Black Magic in Deep Learning: How Human Skill Impacts Network Training

Kanav Anand[1]
anandkanav92@gmail.com
Ziqi Wang[1]
z.wang-8@tudelft.nl
Marco Loog[12]
M.Loog@tudelft.nl
Jan van Gemert[1]
j.c.vangemert@tudelft.nl

[1] Delft University of Technology, Delft, The Netherlands
[2] University of Copenhagen Copenhagen, Denmark

Abstract

How does a user's prior experience with deep learning impact accuracy? We present an initial study based on 31 participants with different levels of experience. Their task is to perform hyperparameter optimization for a given deep learning architecture. The results show a strong positive correlation between the participant's experience and the final performance. They additionally indicate that an experienced participant finds better solutions using fewer resources on average. The data suggests furthermore that participants with no prior experience follow random strategies in their pursuit of optimal hyperparameters. Our study investigates the subjective human factor in comparisons of state of the art results and scientific reproducibility in deep learning.

1 Introduction

The popularity of deep learning in various fields such as image recognition [9, 19], speech [11, 30], bioinformatics [21, 24], question answering [3] *etc*. stems from the seemingly favorable trade-off between the recognition accuracy and their optimization burden. LeCun *et al*. [20] attribute their success to feature representation learning as opposed to using hand-engineered features. While deep networks learn features, the hand engineering has shifted to the design and optimization of the networks themselves. In this paper we investigate the influence of human skill in the hand engineering of deep neural network training.

Arguably, one reason for why neural networks were less popular in the past is that compared to 'shallow' learners such as for example LDA [10], SVM [4], *k*NN [5], Naive-Bayes [29], *etc*., deep networks have many more hyperparameters [35] such as the number of layers, number of neurons per layer, the optimizer, optimizer properties, number of epochs, batch size, type of initialization, learning rate, learning rate scheduler, *etc*. A hyperparameter has to be set before training the deep network and setting these parameters can be difficult [32], yet, the excellent results of deep networks [20] as revealed by huge datasets [6] with fast compute [19] offer a compelling reason to use deep learning approaches in practice, despite the difficulty of setting many of those parameters.

Hyperparameters are essential to good performance as many learning algorithms are critically sensitive to hyperparameter settings [12, 18, 28]. The same learning algorithm will have different optimal hyperparameter configurations for different tasks [16] and optimal configurations for one dataset do not necessarily translate to others [34]. The existing state of the art can be improved by reproducing the work with a better analysis of hyperparameter sensitivity [14], and several supposedly novel models in NLP [25] and in GANs [23] were found to perform similarly to existing models, once hyperparameters were sufficiently tuned. These results show that hyperparameters are essential for reproducing existing work, evaluating model sensitivity, and making comparisons between models.

Finding the best hyperparameters is something that can be done automatically by autoML [2, 7, 15, 17] or Neural Architecture Search [8, 22, 27, 36]. Yet, in practice, such methods are not widely used by deep learning researchers. One reason could be that automatic methods are still under active research and not yet ready for consumption. Another reason could be that good tuning adds a significant computational burden [23, 25]. Besides, automated tuning comes with its own set of hyperparameters and, in part, shifts the hyperparameter problem. Thus, in current practice, the hyperparameters are usually set by the human designer of the deep learning models. In fact, it is widely believed that hyperparameter optimization is a task reserved for experts [31, 32], as the final performance of a deep learning model is *assumed* to be highly correlated with background knowledge of the person tuning the hyperparameters. The validation of this claim is one of the main goals of our research. The extraordinary skill of a human expert to tune hyperparameters is what we here informally refer to as "black magic" in deep learning.

1.1 Contributions

Broadly speaking, we investigate how human skill impacts network training. More specifically, we offer the following contributions. 1. We conduct a user study where participants with a variety of experience in deep learning perform hyperparameter optimization in a controlled setting.[1] 2. We investigate how deep learning experience correlates with model accuracy and tuning efficiency. 3. We investigate human hyperparameter search strategies. 4. We provide recommendations for reproducibility, sensitivity analysis, and model comparisons.

