Epoch.

Step 3 – Code

- f. Program a boilerplate neural network training code that is instance agnostic. The code should run the training for at least one epoch. Averaging the Time per Epoch from three epochs is recommended to smooth any performance variation dependent on the instance characteristics.

Step 4 – Determine

- g. Deploy the boilerplate code on an instance from the consideration set.
- h. Execute the code on the data batch. Occasionally, the instance has less memory available than is required by the batch size and the code will report an error, despite the earlier screening in step 1. This is due to unforeseen additional memory needed, in which case eliminate the instance from the consideration set.
- i. Determine and record the average training Time per Epoch for the instance. Measuring other metrics, like accuracy, is not relevant at this stage since only a subset of the data is used.
- j. Repeat step 4 for all instances listed in step 1.

Step 5 – Elect

- k. Normalize the Time per Epoch for all instances that ran successfully. Multiply the normalized data by the hourly price to calculate the Cost Efficiency per instance.
- l. Elect the instance with the lowest dollar per hour.
- m. Rerun the training on the elected instance for the full training dataset.
- n. Observe the training results. If, based on the results, a new neural architecture or batch size is desired, repeat the procedure from step 1.

5. Conclusion

This study took a multidisciplinary approach to study the deep learning services provided by cloud service providers (CSPs). The research goal is to help busy data scientists and companies make better decisions when faced with a multitude of deep learning service options. It started with an analysis of value and pricing strategies of general and special purpose CSPs, then moved more specifically to the pricing models of deep learning services and simulated two case studies to measure cost efficiency of training convolutional neural network (CNN) and recurrent neural network (RNN) architectures under realistic conditions.

Research focused on the relationship between CSP pricing strategies and Cost Efficiency of training deep neural networks. First, research into CSP utilization by data scientists showed the usage rate of each CSP deep learning service. Additionally, the challenges data scientists face in their work were researched to understand their value drives. From there, two CSPs were chosen, one general purpose (Azure) and one special purpose (Paperspace).

Azure represented general purpose CSP due to its wide array of GPU instances (including ones with multiple GPUs), its variety of pricing models used for deep learning services, the author's familiarity with its environment, and the presence of Azure data centers in Switzerland. Paperspace was the most used special purpose (notebook-based) CSP according to Kaggle (2020), has been in the deep learning service market for five years, and offers instances with a variety of GPU hardware in its instances.

General purpose CSPs are gradually offering more market- and value-based pricing services to grow their market share and fend off new-comers. The variety of value propositions and pricing models poses a challenge for data scientists to objectively and fairly compare the deep learning services between CSPs from a cost point of view. Looking specifically at Azure, it offers a layered approach of deep learning services, with Machine Learning building on top of Compute Infrastructure, and Bot Service utilizing Cognitive Services. These are designed to serve the needs of individual data scientists, data teams, and businesses willing to purchase packaged intelligent services.

In many traditional businesses, deep learning workload is a small fraction of the total incurred cloud cost. Other cloud infrastructure resources used by businesses to keep their website and mobile apps live and responsive to consumer demand have more predictable workloads due to the business familiarity with its historical workload. General purpose CSPs offer discounted multi-year contracts for these businesses which can create a CSP lock-in, making it difficult for special purpose CSPs to access these kinds of business customers. This presents an opportunity for special purpose CSPs to develop more innovative value models.

Using Paperspace as an example, it is focused on value-based pricing strategy. Paperspace only provides services for deep learning and thus focuses on adding value for data scientists by reducing the time needed to switch between instances. Gradient, their flagship service, uses a feature-based pricing model not observed in Azure. Therefore, Paperspace is unique in using a Pay as you Can delivery model to cater to the needs of different data scientists. On the other hand, Paperspace GPU instances do not include hardware as powerful as Azure, limiting their use in very large workloads. This suggests that Paperspace is positioning itself to be as close to data scientists as possible for relatively smaller workloads or where fast performance is not critical. It can then rely on partnerships with a general purpose CSP when selling solutions to companies with larger deep learning workloads.

The second part of the research introduced two deep learning case studies using public data sources. They were modeled after image and sentiment classification applications based on CNN and RNN architectures, respectively. Code experiments were used to measure the Time per Epoch, and Cost Efficiency of single and multi GPU instances offered by Azure and Paperspace.

Testing the Time per Epoch of CNN and RNN architectures on ten test instances supports the conjectured challenge of optimal instance choice. On Azure, both CNN and RNN gained similar performance from using V100 single GPU instance compared to K80 equivalent. However, CNN training on multi-GPU instances was more accelerated than RNN. On Paperspace, like Azure, the performance gain by using V100 GPU was similar for CNN and RNN. P5000, and to a smaller extent P6000, showed better performance gain for RNN compared to CNN.

Normalizing Time per Epoch allows comparing instances from both CSPs. Paperspace instances were all more cost efficient than Azure's on both CNN and RNN architectures. This was also evident in the two single V100 GPU instances (NC6s v3 on Azure, and V100). One possible reason is the lower price of Paperspace instances compared to Azure combined with less memory and per-instance storage, suggesting that Azure's instances were underutilized when training both cases. The subscription fee of Paperspace should be taken into account in this analysis, but should also be included if doing a thorough total cost comparison, although the subscription fee is likely a small fraction of the instance cost incurred by training real world deep learning cases as observed on DAWNBench (Coleman et al, 2017).

The findings are concluded by offering a procedure for data scientists to quickly elect a cost optimal instance under assumptions of neural network architecture and batch size. The procedure took an experimental approach to mediate the training performance unpredictability challenge. The procedure is designed to work on a subset of the full dataset based on the batch size and the maximum number of GPUs on the consideration list of instances. Working with a limited dataset size ensures quick training, and the results can be linearly extrapolated to calculate the expected Time per Epoch per instance. A cost optimal choice can then be made from all considered instances after normalizing the Time per Epoch and scaling with the instance hourly price.

6. Recommendations

There exist several challenges in calculating the value gained from adopting cloud services. Predictability is a key factor in estimating total cost of ownership when using cloud services, yet the cost and performance deep learning applications in particular have proven challenging to predict. Improving the predictability of deep learning workloads is a necessary prerequisite to accelerate the adoption of AI and allow businesses to realize its value potential.

As the adoption of AI applications continues to grow, the cost of using CSPs by businesses will continue to increase, placing more pressure on CSPs to offer more competitive pricing, thus threatening their profitability. General purpose CSPs should continue to evolve their pricing strategies by offering service bundles and Good to Be value propositions to reduce their comparability with other CSPs and create more value for their customers.

New special purpose CSP market entrants have developed innovative value-based pricing models to attract data scientists, yet general purpose CSPs are catching up by mirroring

special purpose CSPs' notebook-based business model. Special purpose CSPs should continue to innovate and offer customer value-based pricing models by abstracting away the problem of instance choice. Already, new CSPs like JarvisLabs allow a more flexible GPU instance bundle creation. This should be taken a step further by offering pricing by easily measured metrics such as neural network architecture complexity (measured in FLOPs) and the size of the dataset. Abstracting away hardware resources – like in the case of Azure Cognitive Services – for training any deep learning model would provide a clear value for data scientists. They would only have to focus on optimizing their model, knowing that only the necessary resources are utilized during training, likely resulting in a higher market adoption of special purpose CSPs.

The performance of different neural network architectures, software libraries, and hardware has been thoroughly studied from the perspective of predictive accuracy. Accuracy is an important metric for deep learning applications, but can come at a very high cost that is widely variable over different hardware and software implementations. Further research into the cost efficiency of training neural networks within the context of CSP services is necessary. One step towards that is that deep learning researchers should always publish the training time and cost for their experiments along with accuracy results.

The problem of instance choice has been highlighted before and is a function of the interaction of several variables, demanding more research into building training performance prediction tools that can (1) work with modern GPUs, (2) take into account the hardware configuration bundle within an instance, (3) incorporate modern neural network layer types, (4) take dataset size and network hyperparameters into account, and (5) simulate the performance efficiency of using multiple GPUs. The availability of such performance and cost prediction tools would simplify and potentially automate the instance choice problem. Finally, as AI continues to consume resources and create value for businesses, more multidisciplinary research bridging commercial and engineering topics is vital.

