[1*]Sudhir Kumar
[2] Anand Kumar
[3]Priya Pankaj Kumar
[4]Om Prakash
[5]Md. Ansarul Haque

Productive blockchain architecture based on parallel machine learning

Journal of Electrical Systems

Abstract: - Through several effective research and applications, machine learning has proven to have limitless potential. When delving into robust machine learning usage, two essential research questions are how to guarantee that no one tampers with a system's search results and how to prevent other users within the same network environment from easily obtaining our personal information from apps or systems. Regarding privacy and security, this circumstance is comparable to another existing systems of information. An alternative strategy for addressing these two issues is presented by the rise of blockchain technology. It is for this reason that several recent studies have attempted to include machine learning methods or blockchain-based technologies into platforms for machine learning. To demonstrate what the potential of combining blockchain technology with machine learning is demonstrated in this study, where we present a parallel framework for utilizing a metaheuristic algorithm to determine appropriate deep learning hyperparameters in a blockchain environment. The problem is also considered in the suggested framework. reducing communication costs by restricting the quantity of data exchanges between blockchain and miners.

Keywords: blockchain, machine learning, artificial intelligence, memory, computation.

I. Introduction

The field of "artificial intelligence" [1] has had significant advancements over the years, with the public being introduced to a number of successful applications that have improved our quality of life. Undoubtedly, one of the largest significant areas of AI technology research is the so-called machine learning [2], which comprises three learning methods: supervised, unsupervised, and semi-supervised, in that order. All three of these methods can function independently in an "intelligent system"; naturally, they may be coupled in tandem to solve a particular problem. Machine learning, a significant area of AI research, is based on the principle of using labelled and unlabelled input data to determine the proper classification rules for the still unknown information or to predict future events.

Figure 1 An example of a supervised and unsupervised learning technology that uses labeled and unlabeled data to classify unlabeled data.

The primary distinction between supervised and unsupervised learning processes, as seen in Figure 1, is that supervised learning's first input data are labeled data, whereas unsupervised learning's input Data from unsupervised learning have no labels. This means that their learning processes are designed quite differently. because unsupervised learning may categorize unlabeled data with less information than supervised learning when there are no labelled data. However, supervised learning is going to build the classification model for the unlabelled data using the labeled data. For this reason, compared to unsupervised learning, supervised learning often yields higher accuracy rates. Unsupervised learning remains beneficial in situations when obtaining labeled data for a machine learning system is difficult, yet this does not imply that it is no longer useful. Since many contemporary information systems have been equipped with AI and ML technologies, enabling them to offer us intelligent services, a Gartner report [3] noted that over the next five years, AI and related applications will

[1] *Sudhir Kumar: Sershah Engineering College, Sasaram, Rohtas, Bihar-821113, India. Email: sudhirkr.cse@gmail.com
[2] Anand Kumar: Sershah Engineering College, Sasaram, Rohtas, Bihar-821113, India
[3]Priya Pankaj Kumar: Sershah Engineering College, Sasaram, Rohtas, Bihar-821113, India
[4]Om Prakash: Sershah Engineering College, Sasaram, Rohtas, Bihar-821113, India
[5]Md. Ansarul Haque: Sershah Engineering College, Sasaram, Rohtas, Bihar-821113, India