2 Experimental Setup

Our experiment is designed to measure and analyze human skill in hyperparameter optimization. All other variations have identical settings. Each participant has the exact same task, model, time limitation, GPU, and even the same random seed. Our participants tune hyperparameters of a deep learning architecture on a given task in a user-interface mimicking a realistic setting while allowing us to record measurements.

2.1 Deep Learning Setup

The deep learning experimental setup includes: the task, the model and the selection of hyperparameters.

[1] The research carried out has been approved by TU Delft's Human Research Ethics Committee: https://www.tudelft.nl/en/about-tu-delft/strategy/integrity-policy/human-research-ethics/.

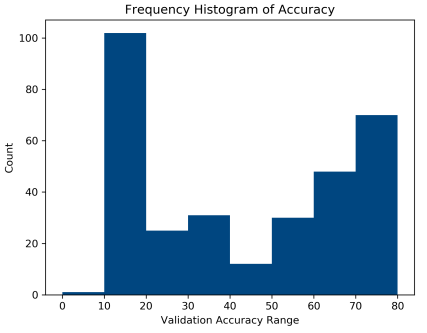


Figure 1: Accuracy histogram over 320 random hyperparameter settings. Their settings matter.

Type	Hyperparameter	Default value
Mandatory	Epochs	-
	Batch size	-
	Loss function	-
	Optimizer	-
Optional	Learning rate	0.001
	Weight decay	0
	Momentum	0
	Rho	0.9
	Lambda	0.75
	Alpha	0.99
	Epsilon	0.00001
	Learning rate decay	0
	Initial accumulator value	0
	Beta1	0.9
	Beta2	0.999

Table 1: The hyperparameters available to participants in our study.

Deep learning task. The choice for the task is determined by the size, difficulty, and realism of the considered dataset. Large datasets take long to train, which limits the number of hyperparameters we can measure. Also, if the dataset is not challenging, it would be relatively easy to achieve a good final performance which limits the variance in the final performance of the model. Taking size and difficulty into account, while staying close to a somewhat realistic setting, we decided on an image classification task on a subset of ImageNet [6] which is called *Imagenette* [1]. To prevent using dataset specific knowledge we did not reveal the dataset name to participants. We only revealed the image classification task and we shared the dataset statistics: 10 classes, 13,000 training images, 500 validation images, and 500 test images

Deep learning model. The model should be well-suited for image classification, have variation in hyperparameter settings, and be somewhat realistic. In addition, it should be relatively fast to train so that a participant can run a reasonable amount of experiments in a reasonable amount of time. We selected *Squeezenet* [13] as it is efficient to train and achieves a reasonable accuracy compared to more complex networks. To prevent exploiting model-specific knowledge, we did not share the network design with the participants.

Hyperparameters. We give participants access to 15 common hyperparameters. Four parameters are mandatory: number of epochs, batch size, loss function, and optimizer. We preset the other 11 optional hyperparameters with their commonly used default values. In Table 1, we show the list of hyperparameters. Please refer to the supplementary material for their full description. Note that none of the hyperparameters under participants control influenced the random seed, as we keep any randomness such as weight initialization and sample shuffling exactly the same for each participant. For 320 random hyperparameter settings, the average random accuracy is 41.8 ± 24.3, where Figure 1 demonstrate that hyperparameters are responsible for ample accuracy variance for this task. Without such variance there may be little differences in human accuracy which would make it difficult to analyse skill.