References

1. Agmon Ben-Yehuda, O., Ben-Yehuda, M., Schuster, A., & Tsafrir, D. (2013). Deconstructing Amazon EC2 spot instance pricing. *ACM Transactions on Economics and Computation (TEAC)*, 1(3), 1-20.
2. Aho, T., Sievi-Korte, O., Kilamo, T., Yaman, S., & Mikkonen, T. (2020, November). Demystifying Data Science Projects: A Look on the People and Process of Data Science Today. In *International Conference on Product-Focused Software Process Improvement* (pp. 153-167). Springer, Cham.
3. Ahokangas, P., Juntunen, M., & Myllykoski, J. (2014). Cloud computing and transformation of international e-business models. In *A Focused Issue on Building New Competences in Dynamic Environments*. Emerald Group Publishing Limited.
4. Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Hasan, M., Van Essen, B. C., Awwal, A. A. S., & Asari, V. K. (2019). A state-of-the-art survey on deep learning theory and architectures. *Electronics (Switzerland)*, 8(3).
<https://doi.org/10.3390/electronics8030292>
5. Amodei, D., & Hernandez, D. (2018). AI and Compute. *OpenAI Blog*. Retrieved from: <https://openai.com/blog/ai-and-compute/>
6. Araci, D. (2019). Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*.
7. AWS. (2016). 10 Years of AWS. Retrieved from: <https://aws.amazon.com/10year/>
8. AWS. (2021). AWS Ground Station. Retrieved from: <https://aws.amazon.com/ground-station/>
9. Bakos, Y., & Brynjolfsson, E. (1999). Bundling information goods: pricing, profits, and efficiency. *Management Science*, 45(12), 1613–1630.
<https://doi.org/10.1287/mnsc.45.12.1613>
10. Barbedo, J. G. (2018). Factors influencing the use of deep learning for plant disease recognition. *Biosystems engineering*, 172, 84-91.
11. Baur, A. W., Genova, A. C., Bühler, J., & Bick, M. (2014, November). Customer is king? A framework to shift from cost-to value-based pricing in software as a service: the case of business intelligence software. In *Conference on e-Business, e-Services and e-Society* (pp. 1-13). Springer, Berlin, Heidelberg.
12. Bell, S., & Bala, K. (2015). Learning visual similarity for product design with convolutional neural networks. *ACM transactions on graphics (TOG)*, 34(4), 1-10.
13. Bergstra, J., Bastien, F., Breuleux, O., Lamblin, P., Pascanu, R., Delalleau, O., ... & Bengio, Y. (2011). Theano: Deep learning on gpus with python. In *NIPS 2011, BigLearning Workshop, Granada, Spain* (Vol. 3, pp. 1-48). Microtome Publishing.
14. Butz Jr, H. E., & Goodstein, L. D. (1996). Measuring customer value: gaining the strategic advantage. *Organizational dynamics*, 24(3), 63-77.
15. Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems*, 25(6), 599-616.
16. Canalys. (2020). Cloud market share Q4 2019 and full-year 2019. Retrieved from: https://canalys-com-public-prod.s3.eu-west-2.amazonaws.com/static/press_release/2020/Canalys---Cloud-market-share-Q4-2019-and-full-year-2019.pdf
17. Carneiro, T., Da Nóbrega, R. V. M., Nepomuceno, T., Bian, G. B., De Albuquerque, V. H. C., & Reboucas Filho, P. P. (2018). Performance analysis of google colaboratory as a tool for accelerating deep learning applications. *IEEE Access*, 6, 61677-61685.

18. Chowdhury, I. N., Gruber, T., & Zolkiewski, J. (2016). Every cloud has a silver lining — Exploring the dark side of value co-creation in B2B service networks. *Industrial Marketing Management*, 55, 97-109.
19. Coleman, C., Narayanan, D., Kang, D., Zhao, T., Zhang, J., Nardi, L., ... & Zaharia, M. (2017). Dawnbench: An end-to-end deep learning benchmark and competition. *Training*, 100(101), 102.
20. Coleman, C., Zaharia, M., Kang, D., Narayanan, D., Nardi, L., Zhao, T., ... Ré, C. (2019). Analysis of DAWNbench, a Time-to-Accuracy Machine Learning Performance Benchmark. *ACM SIGOPS Operating Systems Review*, 53(1), 14–25. doi:10.1145/3352020.3352024
21. Cui, H., Zhang, H., Ganger, G. R., Gibbons, P. B., & Xing, E. P. (2016, April). Geeps: Scalable deep learning on distributed GPUs with a GPU-specialized parameter server. In *Proceedings of the Eleventh European Conference on Computer Systems* (pp. 1-16).
22. De Toni, D., Milan, G. S., Saciloto, E. B., & Larentis, F. (2017). Pricing strategies and levels and their impact on corporate profitability. *Revista de Administração (São Paulo)*, 52(2), 120-133.
23. Deng, J., Dong, W., Socher, R., Li, L.-J., Kai Li, & Li Fei-Fei. (2009). ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition. doi:10.1109/cvpr.2009.5206848
24. Dertouzos, J., Duncan, E., Kässer, M., Rao, S., & Richte, W. (2020). Making the cloud pay: How industrial companies can accelerate impact from the cloud. *McKinsey Quarterly*. Retrieved from: <https://www.mckinsey.com/industries/advanced-electronics/our-insights/making-the-cloud-pay-how-industrial-companies-can-accelerate-impact-from-the-cloud>
25. FastAI. (2021a). Azure Data Science Virtual Machine. Retrieved from: https://course.fast.ai/start_azure_dsvm
26. FastAI. (2021b). Transfer learning in text. Retrieved from: <https://docs.fast.ai/tutorial.text.html>
27. FastAI. (2021c). Tutorial - Training a model on Imagenette. Retrieved from: <https://docs.fast.ai/tutorial.imagenette.html>
28. FloydHub. (2021). FloydHub Pricing. Retrieved from: <https://www.floydhub.com/pricing>
29. Forrest, W., Gu, M., & Kaplan, J. (2021). Cloud's trillion-dollar prize is up for grabs. *McKinsey Quarterly*. Retrieved from: <https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/clouds-trillion-dollar-prize-is-up-for-grabs>
30. Gong, Y., Jia, Y., Leung, T., Toshev, A., & Ioffe, S. (2013). Deep convolutional ranking for multilabel image annotation. *arXiv preprint arXiv:1312.4894*.
31. Google. (2021). Colaboratory Frequently Asked Questions. Retrieved from: <https://research.google.com/colaboratory/faq.html>
32. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2016.90
33. Hinterhuber, A. (2004). Towards value-based pricing—An integrative framework for decision making. *Industrial marketing management*, 33(8), 765-778.
34. Hinterhuber, A. (2008). Customer value-based pricing strategies: why companies resist. *Journal of business strategy*.
35. Howard, J. (2019). Imagenette dataset. Retrieved from: <https://github.com/fastai/imagenette>.
36. Howard, J., & Gugger, S. (2020). FastAI: A layered API for deep learning. *Information*, 11(2), 108.

