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

Summary written by Sahana Srihari
March 30th , 2020

Summary: Deep learning networks usually require careful initialization of parameters and a slow learning rate in order to train the network. One of the main problems encountered is internal covariance shift(ICS)[6] which is the constant change in distribution of the layer's input parameters. The paper proposes a method to address this problem by normalization of every training mini-batch which results in higher accuracies while allowing for higher learning rates. Through their methods they attained an impressive 4.9% top-5 validation error on the imageNet dataset.

Related work: Stochastic Gradient Descent (SGD) changed the way people perceived neural networks due to its efficiency in training and along with its variants- Momentum [7] and Adagrad [1].With BN, *ICS* which poses to be a very serious problem as proposed by [6] is solved. The authors propose BN bearing similarity to [2] to normalize the input to each layer [3][4] applied on activations- ReLU [5] as a vital part of a neural network which solves the saturation and vanishing gradient problems. It was proven in [4] that whitened inputs help the networks converge faster which was a step in the motivation behind BN and further changes were developed to improve the system.

Approach: Fixing the distribution eliminates the need of the succeeding layers to readjust, thereby removing *ICS* through BN. The method introduces a normalization step which fixes the activations of each layer as mean = 0 and variance = 1.

(i) Normalization via Mini-Batch Statistics: To represent the identity transform and thus to avoid linear regime of the nonlinearity, 2 extra parameters $\gamma^{(k)}$ and $\beta^{(k)}$ such that $y^{(k)} = \gamma^{(k)}\hat{x}^{(k)} + \beta^{(k)}$ for every mini-batch $\mathcal{B}$ =$x_{1..m}$. The $BN_{\gamma,\beta}$ is computed for every normalized $\widehat{x}_i = \frac{x_i - \mu_\mathcal{B}}{\sqrt{\sigma_\mathcal{B}^2 + \epsilon}}$ where μB , $\sigma_\mathcal{B}^2$ mean and variance of mini-batch. During training, the gradient of loss l is back-propagated through this transformation as it is differentiable. Now, each layer receives an input of BN(x) which trains to optimize parameters $\gamma^{(k)}$, $\beta^{(k)}$ for multiple mini-batches $\mathcal{B}$ such that-

$$y = \frac{\gamma}{\sqrt{\mathrm{Var}[x] + \epsilon}} x + \left(\beta - \frac{\gamma \mathrm{E}[x]}{\sqrt{\mathrm{Var}[x] + \epsilon}} \right)$$

(ii) BN CNNs: BN is applied before the non-linearity on the normalized x = Wu + b such that the new activation z = *g*(BN(Wu)). In the case of CNNs $\mathcal{B}(m' = m.p.q)$ is the set of all values in the feature map(*pxq*) across the mini-batch & spatial locations. The parameters $\gamma^{(k)}$, $\beta^{(k)}$ are learnt on the feature map and the remaining steps follows (i).

Datasets, Experiments and Results: (i)MNIST dataset with a network of 3 FC layers, 100 activations and signmoid activation utilized to check the performance of test accuracies in accordance to training steps & the change in distribution. BN increases stability and accuracy with reduction in internal covariate shifts.

(ii) ImageNet Classification: Network consisted of 2 consecutive 3x3 layers(128 filters) trained on the LSVRC2012 training data. *Results:* BN-baseline achieves contending accuracies with lesser training steps. Significant increase in the training speed using BN-x5 has learning rated increased by a factor of 5.BN-x30 achieves the highest accuracy of 74.8%. BN also produces excellent results(69.8%) when using sigmoid as the activation function. Their method of ensemble classification garnered top-5 error rate of 4.9% compared the previous state of the art.

Strengths: The paper addresses and provides insight on one of the major problems with training deep nets which is internal covariate shift. The proposed method accelerates the training and also has a beneficial effect on the gradient flow thereby allowing higher learning rate and less stringency on the initialized parameters(resilience to parameter scale). It also offers a method regularization thereby reducing the dependency on dropout. Using BN also gives the benefit of using saturating non-linearities with acceptable performance. It has shown high performance against the bench-marked methods utilizing just 7% of the training steps with increased accuracy.

Weaknesses: This method is not optimal in the case of online learning. They paper has shown the effect of BN for FC and Convolution layers but not on the performance on recurrent neural networks and LSTM. The output depends on all the other mini-batches during training but during testing it follows a starkly different computational path due to the normalization of the moving average in contrast to mini-batch average as observed during training. The batch size **has** to be 1.It also has a large computational overhead during training.

Reflections: The solution to internal covariate shift was provided in a simple and elegant manner. The proposed method was easy to follow and is backed with experimental proof. Since the precise effect on gradient propogation is still a mystery, this provides grounds for further study. Further work can be seen in applications of BN to RNNs

& domain adaptions for addressing severe problems of vanishing gradients & generalizing new data distributions.

References

[1] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(Jul):2121–2159, 2011.

[2] Ç. Gülçehre and Y. Bengio. Knowledge matters: Importance of prior information for optimization. *The Journal of Machine Learning Research*, 17(1):226–257, 2016.

[3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[4] Y. LeCun, L. Bottou, G. Orr, and K.-R. Muller. Efficient backprop. *Neural Networks: Tricks of the Trade. New York: Springer*, 1998.

[5] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[6] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

[7] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.