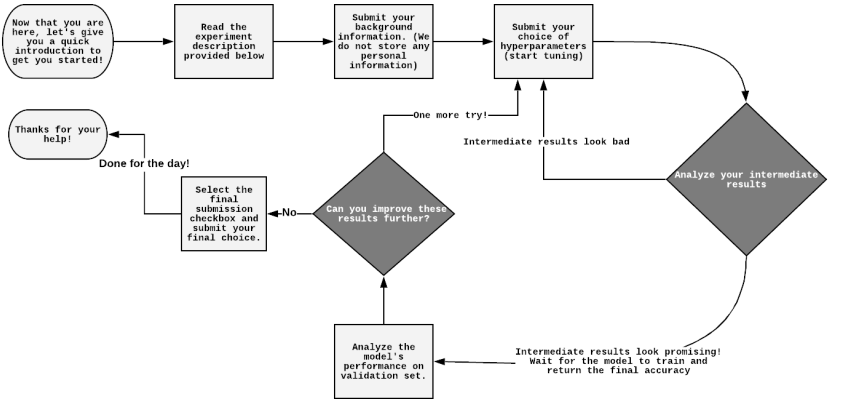


Figure 2: The flow diagram of the user study. The participant starts by entering their information. Next, submit the values for hyperparameters and evaluate intermediate training results. If the training is finished, the participant can decide whether to submit a new configuration for hyperparameter or end the experiment. It can be repeated until the time limit of 120 minutes is reached.

2.2 Participants' Experimental Setup

For managing participants we need: a user-interface, a detailed task description, and define what to measure.

User-interface. We simulate a realistic hyperparameter optimization setting, while providing a controlled environment. We designed a web interface to let participants submit their choice of hyperparameters, view their submission history with validation accuracy and view the intermediate training results with an option for early stopping. Few preliminary tries were done (by the participants not included in result dataset) to test and verify the design and hyperparameter optimization process. By using a web server we collect all the data for analysis. We make all data and source code available[2].

Participant's task. The task given to participants is to find the optimal set of hyperparameters, *i.e.*, those maximizing classification accuracy on the test set. After submitting a choice of hyperparameters, the deep learning model is trained in the background using these parameters. While the model is training, the participant can view the intermediate batch loss and epoch loss in real time. The participant has an option to cancel training if the intermediate results do not look promising. As there is an upper limit of 120 minutes to how much time a participant can use on the optimization of the model, early stopping enables them to try more hyperparameter configurations. After training the model is finished, the accuracy on a validation set is provided to the participant. Participants are encouraged to add optional comments to each choice of hyperparameters. The experiment ends when the participant decides that the model has reached its maximum accuracy or if the time limit of the experiment is reached (120 minutes). The flow diagram of the user study is depicted in Figure 2.

[2] https://github.com/anandkanav92/htune

Measurements per participant. As an indication for the degree of expertise a participant has, we record the number of months of deep learning experience. During deep model training, we record all the hyperparameter combinations tried by the participant, together with the corresponding accuracy on the validation set, for as many epochs as the participant chooses to train. The experiment ends by a participant submitting their final choice of hyperparameters. This optimal hyperparameter configuration is then trained ten times on the combined training and the validation set after which the accuracy on the independent test set is recorded. Each of the 10 repeats have a different random seed, while the seeds are the same for each participant.

2.3 Selection of Participants

The participants were selected based on their educational background and their prior experience in training deep learning models. The participants with no prior experience comprised of people recruited from different specialisations using poster ads and email groups. Experienced candidates were invited through our deep learning course provided to master students and researchers.

3 Results

We collected 463 different hyperparameter combinations from 31 participants. The prior deep learning experience for these participants is well distributed as shown in Figure 3. For the final selected hyperparameters the average classification accuracy is 55.9 ± 26.3.

For ease of analysis we divide participants into groups based on experience. The *Novice* group contains 8 participants with no experience in deep learning, the 12 participants in the *medium* group have less than nine months of experience and the 11 participants in the *expert* group has more than nine months experience.

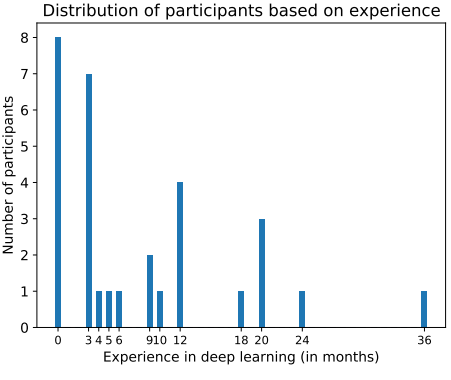


Figure 3: A broad range of deep learning experience in the 31 participants of our study.