37. IBM. (2018). Beyond the hype: A guide to understanding and successfully implementing artificial intelligence within your business. Retrieved from: <https://www.ibm.com/downloads/cas/8ZDXNKQ4>
38. IBM. (2021). NVIDIA GPUs on IBM Cloud servers. Retrieved from: <https://www.ibm.com/nl-en/cloud/gpu>
39. IDC. (2020). Worldwide Spending on Artificial Intelligence Is Expected to Double in Four Years. Retrieved from: <https://www.idc.com/getdoc.jsp?containerId=prUS46794720>
40. Jäätmaa, J. (2010). *Financial aspects of cloud computing business models*. [Unpublished master's thesis]. Aalto University.
41. JarvisLabs. (2021). JarvisCloud. Retrieved from: <https://cloud.jarvislabs.ai/>
42. Jin, H., Wang, X., Wu, S., Di, S., & Shi, X. (2014). Towards optimized fine-grained pricing of IaaS cloud platform. *IEEE Transactions on cloud Computing*, 3(4), 436-448.
43. Justus, D., Brennan, J., Bonner, S., & McGough, A. S. (2018, December). Predicting the computational cost of deep learning models. In *2018 IEEE International Conference on Big Data (Big Data)* (pp. 3873-3882). IEEE.
44. Kaggle. 2020. Kaggle Data Science Survey 2020. Retrieved from: <https://www.kaggle.com/c/kaggle-survey-2020/data>
45. Karpathy, A. (2015). The Unreasonable Effectiveness of Recurrent Neural Networks. Retrieved from: <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>
46. Kim, A., Yang, Y., Lessmann, S., Ma, T., Sung, M. C., & Johnson, J. E. (2020). Can deep learning predict risky retail investors? A case study in financial risk behavior forecasting. *European Journal of Operational Research*, 283(1), 217-234.
47. Kim, D. K., & Chen, T. (2015). Deep neural network for real-time autonomous indoor navigation. *arXiv preprint arXiv:1511.04668*.
48. Klems, M., Nimis, J., & Tai, S. (2008, December). Do clouds compute? a framework for estimating the value of cloud computing. In *Workshop on E-Business* (pp. 110-123). Springer, Berlin, Heidelberg.
49. Klyuchnikov, N., Trofimov, I., Artemova, E., Salnikov, M., Fedorov, M., & Burnaev, E. (2020). NAS-Bench-NLP: neural architecture search benchmark for natural language processing. *arXiv preprint arXiv:2006.07116*.
50. Kovalev, V., Kalinovskiy, A., & Kovalev, S. (2016). Deep learning with theano, torch, caffe, tensorflow, and deeplearning4j: Which one is the best in speed and accuracy?.
51. Lazic, N., Lu, T., Boutilier, C., Ryu, M. K., Wong, E. J., Roy, B., & Imwalle, G. (2018). Data center cooling using model-predictive control.
52. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436-444.
53. Lee, I. (2019). Pricing schemes and profit-maximizing pricing for cloud services. *Journal of Revenue and Pricing Management*, 18(2), 112-122.
54. Li, D., Chen, X., Becchi, M., & Zong, Z. (2016, October). Evaluating the energy efficiency of deep convolutional neural networks on CPUs and GPUs. In *2016 IEEE international conferences on big data and cloud computing (BDCloud)* (pp. 477-484). IEEE.
55. Liozu, S. M. (2017). State of value-based-pricing survey: Perceptions, challenges, and impact. *Journal of Revenue and Pricing Management*, 16(1), 18-29.
56. Liozu, S. M., Hinterhuber, A., Boland, R., & Perelli, S. (2012). The conceptualization of value-based pricing in industrial firms. *Journal of Revenue and Pricing Management*, 11(1), 12-34.
57. Lynn, T., Rosati, P., Lejeune, A., & Emeakaroha, V. (2017, December). A preliminary review of enterprise serverless cloud computing (function-as-a-service) platforms. In *2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)* (pp. 162-169). IEEE.

58. Ma, N., Zhang, X., Zheng, H. T., & Sun, J. (2018). Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 116-131).
59. Maas, A., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011, June). Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies* (pp. 142-150).
60. Malta, E. M., Avila, S., & Borin, E. (2019, December). Exploring the Cost-benefit of AWS EC2 GPU Instances for Deep Learning Applications. In *Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing* (pp. 21-29).
61. Marcel, L. (2018). Introducing Paperspace Gradient. Retrieved from: <https://blog.paperspace.com/gradient/>
62. Marn, M., & Rosiello, R. (1992). Managing Price, Gaining Profit. *Harvard Business Review* (September-October 1992). Retrieved from: <https://hbr.org/1992/09/managing-price-gaining-profit>
63. Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J., & Ghalsasi, A. (2011). Cloud computing—The business perspective. *Decision support systems*, 51(1), 176-189.
64. Merity, S., Keskar, N. S., & Socher, R. (2017). Regularizing and optimizing LSTM language models. *arXiv preprint arXiv:1708.02182*.
65. Meyer, A. N., Fritz, T., Murphy, G. C., & Zimmermann, T. (2014, November). Software developers' perceptions of productivity. In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering* (pp. 19-29).
66. Microsoft. (2021a). Azure AI Platform. Retrieved from: <https://azure.microsoft.com/en-us/overview/ai-platform/>
67. Microsoft. (2021b). Azure Spot Virtual Machines. Retrieved from: <https://azure.microsoft.com/en-us/pricing/spot/>
68. Microsoft. (2021c). GPU optimized virtual machine sizes. Retrieved from: <https://docs.microsoft.com/en-us/azure/virtual-machines/sizes-gpu?context=/azure/virtual-machines/context/context>
69. Microsoft. (2021d). NC-series. <https://docs.microsoft.com/en-us/azure/virtual-machines/nc-series>
70. Microsoft. (2021e). NCv3-series. <https://docs.microsoft.com/en-us/azure/virtual-machines/ncv3-series>
71. Microsoft. (2021d). Use Azure Spot Virtual Machines. Retrieved from: <https://docs.microsoft.com/en-us/azure/virtual-machines/spot-vms>
72. Microsoft. (2021). Linux Virtual Machines Pricing. Retrieved from: <https://azure.microsoft.com/en-us/pricing/details/virtual-machines/linux/>
73. Nickolls, J., & Dally, W. J. (2010). The GPU computing era. *IEEE micro*, 30(2), 56-69.
74. Noble, P. M., & Gruca, T. S. (1999). Industrial pricing: Theory and managerial practice. *Marketing science*, 18(3), 435-454.
75. Nuruzzaman, M., & Hussain, O. K. (2018, October). A survey on chatbot implementation in customer service industry through deep neural networks. In *2018 IEEE 15th International Conference on e-Business Engineering (ICEBE)* (pp. 54-61). IEEE.
76. Oracle. (2021). GPU-Virtual Machines and Bare Metal. Retrieved from: <https://www.oracle.com/cloud/compute/gpu.html>
77. Ozbayoglu, A. M., Gudelek, M. U., & Sezer, O. B. (2020). Deep learning for financial applications: A survey. *Applied Soft Computing*, 106384.
78. P. Mattson, C. Cheng, C. Coleman, G. Diamos, P. Micikevicius, D. Patterson, et al., "Mlperf training benchmark", *arXiv preprint arXiv:1910.01500*, 2019.
79. Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. *arXiv preprint cs/0205070*.