continue to provide new corporate value, with a projected $2.9 trillion in value creation in 2021. Subsequent research [4] revealed that "artificial intelligence" and "machine learning" technologies will have a huge impact on a variety of fields, including artificial creativity, autonomous cars, space exploration, healthcare, and agriculture. These applications make it easy to see how "AI and ML" technologies will be applied to improve the services that information systems offer us. Even if some recent reports [5,6] indicated That doesn't indicate that ML and AI technologies are failing. It implies that rather than utilizing ML and AI technology to It could be simpler to demonstrate the potential of these two technologies if they are applied to simpler activities or systems rather than complex systems or uncharted research areas. Along with identifying practical uses for AI and ML technologies, how to prevent aberrant users from attacking intelligent systems, and how to Safeguarding confidential and delicate information in intelligent systems is a crucial area of research at the moment. The architecture and purpose of blockchain and machine learning technologies differ greatly since they come from two distinct fields. One significant distinction is that while the majority of "blockchain technologies" are made for a peer-to-peer view, which divides the data and computations across multiple computers, the majority of "machine learning technologies" are made for an integrated view, which combines the data and computations from one or more computers. This is the primary cause of the rise of parallel computing as one of the newest trends in research combining machine learning and integrating blockchain technology into a unified data system. Blockchain technology may be able to help intelligent systems address the security and privacy concerns that they are now facing, according to a number of recent researches [7, 8]. facing one another. Five categories of artificial intelligence (AI) are identified by Salah et al. [7] based on work on blockchain technologies for intelligent systems: (1) "decentralized AI applications"; (2) "decentralized AI operations"; (3) "blockchain types for AI applications"; (4) "decentralized infrastructure for AI applications"; and (5) "the function of consensus protocols for AI applications". Later research [8] by Tanwar et al. concentrated on the topic of ML algorithms and "blockchain integration", classifying these integrations into four categories: goal, layer, counter, and smart uses.
The topic of how to integrate ML with blockchain technologies now emerges. The machine learning-based blockchain frameworks are the main topic of this study because we could create an effective method of applying ML to blockchain systems or combining ML with blockchain technologies if we had an appropriate ML framework. Thus, the following might be used to summarize the primary contributions of this study:

1. It starts with a quick overview of the parallel machine learning frameworks that leverage blockchain technology and how to apply them to blockchain systems; this serves as a helpful road map for integrating ML with blockchain.
2. Next, it provides a thorough explanation of the suggested blockchain-based parallel deep learning architecture, with an emphasis on utilizing a metaheuristic approach to find "good" deep learning hyperparameters in order to increase accuracy further. This method depends on the collaboration of all nodes.

Frameworks for parallel machine learning

Many effective methods to accelerate "machine learning" computation have been reported dating before to the 1980s, since parallel computing is a natural and practical solution to minimize the reaction time of "machine learning" computation. beforehand. A crucial area of study has been how to integrate "machine learning" computation into parallel computing habitats since the 1990s. The boundaries between distributed and parallel computing have become less clear in recent research, despite the fact that the definitions of these concepts were very different in early studies [9]. While every processor in distributed computing will make use of shared storage, the primary reason is because each processor (node) in that system has its own memory. However, some recent research. The techniques they advised, for which all node having its self remembrance, are referred to as parallel machine learning algorithms in the machine learning research arena. It should be noted that for the purposes of this study, "parallel" refers to both "distributed" and parallel computing. Additionally, a number of parallel versions of genetic algorithms (GAs) were published in the 1990s due to the intense activity in the genetic algorithm research arena [10, 11]. Cantu-Paz separated parallel genetic algorithms in [10] ´. "separate global single-population master-slave", "single-population fine-grained", and "multiple-population coarse-grained representative categories". In a global single-population master-slave system, the master node manages a global population and assigns computing duties, like crossover operations and selection, to the slaves in order to facilitate the evolution of genetic algorithms. Due to the fact that several slaves may now compute the majority of the GA's operations, response times for GA can be greatly accelerated when compared to single processing nodes. As the Distributed computing systems usually use single-population fine-grained algorithms. The fundamental notion is that chromosomes and Information interaction between neighbouring nodes will be limited, and computing workloads will be distributed across distinct computing nodes. The coarse-grained multi-population since it will split the total population into a number of subpopulations, is also known as the island model GA. A chromosome, or sought information, is sent from one island to another after a predetermined number of generations, with each island having its own subpopulation. Particle Swarm Optimization (PSO) and other metaheuristic algorithms employ these three distinct parallel models. Consequently, Lalwani et al. [12] also made advantage of the parallel

PSOs were divided into these three groups, and the potential of PSO in various distributed computing environments—just like "CPU" vs. "GPU" or PSO topology—were examined. The study [13] divided the parallel ACO into four categories—"cellular, multicolony, master–slave, & parallel independent runs".