3.1 Relation between Experience and Accuracy

Figure 4 depicts the relationship between final accuracy and deep learning experience per participant. As the experience increases, the final accuracy tends to increase, which is supported by the strong positive Spearman [33] rank order correlation coefficient of 0.60 with a p-value smaller than 0.001. Additionally, we compared the variance of the accuracy distributions of *Novice*, *medium*, *expert* groups using Levene's statistical test [26]. We use the Levene test because experience and accuracy are not normally distributed. The test values

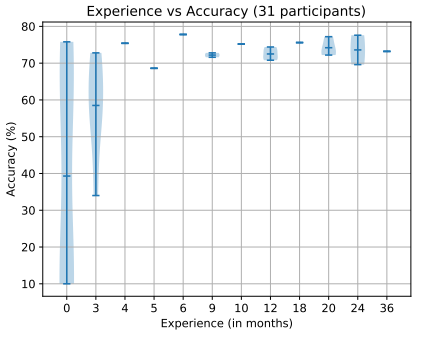


Figure 4: Final accuracy distribution over all participants.

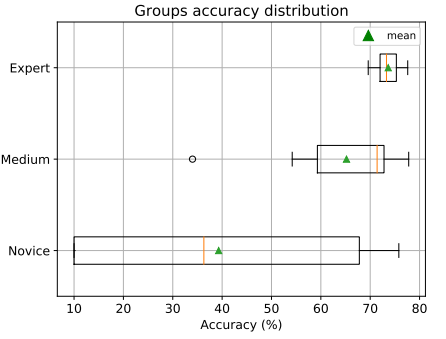
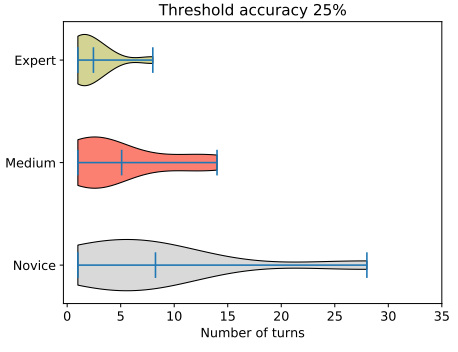


Figure 5: Final accuracy per group boxplot.



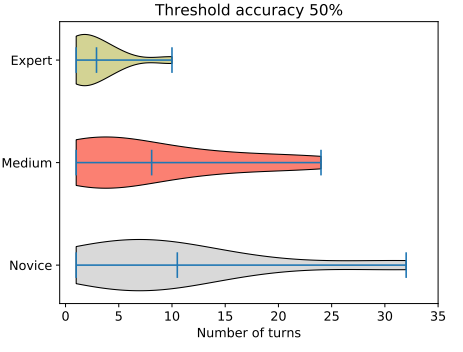


Figure 6: Number of hyperparameter configurations required to achieve a threshold accuracy of 25%, 50% for different experience groups. The violin plot shown above depicts the probability density distribution of the number of turns taken by the participants in each group. The mean value of each group is marked for reference. More experienced participants reach the threshold faster.

presented in Table 2 show all groups significantly differ from each other ($p < 0.05$), where the difference is smallest between *medium* and *expert* and the largest between *Novice* and *expert*, which is in line with the accuracy statistics per group shown in Figure 5.

We further analyze the effect of deep learning experience on the training process. In Figure 6, we show how many tries are used to reach a certain threshold accuracy for the *novice*, *medium*, *expert* groups for final accuracy thresholds. Experts reach the threshold quicker. Furthermore, we show the average accuracy of each group after a number of tries in Figure 7. We can conclude that more experienced participants not only achieve a better accuracy, they also arrive at that better score more efficiently.