80. Pant, D. R., Neupane, P., Poudel, A., Pokhrel, A. K., & Lama, B. K. (2018, October). Recurrent neural network based bitcoin price prediction by twitter sentiment analysis. In *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)* (pp. 128-132). IEEE.
81. Paperspace. (2016). Paperspace public launch & Paperspace for Teams!. *Paperspace Blog*. Retrieved from: <https://blog.paperspace.com/enterpriselaunch/>
82. Paperspace. (2021). Gradient Pricing. <https://gradient.paperspace.com/pricing>
83. Pérez, F., & Granger, B. E. (2007). IPython: a system for interactive scientific computing. *Computing in science & engineering*, 9(3), 21-29.
84. Priem, R. L. (2007). A consumer perspective on value creation. *Academy of Management Review*, 32(1), 219-235.
85. Qi, X., Luo, Y., Wu, G., Boriboonsomsin, K., & Barth, M. J. (2017, June). Deep reinforcement learning-based vehicle energy efficiency autonomous learning system. In *2017 IEEE Intelligent Vehicles Symposium (IV)* (pp. 1228-1233). IEEE.
86. Qin, Z., Yu, F., Liu, C., & Chen, X. (2018). How convolutional neural network see the world-A survey of convolutional neural network visualization methods. *arXiv preprint arXiv:1804.11191*.
87. Raju, J., & Zhang, Z. (2010). *Smart Pricing: How Google, Priceline, and Leading Businesses Use Pricing Innovation for Profitabilit (paperback)*. Pearson Prentice Hall.
88. Reen, N., Hellström, M., Wikström, K., & Perminova-Harikoski, O. (2017). Towards value-driven strategies in pricing IT solutions. *Journal of Revenue and Pricing Management*, 16(1), 91-105.
89. Ren, Y., & Ji, D. (2017). Neural networks for deceptive opinion spam detection: An empirical study. *Information Sciences*, 385, 213-224.
90. Richardson, P. S., Dick, A. S., & Jain, A. K. (1994). Extrinsic and intrinsic cue effects on perceptions of store brand quality. *Journal of marketing*, 58(4), 28-36.
91. Roloff, E., Diener, M., Carissimi, A., & Navaux, P. O. A. (2012). High Performance Computing in the cloud: Deployment, performance and cost efficiency. *CloudCom 2012 - Proceedings: 2012 4th IEEE International Conference on Cloud Computing Technology and Science*, 371-378. <https://doi.org/10.1109/CloudCom.2012.6427549>
92. Saltan, A., & Smolander, K. (2021). Bridging the state-of-the-art and the state-of-the-practice of SaaS pricing: A multivocal literature review. *Information and Software Technology*, 106510.
93. Sharma, A., & Iyer, G. R. (2011). Are pricing policies an impediment to the success of customer solutions?. *Industrial Marketing Management*, 40(5), 723-729.
94. Shaw, M. (1991). Positioning and price: merging theory, strategy, and tactics. *Hospitality Research Journal*, 15(2), 31-39.
95. Shrestha, A., & Mahmood, A. (2019). Review of deep learning algorithms and architectures. *IEEE Access*, 7, 53040-53065. <https://doi.org/10.1109/ACCESS.2019.2912200>
96. Smith, S. L., Kindermans, P. J., Ying, C., & Le, Q. V. (2017). Don't decay the learning rate, increase the batch size. *ArXiv*, 2017, 1-11.
97. Stanford University Human-Centered Artificial Intelligence (HAI). (2021). THE AI INDEX REPORT 2021. Retrieved from: https://aiindex.stanford.edu/wp-content/uploads/2021/03/2021-AI-Index-Report_Master.pdf
98. Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. *arXiv preprint arXiv:1906.02243*.
99. The Economist. (2020a). The cost of training machines is becoming a problem. Retrieved from: <https://www.economist.com/technology-quarterly/2020/06/11/the-cost-of-training-machines-is-becoming-a-problem>

100. The Economist. (2020b). How corporate IT is entering the multi-cloud. Retrieved from: <https://www.economist.com/business/2020/03/14/how-corporate-it-is-entering-the-multi-cloud>
101. The Economist. (2021). Crypto-miners are probably to blame for the graphics-chip shortage. Retrieved from: <https://www.economist.com/graphic-detail/2021/06/19/crypto-miners-are-probably-to-blame-for-the-graphics-chip-shortage>
102. Trigueros, D. S., Meng, L., & Hartnett, M. (2018). Face recognition: From traditional to deep learning methods. *arXiv preprint arXiv:1811.00116*.
103. Tsao, H. Y., Pitt, L. F., & Berthon, P. (2006). An experimental study of brand signal quality of products in an asymmetric information environment. *Omega*, 34(4), 397-405.
104. Vargas, M. R., De Lima, B. S., & Evsukoff, A. G. (2017, June). Deep learning for stock market prediction from financial news articles. In *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)* (pp. 60-65). IEEE.
105. Villamizar, M., Garcés, O., Ochoa, L., Castro, H., Salamanca, L., Verano, M., ... & Lang, M. (2017). Cost comparison of running web applications in the cloud using monolithic, microservice, and AWS Lambda architectures. *Service Oriented Computing and Applications*, 11(2), 233-247.
106. Voelckner, F. (2006). An empirical comparison of methods for measuring consumers' willingness to pay. *Marketing Letters*, 17(2), 137-149.
107. Wang, D., Liao, Q. V., Zhang, Y., Khurana, U., Samulowitz, H., Park, S., ... & Amini, L. (2021). How Much Automation Does a Data Scientist Want?. *arXiv preprint arXiv:2101.03970*.
108. Wang, H., & Raj, B. (2017). On the origin of deep learning. *arXiv preprint arXiv:1702.07800*.
109. Wang, L., Lin, Z. Q., & Wong, A. (2020). Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. *Scientific Reports*, 10(1), 1-12.
110. Weber, F. D., & Schütte, R. (2019). State-of-the-art and adoption of artificial intelligence in retailing. *Digital Policy, Regulation and Governance*.
111. Weimer, D., Scholz-Reiter, B., & Shpitalni, M. (2016). Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection. *CIRP Annals*, 65(1), 417-420.
112. Wu, C., Buyya, R., & Ramamohanarao, K. (2019). Cloud pricing models: Taxonomy, survey, and interdisciplinary challenges. *ACM Computing Surveys*, 52(6). <https://doi.org/10.1145/3342103>
113. Wu, C., Toosi, A. N., Buyya, R., & Ramamohanarao, K. (2018). Hedonic pricing of cloud computing services. *IEEE Transactions on Cloud Computing*.
114. Xu, A., Liu, Z., Guo, Y., Sinha, V., & Akkiraju, R. (2017, May). A new chatbot for customer service on social media. In *Proceedings of the 2017 CHI conference on human factors in computing systems* (pp. 3506-3510).
115. Yamazaki, M., Kasagi, A., Tabuchi, A., Honda, T., Miwa, M., Fukumoto, N., ... & Nakashima, K. (2019). Yet another accelerated sgd: Resnet-50 training on imagenet in 74.7 seconds. *arXiv preprint arXiv:1903.12650*.
116. Yeung, G., Borowiec, D., Friday, A., Harper, R., & Garraghan, P. (2020). Towards {GPU} Utilization Prediction for Cloud Deep Learning. In *12th {USENIX} Workshop on Hot Topics in Cloud Computing (HotCloud 20)*.
117. Zhang, C., Zhou, P., Li, C., & Liu, L. (2015). A convolutional neural network for leaves recognition using data augmentation. In *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable,*

Autonomic and Secure Computing; Pervasive Intelligence and Computing (pp. 2143-2150). IEEE.

118. Zhou, C., Sun, C., Liu, Z., & Lau, F. (2015). A C-LSTM neural network for text classification. *arXiv preprint arXiv:1511.08630*.
119. Zhou, J., Cao, Y., Wang, X., Li, P., & Xu, W. (2016). Deep recurrent models with fast-forward connections for neural machine translation. *Transactions of the Association for Computational Linguistics*, 4, 371-383.

Bibliography

1. Agmon Ben-Yehuda, O., Ben-Yehuda, M., Schuster, A., & Tsafrir, D. (2013). Deconstructing Amazon EC2 spot instance pricing. *ACM Transactions on Economics and Computation (TEAC)*, 1(3), 1-20.
2. Aho, T., Sievi-Korte, O., Kilamo, T., Yaman, S., & Mikkonen, T. (2020, November). Demystifying Data Science Projects: A Look on the People and Process of Data Science Today. In *International Conference on Product-Focused Software Process Improvement* (pp. 153-167). Springer, Cham.
3. Ahokangas, P., Juntunen, M., & Myllykoski, J. (2014). Cloud computing and transformation of international e-business models. In *A Focused Issue on Building New Competences in Dynamic Environments*. Emerald Group Publishing Limited.
4. Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Hasan, M., Van Essen, B. C., Awwal, A. A. S., & Asari, V. K. (2019). A state-of-the-art survey on deep learning theory and architectures. *Electronics (Switzerland)*, 8(3). <https://doi.org/10.3390/electronics8030292>
5. Amodei, D., & Hernandez, D. (2018). AI and Compute. *OpenAI Blog*. Retrieved from: <https://openai.com/blog/ai-and-compute/>
6. Araci, D. (2019). Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*.
7. AWS. (2016). 10 Years of AWS. Retrieved from: <https://aws.amazon.com/10year/>
8. AWS. (2021). AWS Ground Station. Retrieved from: <https://aws.amazon.com/ground-station/>
9. Bakos, Y., & Brynjolfsson, E. (1999). Bundling information goods: pricing, profits, and efficiency. *Management Science*, 45(12), 1613-1630. <https://doi.org/10.1287/mnsc.45.12.1613>
10. Barbedo, J. G. (2018). Factors influencing the use of deep learning for plant disease recognition. *Biosystems engineering*, 172, 84-91.
11. Baur, A. W., Genova, A. C., Bühler, J., & Bick, M. (2014, November). Customer is king? A framework to shift from cost-to value-based pricing in software as a service: the case of business intelligence software. In *Conference on e-Business, e-Services and e-Society* (pp. 1-13). Springer, Berlin, Heidelberg.
12. Bell, S., & Bala, K. (2015). Learning visual similarity for product design with convolutional neural networks. *ACM transactions on graphics (TOG)*, 34(4), 1-10.
13. Bergstra, J., Bastien, F., Breuleux, O., Lamblin, P., Pascanu, R., Delalleau, O., ... & Bengio, Y. (2011). Theano: Deep learning on gpus with python. In *NIPS 2011, BigLearning Workshop, Granada, Spain* (Vol. 3, pp. 1-48). Microtome Publishing.
14. Borjigin, W., Ota, K., & Dong, M. (2019, May). Dealer: An efficient pricing strategy for deep-learning-as-a-service. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)* (pp. 1-6). IEEE.
15. Butz Jr, H. E., & Goodstein, L. D. (1996). Measuring customer value: gaining the strategic advantage. *Organizational dynamics*, 24(3), 63-77.

16. Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems*, 25(6), 599-616.
17. Canals. (2020). Cloud market share Q4 2019 and full-year 2019. Retrieved from: https://canalys-com-public-prod.s3.eu-west-2.amazonaws.com/static/press_release/2020/Canalys---Cloud-market-share-Q4-2019-and-full-year-2019.pdf
18. Carneiro, T., Da Nóbrega, R. V. M., Nepomuceno, T., Bian, G. B., De Albuquerque, V. H. C., & Reboucas Filho, P. P. (2018). Performance analysis of google colab as a tool for accelerating deep learning applications. *IEEE Access*, 6, 61677-61685.
19. Choi, H. S., Ko, M. S., Medlin, D., & Chen, C. (2018). The effect of intrinsic and extrinsic quality cues of digital video games on sales: An empirical investigation. *Decision Support Systems*, 106, 86-96.
20. Chowdhury, I. N., Gruber, T., & Zolkiewski, J. (2016). Every cloud has a silver lining — Exploring the dark side of value co-creation in B2B service networks. *Industrial Marketing Management*, 55, 97-109.
21. Coleman, C., Narayanan, D., Kang, D., Zhao, T., Zhang, J., Nardi, L., ... & Zaharia, M. (2017). Dawnbench: An end-to-end deep learning benchmark and competition. *Training*, 100(101), 102.
22. Coleman, C., Zaharia, M., Kang, D., Narayanan, D., Nardi, L., Zhao, T., ... Ré, C. (2019). Analysis of DAWN Bench, a Time-to-Accuracy Machine Learning Performance Benchmark. *ACM SIGOPS Operating Systems Review*, 53(1), 14–25. doi:10.1145/3352020.3352024
23. Cui, H., Zhang, H., Ganger, G. R., Gibbons, P. B., & Xing, E. P. (2016, April). Geeps: Scalable deep learning on distributed GPUs with a GPU-specialized parameter server. In *Proceedings of the Eleventh European Conference on Computer Systems* (pp. 1-16).
24. De Toni, D., Milan, G. S., Saciloto, E. B., & Larentis, F. (2017). Pricing strategies and levels and their impact on corporate profitability. *Revista de Administração (São Paulo)*, 52(2), 120-133.
25. Deng, J., Dong, W., Socher, R., Li, L.-J., Kai Li, & Li Fei-Fei. (2009). ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition. doi:10.1109/cvpr.2009.5206848
26. Dertouzos, J., Duncan, E., Kässer, M., Rao, S., & Richte, W. (2020). Making the cloud pay: How industrial companies can accelerate impact from the cloud. *McKinsey Quarterly*. Retrieved from: <https://www.mckinsey.com/industries/advanced-electronics/our-insights/making-the-cloud-pay-how-industrial-companies-can-accelerate-impact-from-the-cloud>
27. FastAI. (2021a). Azure Data Science Virtual Machine. Retrieved from: https://course.fast.ai/start_azure_dsvm
28. FastAI. (2021b). Transfer learning in text. Retrieved from: <https://docs.fast.ai/tutorial.text.html>
29. FastAI. (2021c). Tutorial - Training a model on Imagenette. Retrieved from: <https://docs.fast.ai/tutorial.imagenette.html>
30. FloydHub. (2021). FloydHub Pricing. Retrieved from: <https://www.floydhub.com/pricing>
31. Forrest, W., Gu, M., & Kaplan, J. (2021). Cloud's trillion-dollar prize is up for grabs. *McKinsey Quarterly*. Retrieved from: <https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/clouds-trillion-dollar-prize-is-up-for-grabs>
32. Gong, Y., Jia, Y., Leung, T., Toshev, A., & Ioffe, S. (2013). Deep convolutional ranking for multilabel image annotation. *arXiv preprint arXiv:1312.4894*.

33. Google. (2021). Colaboratory Frequently Asked Questions. Retrieved from: <https://research.google.com/colaboratory/faq.html>
34. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2016.90
35. Hinterhuber, A. (2004). Towards value-based pricing—An integrative framework for decision making. *Industrial marketing management*, 33(8), 765-778.
36. Hinterhuber, A. (2008). Customer value-based pricing strategies: why companies resist. *Journal of business strategy*.
37. Howard, J. (2019). Imagenette dataset. Retrieved from: <https://github.com/fastai/imagenette>.
38. Howard, J., & Gugger, S. (2020). FastAI: A layered API for deep learning. *Information*, 11(2), 108.
39. IBM. (2018). Beyond the hype: A guide to understanding and successfully implementing artificial intelligence within your business. Retrieved from: <https://www.ibm.com/downloads/cas/8ZDXNKO4>
40. IBM. (2021). NVIDIA GPUs on IBM Cloud servers. Retrieved from: <https://www.ibm.com/nl-en/cloud/gpu>
41. IDC. (2020). Worldwide Spending on Artificial Intelligence Is Expected to Double in Four Years. Retrieved from: <https://www.idc.com/getdoc.jsp?containerId=prUS46794720>
42. Jäätmaa, J. (2010). *Financial aspects of cloud computing business models*. [Unpublished master's thesis]. Aalto University.
43. JarvisLabs. (2021). JarvisCloud. Retrieved from: <https://cloud.jarvislabs.ai/>
44. Jin, H., Wang, X., Wu, S., Di, S., & Shi, X. (2014). Towards optimized fine-grained pricing of IaaS cloud platform. *IEEE Transactions on cloud Computing*, 3(4), 436-448.
45. Justus, D., Brennan, J., Bonner, S., & McGough, A. S. (2018, December). Predicting the computational cost of deep learning models. In *2018 IEEE International Conference on Big Data (Big Data)* (pp. 3873-3882). IEEE.
46. Kaggle. 2020. Kaggle Data Science Survey 2020. Retrieved from: <https://www.kaggle.com/c/kaggle-survey-2020/data>
47. Kaplunovich, A., & Yesha, Y. (2017, December). Cloud big data decision support system for machine learning on AWS: Analytics of analytics. In *2017 IEEE International Conference on Big Data (Big Data)* (pp. 3508-3516). IEEE.
48. Karpathy, A. (2015). The Unreasonable Effectiveness of Recurrent Neural Networks. Retrieved from: <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>
49. Kim, A., Yang, Y., Lessmann, S., Ma, T., Sung, M. C., & Johnson, J. E. (2020). Can deep learning predict risky retail investors? A case study in financial risk behavior forecasting. *European Journal of Operational Research*, 283(1), 217-234.
50. Kim, D. K., & Chen, T. (2015). Deep neural network for real-time autonomous indoor navigation. *arXiv preprint arXiv:1511.04668*.
51. Klems, M., Nimis, J., & Tai, S. (2008, December). Do clouds compute? a framework for estimating the value of cloud computing. In *Workshop on E-Business* (pp. 110-123). Springer, Berlin, Heidelberg.
52. Klyuchnikov, N., Trofimov, I., Artemova, E., Salnikov, M., Fedorov, M., & Burnaev, E. (2020). NAS-Bench-NLP: neural architecture search benchmark for natural language processing. *arXiv preprint arXiv:2006.07116*.
53. Kovalev, V., Kalinovskiy, A., & Kovalev, S. (2016). Deep learning with theano, torch, caffe, tensorflow, and deeplearning4j: Which one is the best in speed and accuracy?.
54. Lazic, N., Lu, T., Boutilier, C., Ryu, M. K., Wong, E. J., Roy, B., & Imwalle, G. (2018). Data center cooling using model-predictive control.
55. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436-444.

