Grad Queue : A probabilistic framework to reinforce sparse gradients

Irfan Mohammad Al Hasib
irfanhasib.me@gmail.com

Abstract

Informative gradients are often lost in large batch updates. We propose a robust mechanism to reinforce the sparse components within a random batch of data points. A finite queue of online gradients is used to determine their expected instantaneous statistics. We propose a function to measure the scarcity of incoming gradients using these statistics and establish the theoretical ground of this mechanism. To minimize conflicting components within large mini-batches, samples are grouped with aligned objectives by clustering based on inherent feature space. Sparsity is measured for each centroid and weighted accordingly. A strong intuitive criterion to squeeze out redundant information from each cluster is the backbone of the system. It makes rare information indifferent to aggressive momentum also exhibits superior performance with larger mini-batch horizon. The effective length of the queue kept variable to follow the local loss pattern. The contribution of our method is to restore intra-mini-batch diversity at the same time widening the optimal batch boundary. Both of these collectively drive it deeper towards the minima. Our method has shown superior performance for CIFAR10, MNIST, and Reuters News category dataset compared to mini-batch gradient descent.

1. Introduction

Pursuing the global minima is a fundamental problem in gradient based learning. Stochastic and batch gradient descent are the two opposite extremes in this regime. Stochastic gradient descent suffers from the inherent sample variance (Johnson & Zhang, 2013; Fang, Li, Lin, & Zhang, 2018; Wang, Ji, Zhou, Liang, & Tarokh, 2018, 2019). In contrast full batch gradient descent has different problems i.e - (i) computation cost per update for large datasets (ii) poor performance due to generalization gap. Computational cost can be counteracted by distributed learning (Goyal, Dollar, Girshick, Noordhuis, Wesolowski, Kyrola, Tulloch, Jia, & He, 2017). The reason of generalization gap is not well defined yet. Apart from over-fitting, (Keskar, Mudigere, Nocedal, Smelyanskiy, & Tang, 2016) addresses lack of exploratory property and attraction to saddle points as reasons. (Yin, Pananjady, Lam, Papailiopoulos, Ramchandran, & Bartlett, 2018) addresses lack of diversity in consecutive updates as a potential reason. Informative gradients can disappear in large mini-batch causing reduced diversity. In this work we aim to resolve this by boosting the rare updates. Availability of large GPU (graphical processing unit) memory and distributed learning leads to unprecedented growth in batch size with reasonable time constraint. Therefore, pushing the upper bound of mini-batch is one of the prime focus in deep learning research (Goyal et al., 2017; Yin et al., 2018; De, Yadav, Jacobs, & Goldstein, 2016).

Monotonous information keeps repeating thus often learnt quickly, leaving rare information the prime key to dive lower in the loss curve. The success of the adaptive optimizers relies on this fact (Duchi, Hazan, & Singer, 2011) (Hinton, Srivastava, & Swersky, 2012) (Kingma & Ba, 2014). The common goal of all these algorithms is to extract the most informative updates from a set of gradients. To aid this objective, we detect the sparse updates by quantifying its distance from magnitude spectrum in the recent past. We append an amplification plugin on top of any arbitrary gradient acquisition pipeline. It amplify the sparse and minimize the monotonous values, resulting a good contrast. The margin of this reinforced diversity is controlled inherently by their variance. The larger the batch size gets the more invisible the sparse signal becomes, our method come into rescue. It demonstrates the potential to increase accuracy at optimal batch size and specially beyond it. We weight the new updates by its scarcity based on past trend from the queue. For beyond optimal batch size, we group them using k-means clustering (based on a latent feature space). Then extract and emphasize the cluster centers with highest informativeness before totaling them at the end. Informativeness is determined by the scarcity of the occurrence. This intra-mini-batch grouping and enforcing the sparse components per group, ensures minimal loss of informative gradients. Lower risk of loosing diversity allows larger batch size. In case of a single cluster this method will still boost the rarely occurring signals to aid better utilization of the unique updates. The length of the queue is updated based on the trend in change of loss value for the past updates. By making the queue length short and flexible we focus on the subset of the past gradients which can help the current update most. Incorporating a probabilistic approach makes it more suitable for the stochastic nature of the process. We also classify monotonous, sparse and noisy updates and their effect on mini-batch update for better understanding our approach.

