Cognitive Horses of Deep Learning: Regularization and Optimization Practices

Dr. Ajeet K. Jain[1], Dr. PVRD Prasad Rao[2], Dr. K. Venkatesh Sharma[3], & Karan Jain[4]

[1] Asst. Prof., CSE, KMIT, Hyderabad, India
jainajeet123@gmail.com
[2] Professor, CSE KLEF, Vaddeswaram, AP India
pvrdprasad@kluniversity.in
[3] Professor, CSE CVR College of Engg., Hyderabad India
venkateshsharmacse@gmail.com
[4] Scholar, VIT, Vellore India
kjain1496@gmail.com

ABSTRACT

Deep learning networks incorporate various regularization and optiization techniques motivated by effective parameters tuning. The arduous task of training a deep network invariably requires lot of parameter setting along with different architectural design in order to deal with problems of overfitting and convergence speed. These two cognitive working horses play the critical role in convergence for better performance! In recent past, new modifications in these two avatar horses being put into practice and their suitability and applicability has been reported on various domains. Though technique like batch-normalization, data augmentation, dropout and skip connections have promising results in abating overfitting, each method reported have peculiarities and sometime not well addressed and have dithering understanding towards their use. Similarly optimizing algorithms use bias correction second order moment estimates for faster convergence (ADAM, NADAM, YOGI and others). Though this fine, but findings have reported that a simple SGD based optimizers sometime outperform than advanced version of the same implantation. This contradictory and agnostic behavior suggests that, one intuitive way to get into their appropriateness is empirically apply them on different data set and measure their relative performances. There is no single trending horse practices which can either perform better or outperform over another variant for a given application domain. Eventually, it is a difficult practitioner's choice to foresee and get a feeling of which regularizer | optimizer is a best choice! This is indeed a challenging task and is an issue of cognitive deep learning research—to choose an intelligent best-fitting regularizer | optimizer performing optimally for a given model. Additionally, increasing the number of deep layers with hyperparameter tuning introduces higher complexity and a tradeoff is one way to find a niche fit.

The paper delves deeper into the best practices offered by various parameters tuning and reports the agnostic behavior on various datasets.

Keywords: Regularization, Optimization, deep learning, tuning parameters

5. Discussion and Conclusion

As can be visually analysed from various graphs from the experimental analysis, some salient points emerge:

—A good start is to have technique that usually works well with ReLU and tuning of a few hyperparameters

—Error function should reflect the learning goals so that it could provide intuitive learning idea and guessing with approximations

—DL is more about straightening-out the factors, and an appropriate data representation can provide meaningful data transformations

—Start with a simple dataset having fewer (easier) samples with a simple network, and after obtaining promising results gradually increase the complexity of both data and network while tuning hyperparameters and trying regularization methods.

—Training with random initialization is first choice and later one can go with pre-trained model to speed up

—Begin with well-known optimizer to get results and trying out with each will provide a reconnaissance and might lead to improvement. The correctly chosen optimizer with learning rate and other parameters make a good deal. However, there no way a single one optimizers will fit for all types of date set (NO FIT ALL RULE!)

5.A Conclusion

The GD model has been implemented in the work on the principle of Occam Razor : “simplest thing which works first, are better “ . The model takes gradient based approach to fit the regressor on Kaggle2 dataset and it can be best suited on any linearly predictable scenario. Further, the model is able to predict the outcome as percentage of the chance of a GRE score. The Pearson correlation coefficient ‘r’ has a strong bearing on the underlying parameters- as it is evident from the program run and various plots. As the linear regressor is one of the most suitable models fit for such type of data, there are improvements also possible-like using Stochastic GD for better performance and efficiency with a niche fit. Moreover, the model can further be extended to undertake features of multiple dependencies as well by way of looking at the extracted features of the data set and fitting it intuitively.