56. Lee, I. (2019). Pricing schemes and profit-maximizing pricing for cloud services. *Journal of Revenue and Pricing Management*, 18(2), 112-122.
57. Li, D., Chen, X., Becchi, M., & Zong, Z. (2016, October). Evaluating the energy efficiency of deep convolutional neural networks on CPUs and GPUs. In *2016 IEEE international conferences on big data and cloud computing (BDCloud)* (pp. 477-484). IEEE.
58. Liozu, S. M. (2017). State of value-based-pricing survey: Perceptions, challenges, and impact. *Journal of Revenue and Pricing Management*, 16(1), 18-29.
59. Liozu, S. M., Hinterhuber, A., Boland, R., & Perelli, S. (2012). The conceptualization of value-based pricing in industrial firms. *Journal of Revenue and Pricing Management*, 11(1), 12-34.
60. Lynn, T., Rosati, P., Lejeune, A., & Emeakaro, V. (2017, December). A preliminary review of enterprise serverless cloud computing (function-as-a-service) platforms. In *2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)* (pp. 162-169). IEEE.
61. Ma, N., Zhang, X., Zheng, H. T., & Sun, J. (2018). Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 116-131).
62. Maas, A., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011, June). Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies* (pp. 142-150).
63. Malta, E. M., Avila, S., & Borin, E. (2019, December). Exploring the Cost-benefit of AWS EC2 GPU Instances for Deep Learning Applications. In *Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing* (pp. 21-29).
64. Marcel, L. (2018). Introducing Paperspace Gradient. Retrieved from: <https://blog.paperspace.com/gradient/>
65. Marn, M., & Rosiello, R. (1992). Managing Price, Gaining Profit. *Harvard Business Review* (September-October 1992). Retrieved from: <https://hbr.org/1992/09/managing-price-gaining-profit>
66. Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J., & Ghalsasi, A. (2011). Cloud computing—The business perspective. *Decision support systems*, 51(1), 176-189.
67. Merity, S., Keskar, N. S., & Socher, R. (2017). Regularizing and optimizing LSTM language models. *arXiv preprint arXiv:1708.02182*.
68. Meyer, A. N., Fritz, T., Murphy, G. C., & Zimmermann, T. (2014, November). Software developers' perceptions of productivity. In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering* (pp. 19-29).
69. Microsoft. (2021). Linux Virtual Machines Pricing. Retrieved from: <https://azure.microsoft.com/en-us/pricing/details/virtual-machines/linux/>
70. Microsoft. (2021a). Azure AI Platform. Retrieved from: <https://azure.microsoft.com/en-us/overview/ai-platform/>
71. Microsoft. (2021b). Azure Spot Virtual Machines. Retrieved from: <https://azure.microsoft.com/en-us/pricing/spot/>
72. Microsoft. (2021c). GPU optimized virtual machine sizes. Retrieved from: <https://docs.microsoft.com/en-us/azure/virtual-machines/sizes-gpu?context=/azure/virtual-machines/context/context>
73. Microsoft. (2021d). NC-series. <https://docs.microsoft.com/en-us/azure/virtual-machines/nc-series>
74. Microsoft. (2021d). Use Azure Spot Virtual Machines. Retrieved from: <https://docs.microsoft.com/en-us/azure/virtual-machines/spot-vms>
75. Microsoft. (2021e). NCv3-series. <https://docs.microsoft.com/en-us/azure/virtual-machines/ncv3-series>
76. Nickolls, J., & Dally, W. J. (2010). The GPU computing era. *IEEE micro*, 30(2), 56-69.

77. Noble, P. M., & Gruca, T. S. (1999). Industrial pricing: Theory and managerial practice. *Marketing science*, 18(3), 435-454.
78. Nuruzzaman, M., & Hussain, O. K. (2018, October). A survey on chatbot implementation in customer service industry through deep neural networks. In *2018 IEEE 15th International Conference on e-Business Engineering (ICEBE)* (pp. 54-61). IEEE.
79. Oracle. (2021). GPU-Virtual Machines and Bare Metal. Retrieved from: <https://www.oracle.com/cloud/compute/gpu.html>
80. Ozbayoglu, A. M., Gudelek, M. U., & Sezer, O. B. (2020). Deep learning for financial applications: A survey. *Applied Soft Computing*, 106384.
81. P. Mattson, C. Cheng, C. Coleman, G. Diamos, P. Micikevicius, D. Patterson, et al., "Mlperf training benchmark", *arXiv preprint arXiv:1910.01500*, 2019.
82. Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. *arXiv preprint cs/0205070*.
83. Pant, D. R., Neupane, P., Poudel, A., Pokhrel, A. K., & Lama, B. K. (2018, October). Recurrent neural network based bitcoin price prediction by twitter sentiment analysis. In *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)* (pp. 128-132). IEEE.
84. Paperspace. (2016). Paperspace public launch & Paperspace for Teams!. *Paperspace Blog*. Retrieved from: <https://blog.paperspace.com/enterpriselaunch/>
85. Paperspace. (2021). Gradient Pricing. <https://gradient.paperspace.com/pricing>
86. Pérez, F., & Granger, B. E. (2007). IPython: a system for interactive scientific computing. *Computing in science & engineering*, 9(3), 21-29.
87. Priem, R. L. (2007). A consumer perspective on value creation. *Academy of Management Review*, 32(1), 219-235.
88. Qi, X., Luo, Y., Wu, G., Boriboonsomsin, K., & Barth, M. J. (2017, June). Deep reinforcement learning-based vehicle energy efficiency autonomous learning system. In *2017 IEEE Intelligent Vehicles Symposium (IV)* (pp. 1228-1233). IEEE.
89. Qin, Z., Yu, F., Liu, C., & Chen, X. (2018). How convolutional neural network see the world-A survey of convolutional neural network visualization methods. *arXiv preprint arXiv:1804.11191*.
90. Raju, J., & Zhang, Z. (2010). *Smart Pricing: How Google, Priceline, and Leading Businesses Use Pricing Innovation for Profitability* (paperback). Pearson Prentice Hall.
91. Reen, N., Hellström, M., Wikström, K., & Perminova-Harikoski, O. (2017). Towards value-driven strategies in pricing IT solutions. *Journal of Revenue and Pricing Management*, 16(1), 91-105.
92. Ren, Y., & Ji, D. (2017). Neural networks for deceptive opinion spam detection: An empirical study. *Information Sciences*, 385, 213-224.
93. Richardson, P. S., Dick, A. S., & Jain, A. K. (1994). Extrinsic and intrinsic cue effects on perceptions of store brand quality. *Journal of marketing*, 58(4), 28-36.
94. Roloff, E., Diener, M., Carissimi, A., & Navaux, P. O. A. (2012). High Performance Computing in the cloud: Deployment, performance and cost efficiency. *CloudCom 2012 - Proceedings: 2012 4th IEEE International Conference on Cloud Computing Technology and Science*, 371-378. <https://doi.org/10.1109/CloudCom.2012.6427549>
95. Saltan, A., & Smolander, K. (2021). Bridging the state-of-the-art and the state-of-the-practice of SaaS pricing: A multivocal literature review. *Information and Software Technology*, 106510.
96. Sharma, A., & Iyer, G. R. (2011). Are pricing policies an impediment to the success of customer solutions?. *Industrial Marketing Management*, 40(5), 723-729.
97. Shaw, M. (1991). Positioning and price: merging theory, strategy, and tactics. *Hospitality Research Journal*, 15(2), 31-39.