3.2 Difference in Strategies

We investigate why more experienced users achieve a higher accuracy in fewer iterations.

Levene's statistical test		
Groups	**Test Statistic**	**p-value**
Novice vs Medium	8.40	0.01
Novice vs Expert	14.338	0.001
Medium vs Expert	5.52	0.029

Table 2: All groups significantly differ from each other ($p < 0.05$); *medium* and *expert* the least and *Novice* and *expert* the most.

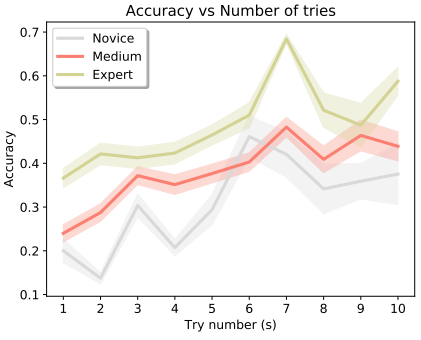


Figure 7: Experts need fewer tries to get better accuracy. The shaded region is standard error.

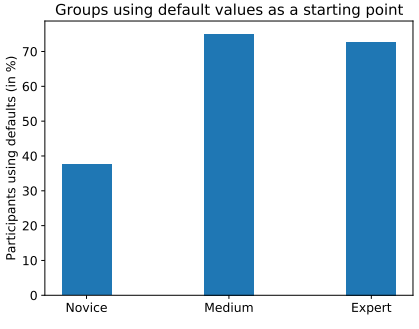


Figure 8: Participants submitting their initial hyperparameter configuration using all default values.

ID	Comment
1	It is just a guess.
2	It is a suggested default value.
3	It is the value that has worked well for me in the past.
4	It is the value I learnt from previous submissions.
5	Other

Table 3: Predefined comments used in user study.

Use of suggested default values. We offer mandatory and optional hyperparameters, as shown in Table 1, where the optional hyperparameters are preset to their default values. Figure 8 shows the number of participants in each group using these default values as the starting point. A large majority in the *medium* or *expert* groups begin with all optional hyperparameter values set to their suggested default values and subsequently build on them. In contrast, *novice* users directly explore the optional values. Using defaults for optional parameters does not necessarily lead to an optimal hyperparameter configuration, however, all participants who started with defaults achieved a final performance greater than 50%.

Analysis of comments. Participants were encouraged to leave comments explaining the reasoning behind choosing a specific value of a hyperparameter. In a bid to gather maximum comments, we let users choose from predefined comments shown in Table 3. Figure 9 shows the distribution of comments for each group of *novice*, *medium*, or *expert* participants. We noticed that there was confusion between 'past experience' and 'learned from previous sub-

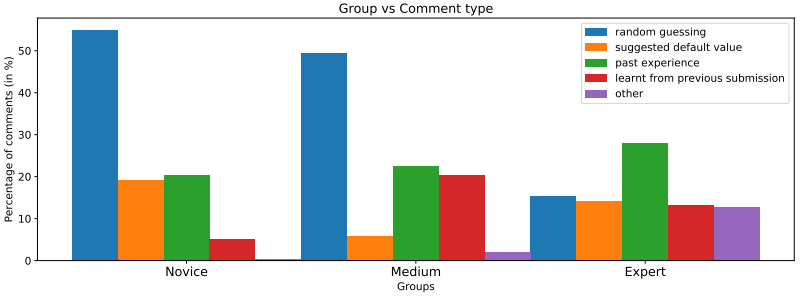


Figure 9: The distribution of comments for the groups of *novice*, *medium*, *expert* participants. Inexperienced users rely more on random guessing (blue).

mission' as 22% of hyper parameter values used by *novice* participants were based on their prior experience in deep learning. As this confusion may also effect other groups, we refrain from drawing hard conclusions based on the observed increase in the use of the comment 'past experience' for more experienced participants. For *novice* participants, the majority is based on random guessing. Random guessing was found to be strongly negatively correlated with the increasing experience. We used Spearman rank-order correlation, and the value was found to be -0.58 with a p-value smaller than 0.001. As the amount of experience increases, the results show a decrease in random guessing.