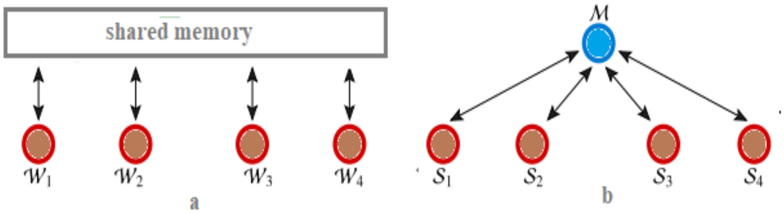

Figure-2(a) and 2(b)

The shared memory allows all of the threads Wi to access the same data, as seen in Fig. 2(a). As Fig. 2(b) shows, the main differences between slave nodes and master nodes are that the former are governed by the latter and that each slave node is unique, even though the slave nodes Si of a master-slave pair would exchange identical data through master node M, for example, through multithreading. Each node in the coarse-grained network transfers its information that was looked for a different one, but each node in the fine-grained network transmits its searched information to a group of adjacent nodes. The attention of researchers has been drawn to many present studies [14,15] that elaborated on construct machine learning algorithms function in distributed computing settings. attention of scientists nowadays, spanning various fields. The fact that the majority of deep learning algorithms require a lot of computation and time is one of the primary causes; thus, how the response time of such an intelligent system is related to the acceleration of these algorithms and is a primary concern of consumers in the modern world. Data parallelism, model parallelism, and layer pipelining are the three categories into which parallel "machine learning" algorithms can be categorized, as seen in Fig. 3, to explain various approaches to building learning models based on the categorization provided in [14, 15]. As illustrated in Fig. 3(a), partitioning the data (e.g., D) is a straightforward method of achieving parallel computing for machine learning. D1, D 2, and D 3) and send them to alternative nodes. Then, in order to integrate these models, each node will build and transmit its model to another node. The so-called minibatch, which has been extensively employed in recent deep learning research, is one exemplary method [14]. According to Fig. 3(b), each node in model parallelism, also known as network parallelism, will obtained a copy of all the information and build distinct portions of the model Mi. Afterwards, a node will receive all of these component components and combine them to create a whole model [15]. The design of a parallel deep learning algorithm can also take into account overlapping computation, as demonstrated in Fig. 3(c) [14]. The compute tasks of the partition may be computation duties of training data elements and other training data elements in one layer and the next layer.

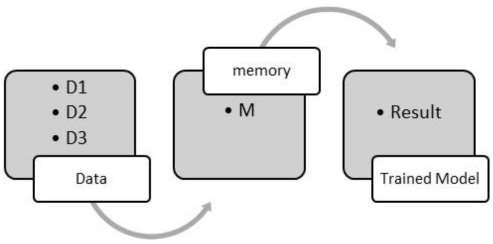

Fig 3a- Data parallelism

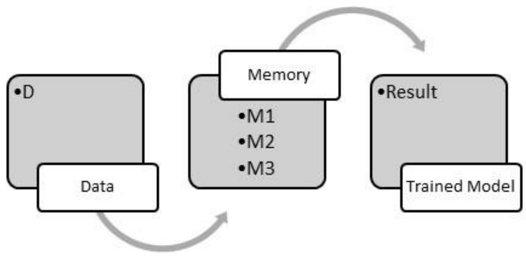

Fig 3b- model parallelism

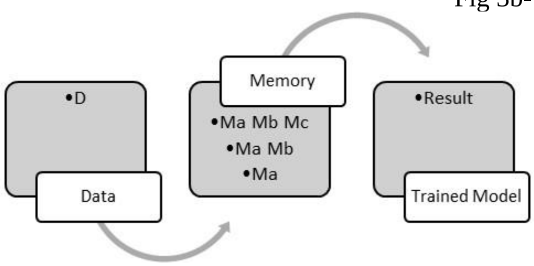

Fig 3c - layer pipelining

Figure 3 a, b, c example showing parallel machine learning strategies from the standpoint of model building

In conclusion, there are numerous classification schemes for deep learning and machine learning algorithms that operate in parallel. Verbraeken et al. [15], for instance, categorized the parallel Topology-based machine learning methods fall into four categories: (i) ensembling, (ii) tree, (iii) parameter server, and (iv) peer-to-peer. Ben-Nun and Hoefler noted in [14] that three key elements are needed to categorize distributed deep learning techniques: training distribution, parameter distributed and communication, and model consistency. We may then categorize these characteristics into four groups to describe how to modify the models, weights, and parameters for parallel deep learning. Asynchronous and parameter servers (1), synchronous and decentralized servers (2), asynchronous and parameter servers (3), and (4) are the current ones. decentralized and stale-synchronous. We can consequently comprehend the design, functionality, and execution of these parallel machine learning and deep learning algorithms from these classifications. function, as well as the various ways that a node communicates information to other nodes. Our observations suggest that the parallel machine learning architecture for blockchain could fit into one or additional parallel groupings. The talk that follows will centre on a few recent researches that combine blockchain technology and parallel machine learning.