98. Shrestha, A., & Mahmood, A. (2019). Review of deep learning algorithms and architectures. *IEEE Access*, 7, 53040–53065. <https://doi.org/10.1109/ACCESS.2019.2912200>
99. Siggelkow, N. (2007). Persuasion with case studies. *Academy of management journal*, 50(1), 20-24.
100. Smith, S. L., Kindermans, P. J., Ying, C., & Le, Q. V. (2017). Don't decay the learning rate, increase the batch size. *ArXiv*, 2017, 1–11.
101. Stanford University Human-Centered Artificial Intelligence (HAI). (2021). THE AI INDEX REPORT 2021. Retrieved from: https://aiindex.stanford.edu/wp-content/uploads/2021/03/2021-AI-Index-Report_Master.pdf
102. Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. *arXiv preprint arXiv:1906.02243*.
103. The Economist. (2020a). The cost of training machines is becoming a problem. Retrieved from: <https://www.economist.com/technology-quarterly/2020/06/11/the-cost-of-training-machines-is-becoming-a-problem>
104. The Economist. (2020b). How corporate IT is entering the multi-cloud. Retrieved from: <https://www.economist.com/business/2020/03/14/how-corporate-it-is-entering-the-multi-cloud>
105. The Economist. (2021). Crypto-miners are probably to blame for the graphics-chip shortage. Retrieved from: <https://www.economist.com/graphic-detail/2021/06/19/crypto-miners-are-probably-to-blame-for-the-graphics-chip-shortage>
106. Trigueros, D. S., Meng, L., & Hartnett, M. (2018). Face recognition: From traditional to deep learning methods. *arXiv preprint arXiv:1811.00116*.
107. Tsao, H. Y., Pitt, L. F., & Berthon, P. (2006). An experimental study of brand signal quality of products in an asymmetric information environment. *Omega*, 34(4), 397-405.
108. Vargas, M. R., De Lima, B. S., & Evsukoff, A. G. (2017, June). Deep learning for stock market prediction from financial news articles. In *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)* (pp. 60-65). IEEE.
109. Villamizar, M., Garcés, O., Ochoa, L., Castro, H., Salamanca, L., Verano, M., ... & Lang, M. (2017). Cost comparison of running web applications in the cloud using monolithic, microservice, and AWS Lambda architectures. *Service Oriented Computing and Applications*, 11(2), 233-247.
110. Voelckner, F. (2006). An empirical comparison of methods for measuring consumers' willingness to pay. *Marketing Letters*, 17(2), 137-149.
111. Wang, D., Liao, Q. V., Zhang, Y., Khurana, U., Samulowitz, H., Park, S., ... & Amini, L. (2021). How Much Automation Does a Data Scientist Want?. *arXiv preprint arXiv:2101.03970*.
112. Wang, H., & Raj, B. (2017). On the origin of deep learning. *arXiv preprint arXiv:1702.07800*.
113. Wang, L., Lin, Z. Q., & Wong, A. (2020). Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. *Scientific Reports*, 10(1), 1-12.
114. Weber, F. D., & Schütte, R. (2019). State-of-the-art and adoption of artificial intelligence in retailing. *Digital Policy, Regulation and Governance*.
115. Weimer, D., Scholz-Reiter, B., & Shpitalni, M. (2016). Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection. *CIRP Annals*, 65(1), 417-420.
116. Wu, C., Buyya, R., & Ramamohanarao, K. (2019). Cloud pricing models: Taxonomy, survey, and interdisciplinary challenges. *ACM Computing Surveys*, 52(6). <https://doi.org/10.1145/3342103>

117. Wu, C., Toosi, A. N., Buyya, R., & Ramamohanarao, K. (2018). Hedonic pricing of cloud computing services. *IEEE Transactions on Cloud Computing*.
118. Wu, Y., Liu, L., Pu, C., Cao, W., Sahin, S., Wei, W., & Zhang, Q. (2019). A comparative measurement study of deep learning as a service framework. *IEEE Transactions on Services Computing*.
119. Xu, A., Liu, Z., Guo, Y., Sinha, V., & Akkiraju, R. (2017, May). A new chatbot for customer service on social media. In *Proceedings of the 2017 CHI conference on human factors in computing systems* (pp. 3506-3510).
120. Yamazaki, M., Kasagi, A., Tabuchi, A., Honda, T., Miwa, M., Fukumoto, N., ... & Nakashima, K. (2019). Yet another accelerated sgd: Resnet-50 training on imagenet in 74.7 seconds. *arXiv preprint arXiv:1903.12650*.
121. Yeung, G., Borowiec, D., Friday, A., Harper, R., & Garraghan, P. (2020). Towards {GPU} Utilization Prediction for Cloud Deep Learning. In *12th {USENIX} Workshop on Hot Topics in Cloud Computing (HotCloud 20)*.
122. Yin, R. K. (2004). *The case study anthology*. Sage.
123. Zhang, C., Zhou, P., Li, C., & Liu, L. (2015). A convolutional neural network for leaves recognition using data augmentation. In *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing* (pp. 2143-2150). IEEE.
124. Zhou, C., Sun, C., Liu, Z., & Lau, F. (2015). A C-LSTM neural network for text classification. *arXiv preprint arXiv:1511.08630*.
125. Zhou, J., Cao, Y., Wang, X., Li, P., & Xu, W. (2016). Deep recurrent models with fast-forward connections for neural machine translation. *Transactions of the Association for Computational Linguistics*, 4, 371-383.

Appendix

This appendix includes a selection of the pricing pages screenshots of the CSP deep learning services analyzed in this study.

Azure GPU instances (NC series)

<https://azure.microsoft.com/en-us/pricing/details/virtual-machines/linux/>

OS/Software:
CentOS or Ubuntu Linux

Region:
East US

Currency:
US Dollar (\$)

Display pricing by:
Hour

Category: All General purpose Compute optimized Memory optimized Storage optimized **GPU** High performance compute

NC-series

N-series virtual machines are ideal for compute and graphics-intensive workloads, helping customers to fuel innovation through scenarios like high-end remote visualization, deep learning, and predictive analytics. NC-series virtual machines feature the NVIDIA Tesla accelerated platform and these virtual machines do not support the NVIDIA GRID 2.0 technology for graphics and visualization applications. In addition, N-series offers a NC24r configuration that provides a low latency, high-throughput network interface optimized for tightly coupled parallel computing workloads.

Add to estimate	Instance	Core	RAM	Temporary storage	GPU	Pay as you go	1 year reserved (% Savings)	3 year reserved (% Savings)	Spot (% Savings)
+	NC6	6	56 GiB	340 GiB	1X K80	\$0.90/hour	\$0.5733/hour (~36% savings)	\$0.3996/hour (~56% savings)	\$0.1227/hour (~86% savings)
+	NC12	12	112 GiB	680 GiB	2X K80	\$1.80/hour	\$1.1466/hour (~36% savings)	\$0.7991/hour (~56% savings)	\$0.2454/hour (~86% savings)
+	NC24r	24	224 GiB	1,440 GiB	4X K80	\$3.96/hour	\$2.5224/hour (~36% savings)	\$1.7578/hour (~56% savings)	\$0.5398/hour (~86% savings)
+	NC24	24	224 GiB	1,440 GiB	4X K80	\$3.60/hour	\$2.2932/hour (~36% savings)	\$1.5981/hour (~56% savings)	\$0.4907/hour (~86% savings)

Anomaly Detector

<https://azure.microsoft.com/en-us/pricing/details/cognitive-services/anomaly-detector/>

Instance	Features	Price
Free - Web/Container	Univariate anomaly detection	20000 transactions free per month
Standard - Web/Container	Univariate anomaly detection Multivariate anomaly detection ¹	\$0.314 per 1,000 transactions Free

Azure Content Moderator

<https://azure.microsoft.com/en-us/pricing/details/cognitive-services/content-moderator/>

Instance	Transactions Per Second (TPS)	Features	Price
Free	1 TPS	Moderate	5,000 transactions free per month
		Review	5,000 transactions free per month
Standard	10 TPS	Moderate, Review*	0-1M transactions - \$1 per 1,000 transactions 1M-5M transactions - \$0.75 per 1,000 transactions 5M-10M transactions - \$0.60 per 1,000 transactions 10M+ transactions - \$0.40 per 1,000 transactions

Azure Personalizer

<https://azure.microsoft.com/en-us/pricing/details/cognitive-services/personalizer/#pricing>