4 Discussion and Conclusion

We identify main limitations to this study, draw conclusions, and make recommendations.

4.1 Main Limitations

Limited data. We have a fairly restricted number of 31 participants. Collecting more data and inviting more participants in the user study will make the result and conclusions more robust to potential outliers. In addition, it can of course provide better insight into the process of hyperparameter optimization, generalize our findings over a broader audience, and give us the possibility to test more refined hypotheses.

Stratified experience groups. Currently, the three participant groups that we used in our analysis, *i.e.*, novice, medium, and expert, were identified based on the amount of experience, as measured months, they had. It may be of interest, of course, to consider information different from experience to stratify participants in different groups. Maybe the amount of programming experience or the amount of machine learning experience correlates better with performance achievements. What should maybe also be considered, however, is the way to measure something like experience. Rather than using a measure like 'months of experience,' one can also resort, for instance, to often used self-evaluations, in which every participant decided for themselves which level they have. In more extensive experiments, it would definitely be of interest to collect such additional meta-data.

Only one deep learning setting This study focuses only on an image recognition task with a single model and a single dataset in a limited time. Thus, it can be argued that the findings of this study could not be generalized to other deep learning settings. This work is the first study explicitly analyzing human skill in hyperparameter tuning; it is interesting to extend this study further by including multiple tasks, models and datasets.

4.2 Conclusions

Human skill impacts accuracy. Through this user study, we found for people with similar levels of experience tuning the exact same deep learning model, the model performs differently. Every source of variation was eliminated by fixing the task, the dataset, the deep learning model, and the execution environment (random seed, GPUs used for execution) except the choice of hyperparameters. Figure 5 shows the variance in the final performance of the model. This suggests that final performance of the model is dependent on the human tuning it. Even for experts the difference can be an accuracy difference of 5%.

More experience correlates with optimization skill. We show a strong positive correlation between experience and final performance of the model. Moreover, the data suggests that more experienced participants achieve better accuracy more efficiently, while inexperienced participants follow a random search strategy, where they often start by tuning optional hyperparameters which may be best left at their defaults initially.

4.3 Recommendations

Concluding our work, we would like to take the liberty to propose some recommendations regarding experiments and their outcome. We base these recommendations on our observed results that even expert accuracy can differ as much as 5% due to hyperparameter tuning. Thus, hyperparameters are essential for reproducing the accuracy of existing work, for making comparisons to baselines, and for making claims based on such comparisons.

- Reproducability: Please share the final hyperparameter settings.
- Comparisons to baselines: Please optimize and report the hyperparameter settings for the baseline with equal effort as the proposed model.
- Claims of (the by now proverbial) superior performance: It is difficult to say if the purported superior performance is due to a massive supercomputer trying all settings [23, 25], due to a skilled human as we show here, or due to qualities of the proposed model. Bold numbers correlate with black magic and we recommend to make bold numbers less important for assessing the contribution of a research paper.
- To the deep learning community: Make reviewers pay more attention to reproducability, baseline comparisons, and put less emphasis on superior performance. There is no need to burn wielders of black magic at the stake, but herald the enlightenment by openness and clarity in hyperparameter tuning.

Acknowledgement

This work is part of the research programme C2D–Horizontal Data Science for Evolving Content with project name DACCOMPLI and project number 628.011.002, which is (partly) financed by the Netherlands Organisation for Scientific Research (NWO).

References

[1] Imagenette. https://github.com/fastai/imagenette.

[2] J. Bergstra, D. Yamins, and D. D. Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. 2013.

[3] A. Bordes, S. Chopra, and J. Weston. Question answering with subgraph embeddings. *arXiv e-prints*, 2014.