Blockchain-based parallel data mining systems

That the blockchain will fundamentally alter some aspects of internet architecture and offer a private and safe means for messages to be sent between information systems Numerous studies have noted the existence of information transmission in some information systems [16–18]. The 2008 release of the Bitcoin white paper marked the beginning of blockchain development. The majority of internet hardware and technologies have advanced considerably since their inception in that year when compared to the 1980s. Blockchain and AI can be combined easily, as Chen et al. demonstrated in [21]. The data D in this system will initially be split up into a collection of subsets that will be given to k parties (see referred to as data holders). According to Fig. 4, Mureddu et al. [25] tried to tackle the smart grid scheduling problem by combining blockchain technology with evolutionary algorithms, as illustrated in Fig. 5. Every miner node mi has a local population (LPOP) built into it. The master ledger on blockchain technology includes a global population (GPOP), and the genetic algorithm will be carried out by each miner using the global and local populations. Upon completion of each miner's evolutionary process, if the newly discovered search result (such as a scheduling solution) outperforms the ones previously preserved in the master ledger, the miner will add it to the blockchain master ledger.

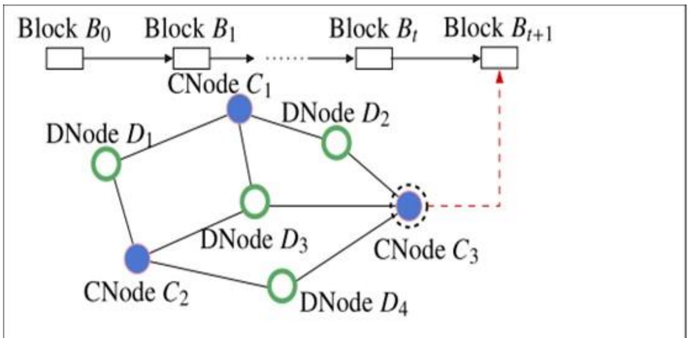

Figure-4 parallel machine learning algorithm for blockchain

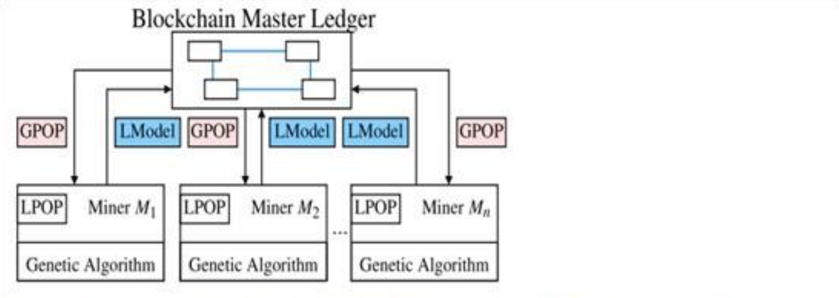

Figure-5 design of blockchain as a shared memory for machine learning algorithm

A parallel deep learning framework for blockchain

It is now feasible for all data owner or computer node to submit a appeal to all other nodes inside a single network to train the entire model jointly thanks to a number of integrated studies that combine blockchain technology and machine learning. Still, Sometimes, in order to create a learning plan and identify a "good trained model" or "applicable hyperparameters," we still require a principal node. In light of this, we introduce PDLKC, a brand-new parallel deep learning framework.

