# Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift

Summary written by Nicholas Hinke
March 08, 2022

**Summary:** This paper presents a novel technique to remedy many of the complications that arise when training deeper networks. At the time, there were several other papers that demonstrated the power and flexibility of using deeper networks for classification tasks [4], and they were consequently gaining popularity. Along with that popularity, however, came many challenges including: much longer training times, exacerbated vanishing/exploding gradients, parameter sensitivity to initial values, more limited choice of activation functions, *etc.*. The method presented in this paper sought to combat all of those issues, as well as to even improve model performance. Denoted as *Batch Normalization* (BN), the authors primarily sought to address the problem of what they called "internal covariate shift" through their new technique. Inspired by the then-recent work on domain adaptation [3], standardization layers [11], normalization statistics [6], and activation whitening [5, 8, 9], the authors claim to have implemented their technique within a model that has achieved a higher classification accuracy than human raters on the ImageNet dataset [2, 7].

**Approach:** As noted above, the authors were primarily attempting to solve the problem of "internal covariate shift" using their new technique. The effects of this problem can be essentially thought of as twofold: 1) the nonlinearity inputs can become highly unstable as the network trains (due to a variety of issues, but especially vanishing/exploding gradients), and 2) the parameter updates at a given layer become relatively *un*-meaningful when their effects are highly sensitive to the changes in the all of the layers preceding them. Clearly, both of these issues are paramount as networks grow deeper, and result in much slower training times, the necessity of lower learning rates, and careful parameter initialization [2].

Although previous work on activation whitening [5, 8, 9] had been shown to help with some of these issues, the authors found the algorithm very computationally (and time!) expensive. As such, they developed their own algorithm known as the *Batch Normalization Transform* ($BN_{\gamma,\beta}$). The algorithm is computed over a mini-batch of inputs $\{x_{1...m}\}$ so as to approximate the statistics of the whole population, and is comprised of four steps: 1) computing the mean of the mini-batch ($\mu_B$), 2) computing the variance of the mini-batch ($\sigma_B^2$), 3) normalizing the inputs using the computed mini-batch mean and variance and an additional term $\epsilon$ for computational stability ($\hat{x_i}$), and 4) scaling and shifting the normalized values to get the new layer inputs ($y_i = \gamma\hat{x_i} + \beta \equiv BN_{\gamma,\beta}(x_i)$). It should also be noted that the parameters $\gamma$ and $\beta$ can be efficiently learned along with the rest of the parameters in the network [2].

As can be seen by the structure of the algorithm, the BN transform can essentially be thought of as another layer in the network. In fact, the authors demonstrated that this 'layer' can be inserted between any two fully-connected or convolutional layers in a given network [2].

After developing their new method, the authors implemented it in a variety of different models in order to study the resulting performances. Indeed, the authors were able to show that utilizing batch normalization greatly increases training speed for a variety of networks, and even sometimes improves the model's best performance. Moreover, when applying their technique in ensemble models similar to those in [1, 10], the authors were able to show superior performance to the best known previous results on a specified portion of the ImageNet dataset [2].

**Strengths:** Most importantly, batch normalization solves many of the challenges of training very deep networks, including faster training times, higher learning rates, reduced chance of nonlinearity saturation, lesser need for other regularization techniques, *etc.*. Additionally, perhaps even more notably, the implementation of the batch normalization technique led to the best ever results on the ImageNet dataset, surpassing even some human raters [7]. Finally, it is *very* easy for anyone to utilize this algorithm, as the BN transform can be inserted between any two network layers and adds only two additional network parameters [2].

**Weaknesses:** Although very effective at empirically demonstrating the performance of their method, the authors spend very little time on theoretical analysis of the algorithm. This leaves much to be desired when attempting to understand exactly *why* batch normalization works as well as it does in practice. In fact, the authors admit on multiple occasions that they are not entirely certain regarding many of the properties of this technique, especially pertaining regularization [2]. Consequently, it would be quite helpful to the reader if more discussion or conjecture was offered in an attempt to explain these unknowns.

**Reflections:** As stated above, much work is needed to further analyze and understand the properties of the batch normalization algorithm. Additionally, as briefly mentioned by the authors, it may prove quite valuable to study the performance implications when applying this technique to other types of deep networks, such as Recurrent Neural Networks (RNNs) [2]. Finally, it would be very interesting to study just how fast a network can be trained by implementing this method on today's modern computer hardware and GPUs.

## References

[1] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.

[2] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, volume 37 of *ICML'15*, page 448–456. JMLR.org, 2015.

[3] J. J. Jiang. A literature survey on domain adaptation of statistical classifiers. 2007.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86:2278–2324, 1998.

[6] S. Lyu and E. P. Simoncelli. Nonlinear image representation using divisive normalization. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[7] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2015.

[8] S. Wiesler and H. Ney. A convergence analysis of log-linear training. In *NIPS*, 2011.

[9] S. Wiesler, A. Richard, R. Schlüter, and H. Ney. Mean-normalized stochastic gradient for large-scale deep learning. *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 180–184, 2014.

[10] R. Wu, S. Yan, Y. Shan, Q. Dang, and G. Sun. Deep image: Scaling up image recognition. *ArXiv*, abs/1501.02876, 2015.

[11] Çaglar Gülçehre and Y. Bengio. Knowledge matters: Importance of prior information for optimization. *J. Mach. Learn. Res.*, 17(8):1–32, 2016.