EECS 6322 Week 3, Paper 2
Group Normalization

Shivani Sheth (Student #: 218011783)
shivs29@yorku.ca

I. Summary and contributions

This paper proposes a novel normalization technique known as 'Group Normalization'. It is proposed as an alternative to Batch Normalization in cases where smaller batch sizes are taken due to memory constraints. The authors successfully prove that Group Normalization performs almost at par with Batch Normalization for higher batch sizes (Batch Normalization performs a little better) whereas it significantly outperforms its counterpart for smaller batch sizes.

Drawbacks of Batch Normalization (BN): The authors argue that despite Batch Normalization's great success, it does not perform very well on smaller batch sizes. This is because the BN estimates the statistics such as mean and variance of the training data based on its 'batches' assuming that the statistics for these batches would closely represent the overall statistics of the training data. This assumption is true in cases where large batch sizes are used, but fails in cases where the batch sizes are reduced. The authors show that as the batch size decreases for BN, the error rate increases significantly.

Furthermore, some computer vision methods such as segmentation, video recognition, and detection require the computation of high resolution inputs, which generally limits the batch sizes to very small batches.This in turn limits the performance of those models when used with BN and compromises the model design for better efficiency. Hence, in many cases the memory limitations prohibit people from exploring high-capacity models. Other drawbacks of BN include varying batch sizes for different implementations such as training vs. testing, pre-training vs. fine-tuning, and backbone vs. head in object detection and segmentation.

Advantages of Group Normalization (GN): As the batch size decreases, the error rate of the training data is almost similar to the error rate for large batch sizes, that is, the training data error is batch independent. This is because unlike BN, GN divides the channel dimension into groups which are assumed to have similar features and normalizes the features within each group. Based on these groups of channels, statistics such as mean and variance are calculated which present more accurate results.

Group Normalization is motivated by the fact that many features across different channels such as frequency, shapes, and textures across different channels could have interdependent coefficients and hence could lead to grouping. It calculates the mean and variance based on the formulas in Eqn. (1). where Si is the set of pixels given by Eqn.(2). In the equation,

$$\mu_i = \frac{1}{m}\sum_{k\in\mathcal{S}_i} x_k, \quad \sigma_i = \sqrt{\frac{1}{m}\sum_{k\in\mathcal{S}_i}(x_k-\mu_i)^2+\epsilon},$$

Eqn. (1)

G is the number of groups that the channel dimension is divided into and is set to '32' as default, as a predefined hyper-parameter. C/G represents the number of channels per group, and the equation in the floor operation means that the indexes i and k are in the same group of channels.

$$\mathcal{S}_i = \{k \mid k_N = i_N, \lfloor\frac{k_C}{C/G}\rfloor = \lfloor\frac{i_C}{C/G}\rfloor\}.$$

Eqn. (2)

The Group Normalization technique was implemented on ResNet-50 with G=32 (by default) for comparison with Batch Normalization (BN), Layer Normalization (LN), and Instance Normalization (IN). A total of 8 GPUs were used to train the model and the statistics for Batch Normalization were computed within each GPU. The data was augmented for training and the ImageNet training dataset was trained on 100 epochs. The weight decay used was 0.0001 for all weight layers, and the learning rate was 0.1 which was dropped by 10x at 30, 60, and 90 epochs.

On ResNet-50, with the above implementation details and a batch size of 32 images per GPU, Batch Norm performed slightly better than Group Norm on the ImageNet validation set by a margin of 0.5%. Linear Norm performed worse than Batch Norm by a degradation of 1.7%, while Instance Norm had the largest error rate, being 4.8% more than that of Batch Norm. However, when the batch size was decreased from 32 to 2, GN outperformed BN by having a 10.6% lower error rate. This was a significant decrease in the error rate which showed that GN was more optimal for smaller batch sizes. For Group Norm it was found that 32 number of groups (G) with 16 channels per group (C/G) performed the best and hence LN and IN which follow conditions $G = 1$ and $C/G = 1$ respectively had a higher error rate. When compared with Batch Renormalization (BR) with the hyperparameters $r_{max} = 1.5$, and $d_{max} = 0.5$, GN still outperformed BR by having an error rate 2.1% lesser than BR.

The Group Norm was also implemented on Mask R-CNN baselines replacing the earlier implemented Batch Norm from the pre-trained model. The model was pre-trained on ImageNet and used a weight decay of 0 for γ and β parameters. 8 GPUs were used with the batch size of 1 image per GPU. The

model was trained on the COCO train2017 set and evaluated in the COCO val2017. BN when replaced with GN on the Mask R-CNN using a C4 backbone performed better by 1.1 box AP and 0.8 mask AP. Similarly when GN was applied on the convolutional layers of the box head and the backbone of the FPN backbone, the efficiency of GN increased over its counterpart by a good margin of 1.4 box AP and 0.6 mask AP. Thus, GN performed significantly better in Object Detection and Segmentation as compared to BN. Additionally, GN can also facilitate training Mask R-CNNs from scratch.