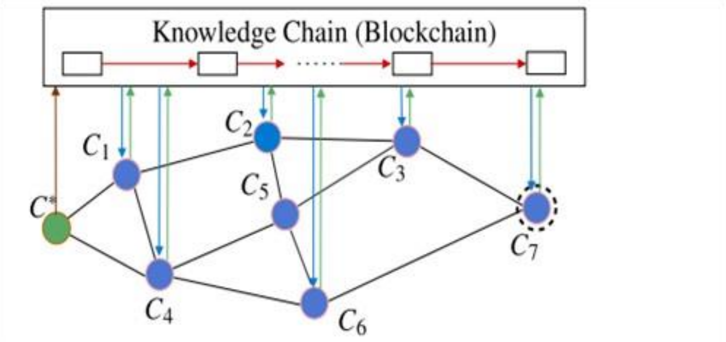

Figure -6 suggested framework's construction in various time slots

According to Fig. 6, the main node C will provide the data and a series of training tasks to “blockchain”, which this study also refers to as the knowledge chain. Following then, during time slot T1, all of the miner nodes will get training jobs and data from the knowledge chain. During time slots T2 and T3, every miner node in the network will try to use knowledge chain to train the models using the hyperparameters from the primary nodes and will choose a training task at random. This implies that in order to train the model, each miner node will use a distinct set of parameters with the same set of data. This manner, it won't have the same parameter value is used by several miner nodes to train the models. The knowledge chain will then be updated by each miner node using its learned model, the outcomes, and the hyperparameter settings.

Simulation results

he DNN classification model is trained using the MNIST dataset in order to assess the performance of the proposed framework (PDLKC). To identify appropriate hyperparameters for DNN, grid search (GS) and simulated annealing (SA) [27] are utilized; that is, for resolving the optimization hyperparameter issue [28]. Python was utilized to implement both GS and the suggested framework, while Ethereum (web3 5.20.0 and solidity 0.5.0) was used as the blockchain. Python 3.6.9 is used to write the applications for the miner nodes, and the simulation is conducted on three PCs running Ubuntu 18.04.5 LTS. These three PCs differ in terms of their CPUs, GPUs, RAM capacities, and even TensorFlow versions. The hyperparameter settings for the grid search are as follows: The batch size, number of hidden layers, number of neurons in each hidden layer, number of epochs, and number of neurons in each hidden layer are all set to 16, 96, 176, and 256. The learning rate is set to 0.0001, 0.0005, 0.001, 0.005, and 0.01; The hyperparameter parameters for SA are as follows: The number in [0.0001, 0.01] is the learning rate; a number in [1, 9] for the number of hidden layers, a number in [16, 512] for the batch size, and a number in [16, 512] for the number of neurons in each hidden layer; additionally the total number of eras to ten. All of the parameter settings for the suggested framework are equal as for SA, with the exception that all miner node will train a prototype using the outcomes si that it obtained from the blockchain before training three more models using s′i, s′′i, and s′′′i, which are transferred from si, s′i, and s′′i, respectively.

Conclusion

We proposed a straightforward yet practical parallel deep learning system for blockchain environments in this study. The characteristics of blockchain are carried over into the suggested architecture, and it can because the majority of miner nodes in the knowledge chain must validate the expert model, offer a secure method of obtaining it. Nevertheless, the suggested framework also faces the same research difficulties that have also been encountered by other parallel machine frameworks in a blockchain environment. These difficulties are (1) extra discussion costs for data, models, and parameter transfers; and (2) needless waiting for coordination and discussion. In the future, we will try to come up with finer solutions to the aforementioned unresolved problems in order to improve the functionality of the suggested framework even more.

Reference

[1] S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, third ed., Prentice Hall Press, USA, 2009.
[2] E. Alpaydin, Introduction to Machine Learning, second ed., The MIT Press, 2010.
[3] Gartner, Gartner says AI augmentation will create $2.9 trillion of business value in 2021, 2019, [Online]. Available: https://www.gartne r.com/en/newsroom/press-releases/2019-08-05-gartner-says-ai-augment ation-will-create-2point9-trillion-of-business-value-in-2021.
[4] A. Johari, AI applications: Top 10 real world artificial intelligence applications, 2020, [Online]. Available: https://www.edureka.co/blog/a rtificial-intelligence-applications/.