Instance	Price	Storage quota
Free	50,000 transactions free /month	10 GB
S0	First 1M transactions \$1 per 1000 transactions Next 9M transaction \$0.35 per 1000 transactions Next 90M transaction \$0.20 per 1000 transactions Above 100M transactions \$0.05 per 1000 transactions	10 GB/1M transactions/month

Azure Language Understanding

<https://azure.microsoft.com/en-us/pricing/details/cognitive-services/language-understanding-intelligent-services/>

Instance	Transactions Per Second (TPS) ¹	Features	Price
Free ² Authoring - Web	5 TPS	Text Requests	N/A in selected region
Free ² Prediction - Web/Container	5 TPS	Text Requests	N/A in selected region
Standard - Web/Container	50 TPS	Text Requests	\$1.50 per 1,000 prediction transactions*
		Speech Requests	\$5.50 per 1,000 prediction transactions*

Azure Text Analytics

<https://azure.microsoft.com/en-us/pricing/details/cognitive-services/text-analytics/#pricing>

Instance	Features	Price
Free - Web/Container	Sentiment Analysis (and Opinion Mining) Key Phrase Extraction Language Detection Named Entity Recognition (not available in Container)	5,000 text records free per month
Standard - Web/Container	Sentiment Analysis (and Opinion Mining) Key Phrase Extraction Language Detection Named Entity Recognition (not available in Container)	0-500,000 text records — \$1 per 1,000 text records 0.5M-2.5M text records — \$0.75 per 1,000 text records 2.5M-10.0M text records — \$0.30 per 1,000 text records 10M+ text records — \$0.25 per 1,000 text records

Azure Translator

<https://azure.microsoft.com/en-us/pricing/details/cognitive-services/translator/>

Pay-as-you-go

Instance	Features	Price
S1 - Web/Container	Standard Translation	\$10 per million chars of standard translation
	Text Translation	
	Language Detection	
	Bilingual Dictionary	
	Transliteration	
	Document Translation	\$15 per million chars of document translation
	Custom Translation	
	Text Translation	\$40 per million chars of custom translation
	Document Translation	\$40 per million chars of document translation
	Training	\$10 per million source + target chars of training data (max. \$300 /training)
	Custom model hosting	\$10 per hosted custom translation model per region, per month

Volume discount

Instance	Features	Price
S2 - Web/Container	Standard Translation	\$2,055.001 /month 250M chars per month included Overage: \$8.22 per million chars
	Document Translation	
	Using standard translation	\$15 per million chars of document translation
	Using custom translation	\$40 per million chars of document translation
	Custom Translation	S1 rates apply for custom translation, model training and hosting
S3 - Web/Container	Standard Translation	\$6,000 /month Up to 1B chars per month Overage: \$6 per million chars
	Document Translation	
	Using standard translation	\$15 per million chars of document translation

Azure Speech

<https://azure.microsoft.com/en-us/pricing/details/cognitive-services/speech-services/>

Standard - Web/Container 100 concurrent requests for Base model 20 concurrent requests for Custom model ¹	Speech to Text	Standard ²	\$1 per audio hour
		Custom	\$1.40 per audio hour Endpoint hosting: \$0.0538 per model per hour
		Conversation Transcription Multichannel Audio ^{PREVIEW 4}	\$2.10 per audio hour ⁵
	Text to Speech	Standard	\$4 per 1M characters
		Neural	\$16 per 1M characters ⁶ Long audio creation: \$100 per 1M characters
		Custom	\$6 per 1M characters Endpoint hosting: \$0.0537 per model per hour

Azure Computer Vision

<https://azure.microsoft.com/en-us/pricing/details/cognitive-services/computer-vision/>

Instance	Transactions Per Second (TPS)**	Features	Price
Free - Web/Container	20 per minute		5,000 transactions free per month
S1 - Web/Container	10 TPS	Tag Face GetThumbnail Color Image Type GetAreaOfInterest	0-1M transactions — \$1 per 1,000 transactions 1M-10M transactions — \$0.65 per 1,000 transactions 10M-100M transactions — \$0.60 per 1,000 transactions 100M+ transactions — \$0.40 per 1,000 transactions
		OCR Adult Celebrity Landmark Detect, Objects Brand	0-1M transactions — \$1 per 1,000 transactions 1M-10M transactions — \$0.65 per 1,000 transactions 10M-100M transactions — \$0.60 per 1,000 transactions 100M+ transactions — \$0.40 per 1,000 transactions
		Describe* Read	0-1M transactions — \$1.50 per 1,000 transactions 1M+ transactions — \$0.60 per 1,000 transactions
		Spatial analysis	Free during preview

Azure Face API

<https://azure.microsoft.com/en-us/pricing/details/cognitive-services/face-api/>

Instance	Transactions Per Second (TPS) *	Features	Price
Free - Web/Container	20 transactions per minute	Face Detection Face Verification Face Identification Face Grouping Similar Face Search	30,000 transactions free per month
Standard - Web/Container	10 TPS	Face Detection Face Verification Face Identification Face Grouping Similar Face Search	0-1M transactions - \$1 per 1,000 transactions 1M-5M transactions - \$0.80 per 1,000 transactions 5M-100M transactions - \$0.60 per 1,000 transactions 100M+ transactions - \$0.40 per 1,000 transactions
		Face Storage	\$0.01 per 1,000 faces per month

Azure Bot Services

<https://azure.microsoft.com/en-us/pricing/details/bot-services/>

	Free	\$1
Standard channels	Unlimited messages	Unlimited messages
Premium channels	10,000 messages/month	\$0.50 per 1,000 messages

Azure Machine Learning

<https://azure.microsoft.com/en-us/pricing/details/machine-learning/>

NC-series

Instance	Core	RAM	GPU	Linux VM Price	Machine Learning Service Surcharge	Pay As You Go Total Price	1 year reserved total price	3 year reserved total price
NC6	6	56 GiB	1X K80	\$0.90/hour	\$0/hour	\$0.90/hour	\$0.574/hour ~36% savings	\$0.400/hour ~56% savings
NC12	12	112 GiB	2X K80	\$1.80/hour	\$0/hour	\$1.80/hour	\$1.147/hour ~36% savings	\$0.800/hour ~56% savings
NC24	24	224 GiB	4X K80	\$3.60/hour	\$0/hour	\$3.60/hour	\$2.294/hour ~36% savings	\$1.599/hour ~56% savings

Paperspace Gradient

Individual

Team

Free

Hobbyist & Student

For hobbyists, explorers, and adventurous learners

\$0/mo

+ utilization costs on paid instance types ⓘ

Free* + Low Instance Types

Public Notebooks

5GB Persistent Storage

Community Support

Total Notebooks: 5

Running Notebook limit: 1

*Free instances limited to 6-hours per session though there is no limit to the number of sessions you can run.

G1

Developer

For ML/AI engineers, data scientists, and researchers

\$8/mo

+ utilization costs on paid instance types ⓘ

Free-Mid* Instance Types

Private Notebooks

200GB Persistent Storage

Email Support

Total Notebooks: 10

Running Notebook limit: 5

*Free instances limited to 6-hours per session though there is no limit to the number of sessions you can run.

G2

Professional

For professionals building applications at scale

\$24/mo

+ utilization costs on paid instance types ⓘ

Free-High* Instance Types

Private Notebooks

1TB Persistent Storage

Email Support

Total Notebooks: 50

Running Notebook limit: 10

*Free instances limited to 6-hours per session though there is no limit to the number of sessions you can run.

Paperspace Core

GPU			
<p>P4000</p> <p>\$ 0.51 / hour</p> <p>8 GB GPU</p> <p>30 GB RAM</p> <p>8 vCPU</p>	<p>P5000</p> <p>\$ 0.78 / hour</p> <p>16 GB GPU</p> <p>30 GB RAM</p> <p>8 vCPU</p>	<p>P6000</p> <p>\$ 1.10 / hour</p> <p>24 GB GPU</p> <p>30 GB RAM</p> <p>8 vCPU</p>	<p>V100</p> <p>\$ 2.30 / hour</p> <p>16 GB GPU</p> <p>30 GB RAM</p> <p>8 vCPU</p>