2. Literature review

Extracting the most out of a particular set of data points is a fundamental desideratum in machine learning. For batch update, works include optimizing the size of a mini-batch, training with larger batch size. In contrast, for online gradient descent variance reduction techniques are applied for fast convergence. Some works aim to design efficient data selection strategy for each update for getting more useful gradients. While others try to manipulate the gradient itself by comparing its distribution from a auxiliary sources. In this work we are comparing the gradient with its own past distribution.
Size of Mini-batch : Optimization difficulty caused by large mini-batch is the prime obstacle in speeding up training with large datasets. The speed up saturation with distributed learning drawing more attention to this phenomenon (Keskar et al., 2016)(Goyal et al., 2017) proposes several strategies to overcome it. (Yin et al., 2018) proposed a lower optimal bound for batch size to maintain the diversity of updates . (Friedlander & Schmidt, 2012), (De et al., 2016) attempts to resolve it by adaptively growing batch size with time. (Goyal et al., 2017) proposes several workaround to push the upper limit of batch size. Yet today it is left to be tuned real time by the practitioners by and large.
Variance Reduction: Instead of going for larger batch size, many works propose to reduce the variance of stochistic update. (Johnson & Zhang, 2013)(Fang et al., 2018)(Wang et al., 2018),(Wang et al., 2019) proposed to reduce variance by keeping a snapshot of older weights in the expense of a complete pass through the entire dataset. (Elibol, Lei, & Jordan, 2020) proposes special operator to minimize this computational cost.
Sample selection strategy: These approaches attempt to improve the quality of the mini batch by creating a dataset with informative samples regardless of the size of mini-batch.(Agarwal, D'souza, & Hooker, 2022),(Zhdanov, 2019),(Csiba & Richtárik, 2018). Some works focus on serving the samples in phases based on their inherent complexity (Bengio, Louradour, Collobert, & Weston, 2009) (Alain, Lamb, Sankar, Courville, & Bengio, 2015) (Jiang, Zhou, Leung, Li, & Fei-Fei, 2018)

(Fan, Tian, Qin, Bian, & Liu, 2017) (Kumar, Packer, & Koller, 2010) (Loshchilov & Hutter, 2015). **Gradient manipulation :** Auxiliary task learning aims to preserve the important components of the gradients by comparing their distribution with a separate source of gradients from an auxiliary task (Al Hasib, Sultana, Nyeen, & Sabur, 2023) (Du, Czarnecki, Jayakumar, Farajtabar, Pascanu, & Lakshminarayanan, 2018). In that sense our work compares the gradients with its own past distribution within a close horizon. Clustering the gradients can minimize the variance within the clusters thus reduces destructive interference while averaging. (Faghri, Duvenaud, Fleet, & Ba, 2020) proposed a learnable framework for the cluster centers in expense of additional computation. In this work we cluster the samples thus the gradients, based on the activations from a dense layer. The use of dense outputs from pre-trained model for image matching or retrieval (Efe, Ince, & Alatan, 2021) (Arandjelovic, Gronat, Torii, Pajdla, & Sivic, 2016) inspire us to use dense features for grouping the gradients.

3. Methodology

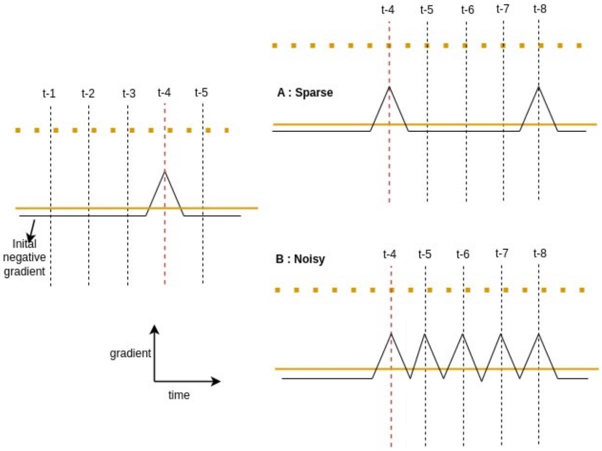

Figure 1: Figure 2:

3.1 Attention to rare signal

From a information science perspective, a useful signal is something that adds some unique value to the process. In that sense signal to noise ratio can be controlled little counter-intuitively by paying attention to the rare spikes, similar to a low pass filter. This hypothesis holds for online gradient acquisition in stochastic method. There is no guaranteed way to distinguish between a useful sparse signal and series of noisy spikes from the first appearance. Its nature can be established based on later occurrences. The more we observe the future values the more it will be clearer. fig-[6] One unavoidable issue in stochastic optimization is that it cannot know the upcoming gradients in advance. At best, the past trend can be stored and analyzed to determine the consistency of the current signal. We acknowledge the exponential averaging methods (Kingma & Ba, 2014; ?)] can preserve the sparsity among gradients to some extant by making a balance between first and second moment. Here we rather aim to design a standalone distance function to compute a numeric measure.