Lastly, Group Norm was tested on the Kinetics dataset for video classification on the ResNet-50 I3D baseline. The models were pre-trained from ImageNet and both 32-frame and 64-frame input clips temporal lengths were studied. 8 images per GPU and 4 images per GPU batch sizes were used for the 32-frame input clips and 4 images per GPU batch size was used for the 64-frame input clips. When tested across the validation set, BN performed better by a margin of 0.3% accuracy for the 8 images per GPU 32-frame input clips. On the other hand, GN performed better by 0.7% accuracy and 0.2% accuracy in the 4 images per GPU in the 32-frame and 64-frame input clips respectively. These results confirm that in Video Classification, GN still outperforms BN on smaller batch sizes.

II. Strengths

The most significant strength that Group Normalization has over its counterparts is its constant error rate across different batch sizes (GN is batch independant). It outperforms Batch Normalization for smaller batch sizes thus enabling more efficient training for larger systems which might have faced memory constraints with batch normalization. In cases where models performed poorly due to change in batch sizes such as pre-training and fine tuning in Object Detection & Segmentation, GN is able to enhance the performance of the model significantly. Thus, it overcomes all limitations faced by batch normalization. Finally, in cases where the input size is large such as Video Classification, the batches that are limited to small sizes were less efficient earlier. Their efficiency has now increased significantly with GN.

III. Weaknesses

Batch Normalization's uncertainty caused by the stochastic batch sampling while computing the mean and variance values contributes towards regularization which helps Batch Normalization perform better for larger samples such as 32 images per batch. Another drawback for GN which is not directly a weakness but prevents GN from reaching its optimal efficiency is that since Batch Normalization is widely used, the existing hyperparameters are optimal for BN. Hence, they might not work best with GN but more research on hyperparameters for GN can improve its efficiency in future.

IV. Correctness

The claims and empirical methodology proposed in the paper are correct and are backed up by results in favour of Group Normalization for small or varying batch sizes in batch norm. More specific versions of the approach have been implemented earlier, which further strengthens the claims of the authors. This normalization technique aims to generalize the previous techniques and is successful in making the models more efficient thus having an overall lower error rate.

V. Clarity

The paper is well written and formatted in general with a few minor errors. Figures, graphs, and statistics have also been provided adequately to represent the facts and results wherever required. It also provides clarity in terms of its previous work and elaborates upon the differences between the related normalization methods. It aims to replace Batch Normalization and presents a fair comparison with respect to its counterpart in different applications and datasets.

VI. Relation to prior work

Related work includes previous normalization techniques such as Layer Normalization (LN) that operated along the channel dimensions and Instance Normalization (IN) that operates for every individual sample. Both normalization techniques do not have the drawbacks of Batch Normalization (BN) but they also fail to reach the accuracy level of batch normalization. LN and IN are in fact, specific versions of Group Normalization (GN). LN equals to GN when G =1 assuming all channels to be grouped in a single layer, whereas IN equals to GN when C/G = 1, and hence G=C which assumes a single channel per group. GN on the other hand, is a more general version of LN and IN and does not have the drawbacks of BN. It either performs similar to BN for higher batch sizes or performs better for lower batch sizes.

Another modification of BN known as Batch Renormalization was also introduced to reduce the training error for smaller batch sizes. It prevented the mean and variance of smaller batches from drifting away from the actual values by keeping an estimated range for those values. Although this decreased the error for smaller batch sizes as compared to BN, it was still batch dependent and suffered from an increase in error when the batch sizes were reduced. On the contrary, Group Normalization is batch independent and hence the training error is not affected by the size of the batch chosen (it almost remains constant across higher and lower batch sizes).

Previous work on group wise convolutions were explored in architectures such as AlexNet by distributing the model into two GPUs. MobileNet, Xception, and ShuffleNet also performed group convolutions by dividing the channel dimensions into groups. Although Group Normalization resembles the operations performed in the channel division, it does not require 'group convolutions' and instead acts as a generic layer.

VII. Reproducibility

The work proposed in the paper can be easily reproduced using common Machine Learning frameworks such as TensorFlow and PyTorch where automatic differentiation is supported. A code snippet for the same in Figure 3 of the

paper has been provided as an example. As per the authors, Group Normalization can be directly applied by changing the dimensions of the tensor by computing the mean and variance in the respective framework and hence can be easily reproduced.

VIII. Additional comments

The paper was well written and provided a thorough analysis overall. A few suggestions and feedback for the paper would include the following. In Figure 5, the description refers to the right graph as LN (Layer Normalization), but the figure is labelled as GN (Group Normalization). Table 7 could include synchronous BN results for clear comparison. Graphs and figures could have been corresponding to their explanations. This is not an error, but it would make it easier for the reader to quickly refer alongside the explanation.