[4] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[5] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[6] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[7] T. Domhan, J. T. Springenberg, and F. Hutter. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[8] T. Elsken, J. H. Metzen, and F. Hutter. Neural architecture search: A survey. *arXiv preprint arXiv:1808.05377*, 2018.

[9] C. Farabet, C. Couprie, L. Najman, and Y. Lecun. Learning hierarchical features for scene labeling. In *IEEE Trans Pattern Anal Mach Intell*. IEEE, 2013.

[10] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[11] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 2012.

[12] F. Hutter, H. Hoos, and K. Leyton-Brown. An efficient approach for assessing hyperparameter importance. In *Proceedings of the 31st International Conference on Machine Learning*, pages 754–762. PMLR, 2014.

[13] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and less than 0.5MB model size. *arXiv e-prints*, 2016.

[14] S. Jasper, L. Hugo, and P. A. Ryan. Practical bayesian optimization of machine learning algorithms. *Bartlett et al. [8], pp. 2960–2968*, 2012.

[15] P. Kerschke, H. H. Hoos, F. Neumann, and H. Trautmann. Automated algorithm selection: Survey and perspectives. *Evolutionary computation*, 27(1):3–45, 2019.

[16] R. Kohavi and G. H. John. Automatic parameter selection by minimizing estimated error. In *Proceedings of the Twelfth International Conference on International Conference on Machine Learning*, pages 304–312. Morgan Kaufmann Publishers Inc., 1995.

[17] L. Kotthoff, C. Thornton, H. H. Hoos, F. Hutter, and K. Leyton-Brown. Auto-weka 2.0: Automatic model selection and hyperparameter optimization in weka. *The Journal of Machine Learning Research*, 18(1):826–830, 2017.

[18] A. Koutsoukas, K. J. Monaghan, X. Li, and J. Huan. Deep-learning: investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data. *Journal of cheminformatics*, 9(1):42, 2017.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[20] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 2015.

[21] M. K. K. Leung, H. Y. Xiong, L. J. Lee, and B. J. Frey. Deep learning of the tissue-regulated splicing code. *Bioinformatics (Oxford, England)*, 2014.

[22] H. Liu, K. Simonyan, and Y. Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

[23] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet. Are GANs Created Equal? A Large-Scale Study. *arXiv e-prints*, 2017.

[24] J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl, and V. Svetnik. Deep neural nets as a method for quantitative structure-activity relationships. *Journal of Chemical Information and Modeling*, 2015.

[25] G. Melis, C. Dyer, and P. Blunsom. On the State of the Art of Evaluation in Neural Language Models. *arXiv e-prints*, 2017.

[26] I. Olkin. Contributions to probability and statistics; essays in honor of Harold Hotelling. *Stanford, Calif., Stanford University Press, 1960.*, 1960.

[27] H. Pham, M. Y. Guan, B. Zoph, Q. V. Le, and J. Dean. Efficient neural architecture search via parameter sharing. *arXiv preprint arXiv:1802.03268*, 2018.

[28] N. Reimers and I. Gurevych. Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. *arXiv preprint arXiv:1707.06799*, 2017.

[29] I. Rish. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

[30] T. N. Sainath, B. Kingsbury, A. Mohamed, G. E. Dahl, G. Saon, H. Soltau, T. Beran, A. Y. Aravkin, and B. Ramabhadran. Improvements to deep convolutional neural networks for lvcsr. *arXiv e-prints*, 2013.

[31] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, pages 148–175, 2016.

[32] L. N. Smith. A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*, 2018.

[33] C. Spearman. *Spearman Rank Correlation Coefficient*, pages 502–505. Springer New York, 2008.

[34] J. N. van Rijn and F. Hutter. Hyperparameter importance across datasets. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2367–2376, 2018.

[35] Y. Zhou, S. Cahya, S. A. Combs, C. A. Nicolaou, J. Wang, P. V. Desai, and J. Shen. Exploring tunable hyperparameters for deep neural networks with industrial adme data sets. *Journal of chemical information and modeling*, 59(3):1005–1016, 2018.

[36] B. Zoph and Q. V. Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.