We measure the instantaneous expected value and standard deviation for each gradient from the queue. The distance of the gradients from its expected value per unit standard deviation measures how rare the current occurrence is [eq-1]. We multiply this distance as a scalar weight with each of the gradients [eq-2]. For sparse values this distance will be large, the opposite is also true. It practically acts like a low pass filter thus passes the desired signals with high amplitude and dampens the rest. For maintaining a stable bound to this amplification factor, we cut it off from $1/\rho$ to ρ range. Empirically $\rho = 3$ found to be effective in our experiments. The method can be used with or without momentum. SGDM (SGD with momentum) always demonstrates superior performance than vanilla SGD (Liu, Gao, & Yin, 2020). SGDM comes with high resistance to sudden change. It makes the process robust to stochastic noise but diminishes sparse change as well. This phenomenon is a widely researched issue(Kingma & Ba, 2014; ?). Since our method is specially designed to enforce rare signal, it makes a effective combination with momentum. The potential of our method to rescue rare signal with momentum proves the effectiveness of the claim in later section. For gradients - $g_1, g_2, g_3 ..., g_t$ with mean μ_t, standard deviation σ_t and the constant $\rho > 1$ we can define the distance operator Δ_ρ in eq-[1].

$$\Delta_\rho(g_t, \mu_t, \sigma_t\ ) \ = \begin{cases} min(abs(g_t - \mu_t)/\sigma_t, \rho) g_t & if,\ abs(g_t - \mu_t)/\sigma_t \ > \ 1 \\ max(abs(g_t - \mu_t)/\sigma_t, 1/\rho) g_t & otherwise \end{cases} \tag{1}$$

$$\begin{aligned} &\mu_t = \mathbb{E}(g_i); \sigma_t = \sqrt{\mathbb{E}(g_i^2) - \mathbb{E}(g_i)^2}; i = t-1, t-2, ...t-n \\ &m_t = beta * m_{t-1} + \Delta_\rho(g_t, \mu_t, \sigma_t) \\ &\theta_t = \theta_{t-1} - \alpha * m_t \end{aligned} \tag{2}$$

We define the following function as a standard template of a sparse signal generator for our analysis. Based on various values of C , u and N we will examine the behaviour of different optimization techniques where $N >= 3$.

$$\text{f}(t) = \begin{cases} C, & \text{if } t \,\%\, N = 0 \\ u, & \text{otherwise} \end{cases} \tag{3}$$

Lemma 3.1. *Using gradients generated from eq. [3] the consecutive values for t=1 to N would be -* $g_1 = g_2 ... = g_{N-1} = u$; $g_N = C$. *Initializing momentum update equation* $m_{t+1} = \beta m_t + m_{t+1}$, *with* $m_0 = 0$, kN^{th} *momentum will be -*

$$m_{kN} \ = \ \beta_k^N (u \beta \beta_{N-1} \ + C) \qquad where,\ \beta_x = \frac{\beta^x - 1}{\beta - 1} \quad ; \ \ k \ > \ 0\, and\, k\, \varepsilon\, \mathbb{Z}$$

If $uC < 0$ (u and C with opposing sign) the effect of momentum will be most destructive. In subsequent sections we will consider this worst case scenerio for our analysis. From [3.1] m_N to follow the direction of C the criterion is as follows -

$$|u\beta\beta_{N-1}| < |C| \ \ => \ |\frac{C}{u}| > \beta\beta_{N-1} \tag{4}$$

Fo example, a typical value of $\beta = 0.9$ and N=3 the term $|\frac{C}{u}| > |\beta * \beta_{N-1}| = 2.44$, for N=9 it grows up to 5.51 Figure [2b]. For N = 9 , $\beta * \beta_{N-1} > 5.5$. Sparse signals below this threshold follows the direction of u adversely, shown in Figure [2a]. In contrast, here GQ will boost the sparse gradient, pushing this limit at least ρ times lower, up to $\approx \rho^2$ depending on how small *qlen* is compared to sparse frequency N.

Lemma 3.2. *For a queue of length L composed of two unique elements u and C where number of u is L-1. Then* $\Delta_\rho(u) = \phi u$ *; where* $\phi \ = max\left(\frac{1}{\sqrt{(L-1)}}, \frac{1}{\rho}\right)$.