[5] G. Marcus, DeepMind's losses and the future of artificial intelligence, 2019, [Online]. Available: https://www.wired.com/story/deepminds-los ses-future-artificial-intelligence/.
[6] L. Plummer, Available: https://www.wired.co.uk/article/ib m-watson-supercomputer-profit.
[7] K. Salah, M.H.U. Rehman, N. Nizamuddin, A. Al-Fuqaha, Blockchain for AI: Review and open research challenges, IEEE Access 7 (2019) 10127–10149.
[8] S. Tanwar, Q. Bhatia, P. Patel, A. Kumari, P.K. Singh, W. Hong, Machine learning adoption in blockchain-based smart applications: The challenges, and a way forward, IEEE Access 8 (2020) 474–488.
[9] Wikipedia, Distributed computing, 2020, [Online]. Available: https://en.wikipedia.org/wiki/Distributed_computing.
[10] E. Cantú-Paz, A survey of parallel genetic algorithms, Calc. Paralleles Reseaux Syst. Repartis 10 (2) (1998) 141–171.
[11] E. Alba, J.M. Troya, A survey of parallel distributed genetic algorithms, Complexity 4 (4) (1999) 31–52.
[12] S. Lalwani, H. Sharma, S.C. Satapathy, K. Deep, J.C. Bansal, A survey on parallel particle swarm optimization algorithms, Arab. J. Sci. Eng. 44 (4) (2019) 2899–2923.
[13] M. Pedemonte, S. Nesmachnow, H. Cancela, A survey on parallel ant colony optimization, Appl. Soft Comput. 11 (8) (2011) 5181–5197.
[14] T. Ben-Nun, T. Hoefler, Demystifying parallel and distributed deep learning: An in-depth concurrency analysis, ACM Comput. Surv. 52 (4) (2019) 1–43.
[15] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, J.S. Rellermeyer, A survey on distributed machine learning, ACM Comput. Surv. 53 (2) (2020) 1–33.
[16] J.-H. Lee, M. Pilkington, How the blockchain revolution will reshape the consumer electronics industry, IEEE Consum. Electron. Mag. 6 (3) (2017) 19–23.
[17] Y. Liu, F.R. Yu, X. Li, H. Ji, V.C. Leung, Blockchain and machine learning for communications and networking systems, IEEE Commun. Surv. Tutor. 22 (2) (2020) 1392–1431.
[18] J. Polge, J. Robert, Y. Le Traon, Permissioned blockchain frameworks in the industry: A comparison, ICT Express 7 (2) (2021) 229–233. W. Gao, C. Su,
[19] Analysis of earnings forecast of blockchain financial products based on particle swarm optimization, J. Comput. Appl. Math. 372 (2020) 1–14.
[20] J. Weng, J. Weng, J. Zhang, M. Li, Y. Zhang, W. Luo, DeepChain: Auditable and privacy-preserving deep learning with blockchain-based incentive, IEEE Trans. Dependable Secure Comput. (2020) 1–18, http://dx.doi.org/10.1109/TDSC.2019.2952332, In Press.
[21] X. Chen, J. Ji, C. Luo, W. Liao, P. Li, When machine learning meets blockchain: A decentralized, privacy-preserving and secure design, in: Proceeding of IEEE International Conference on Big Data Big Data, 2018, pp. 1178–1187.
[22] Y. Lu, X. Huang, Y. Dai, S. Maharjan, Y. Zhang, Blockchain and federated learning for privacy-preserved data sharing in industrial IoT, IEEE Trans. Ind. Inf. 16 (6) (2019) 4177–4186.
[23] H. Kim, J. Park, M. Bennis, S.-L. Kim, Blockchained ondevice federated learning, IEEE Commun. Lett. 24 (6) (2019) 1279–1283.
[24] N.Q. Hieu, T.T. Anh, N.C. Luong, D. Niyato, D.I. Kim, E. Elmroth, Resource management for blockchain-enabled federated learning: A deep reinforcement learning approach, 2020, pp. 1–5, arXiv preprint arXiv:2004.04104.
[25] M. Mureddu, E. Ghiani, F. Pilo, Smart grid optimization with blockchain based decentralized genetic algorithm, in: Proceeding of IEEE Power & Energy Society General Meeting PESGM, 2020, pp. 1–5.
[26] M.A. Ferrag, L. Maglaras, Deepcoin: A novel deep learning and blockchain-based energy exchange framework for smart grids, IEEE Trans. Eng. Manage. 67 (4) (2019) 1285–1297.
[27] C.-W. Tsai, C.-H. Hsia, S.-J. Yang, S.-J. Liu, Z.-Y. Fang, Optimizing hyperparameters of deep learning in predicting bus passengers based on simulated annealing, Appl. Soft Comput. 88 (2020) 1–9.
[28] M. Feurer, J.T. Springenberg, F. Hutter, Initializing bayesian hyperparameter optimization via meta-learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2015, pp. 1128–1135.