6. References

[1] Ajeet K. Jain, Dr. PVRD Prasad Rao and Dr. K Venkatesh Sharma; “Deep Learning with Recursive Neural Network for Temporal Logic Implementation”, International Journal of Advanced Trends in Computer Science and Engineering, Volume 9, No.4, July – August 2020 https://doi.org/10.30534/ijatcse/2020/383942020

[2] Ajeet K. Jain, PVRD Prasad Rao and K Venkatesh Sharma;"A Perspective Analysis of Regularization and Optimization Techniques in Machine Learning", Chapter 16; Computational Analysis and Understanding of Deep Learning for Medical Care: Principles, Methods, and Applications", CUDLMC May 2020.

[3] Haykin Simon 2016, Neural Network and Machine Learning, 3rd Ed. Pearson Edn. 2010

[4] CholletFrancois 2018, Deep Learning with Python, Manning Pub., 1st Ed. 2018

[5] Frassord Davi, Linear regression with numpy 2016 (GitHub); www.github /Frossard.htm

[6] Lachlan Miller, Machine Learning: Cost Function, Gradient Descent and Univariate Linear Regression, www.github/lachlanmiller.htm

[7] Ravidran B 2019, NPTEL Course, Introduction to Machine Learning

[8] Karl Gustav Jensson 2108, Introduction to Machine Learning.ML, NPTEL

[9] Gupta A and Biswas G P 2020, Python Programming,

[10] Problem solving, packages and libraries, 1st Ed.TMH Potter Merle C, Potter, J.L. Goldberg and Edward F. Aboufadel 2010 Advanced Engg. Mathematics, Oxford University Press, 3rd Ed.

[11] Charu C. Agrawal, Neural Networks and Deep Learning, 1st Ed. Springer 2018

[12] Bishop, C.M., Neural Network for Pattern Recognition, Clarendon Press, 1995

[13] Glorot, X. and Bengio, Y., Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS), pages 249–256. (2010)

[14] Glorot, X., Bordes, A., and Bengio, Y, Deep sparse rectifier neural networks. In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS), pages 315–323. 2011.

[15] Zeiler, M. and Fergus, R. , Stochastic pooling for regularization of deep convolutional neural networks. In Proceedings of the International Conference on Learning Representations ,ICLR, 2013

[16] Prajit Ramachandran, Barret Zoph, Quoc V. Le, SWISH: A Self-Gated Activation Function, arXiv:1710.05941v1 [cs.NE] 16 Oct 2017

[17] Fabian Latorre, Paul Rolland and Volkan Cevher, Lipschitz Constant Estimation Of Neural Networks Via Sparse Polynomial Optimization, ICLR 2020

[18] Kavosh Asadi , Dipendra Misra and Michael L. Littman, Lipschitz Continuity in Model- based Reinforcement Learning, Proceedings of the 35 th International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018

[19] Hinton, G., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R., Improving neural networks by preventing co-adaptation of feature detectors. arXiv:1207.0580. (2012)

[20] J. Duchi, E. Hazan and Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, Journal of Machine Learning Research, pp 2121-2159, 2011

[21] D. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980, 2014.

[22] S.J. Reddi, S. Kale and S. Kumar, On the convergence of Adam and beyond, International Conference on Learning Representations, Vancouver, Canada, 2018.

[23] Zaheer, M., Reddi, S., Sachan, D., Kale, S., & Kumar, S. Adaptive methods for nonconvex optimization. Advances in Neural Information Processing Systems (pp. 9793–9803), 2018

[24] Hiroaki Hayashi, Jayanth Koushik and Graham Neubig ; Eve: A Gradient Based Optimization Method with Locally and Globally Adaptive Learning Rates, arXiv:1611.01505v3 [cs.LG] 11 Jun 2018

[25] Liyuan Liu , et al., On The Variance Of The Adaptive Learning Rate And Beyond, arXiv:1908.03265v3 [cs.LG] 17 Apr 2020

[26] Nicola Landro, Ignazio Gallo, Riccardo La Grassa, Mixing ADAM and SGD:a Combined Optimization Method, arXiv:2011.08042v1 [cs.LG] 16 Nov 2020

[27] Jonathan Frankle and Michael Carbin, The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks, arXiv:1803.03635v5 [cs.LG] 4 Mar 2019