Lemma 3.3. *Using the same condition of [3.1] momentum at* kN^{th} *step boosted with [1],*

$$\Delta_\rho(m_{kN}) = \ \beta^{N(k-1)}\left(u\beta\gamma^0_{N-1} \ + \ \rho C\right) \ + \ \beta^N_{k-1}\left(u\beta\gamma_{N-1} \ + \ \rho C\right)$$

$$where, \ \gamma^0_{N-1} \ = \beta^{N-1-L}\beta_L + \ \beta_{(N-1-L)}\frac{1}{\rho} \ \ ; \ \ \gamma_{N-1} \ = \phi\beta^{N-1-L}\beta_L + \ \beta_{(N-1-L)}\frac{1}{\rho}$$

$\gamma^0_x > \gamma_x$ so if the first term follows direction of C the 2nd term will do as well. So, for gq boosted momentum the $|C/u|$ bound for $\Delta_\rho(m_{kN})$ to follow the direction of "C" will be -

$$|u\beta\gamma^0_{N-1}| < |\rho C| \quad => \ |\frac{C}{u}| > \frac{1}{\rho}(\beta\gamma^0_{N-1}) \tag{5}$$

Since $\rho > 1$, $\gamma^0_x = (\beta^{x-L}\beta_L + \beta_{x-L}\frac{1}{\rho}) < (\beta^{x-L}\beta_L + \beta_{x-L}) = \beta_x$. if $\frac{L}{N} << 1$ $\gamma^0_x \approx \frac{\beta_x}{\rho}$
Comparing the $|C/u|$ lower bounds of eq[4] and eq[5] -

$$\begin{aligned} \frac{1}{\rho}\beta\gamma^0_{N-1} < \beta\beta_{N-1} \\ if \ \frac{L}{N} << 1 \ ; \ \gamma^0_{N-1} \rightarrow \frac{\beta_{N-1}}{\rho} \ ; \ \frac{1}{\rho^2}\beta\gamma^0_{N-1} << \beta\beta_{N-1} \end{aligned} \tag{6}$$

So, gq boosting lowers the $|C/u|$ limit for m_N to follow the direction of 'C' at least ρ times to ρ^2 times based on the L/N ratio.

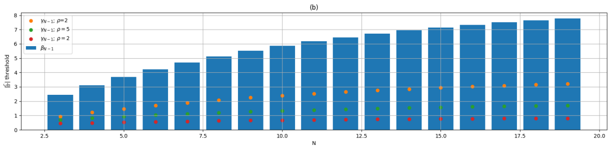

Figure 2: β_N ,γ_N for $\rho = 2,3,5$ is shown for N steps.

We have found too large queue length to be less usefull in our experiments. After several updates the weight will be very different. So gradients wrt older weights will not be a good indicator of the current the trend. We found 3 to 5 queue length good enough for our purpose. We also introduced a variable queue length scheme by monitoring the loss convergence pattern bounded by a upper bound. The more steps back loss pattern shows continuous convergence the higher the queue length will be. A small window is slided over the queue of loss from current step t up to the t-qlen step. It continues to move back as far as the sum within the window increases. It is a measure of how long the loss has been decreasing. The window gives it robustness to noise.

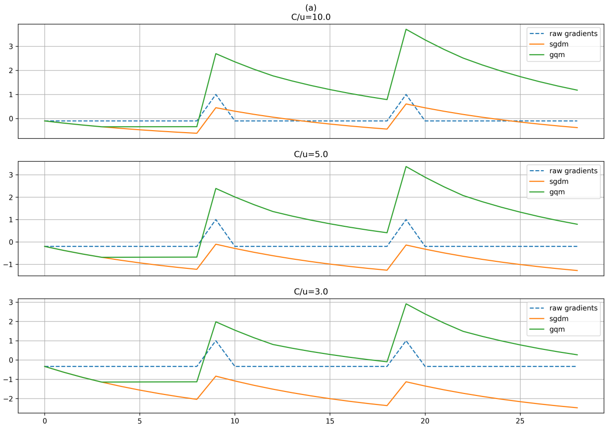

Figure 3: Momentum values for gradient generated from [3].

3.2 Monotonous and Noisy Gradients

We illustrate a problem of line detection for our subsequent analysis. Assume a random uniform pull of B samples resulting in “p” horizontal and “q” with vertical line samples where, $p >> q$ and $p+q=B$. The detector model has 2 parallel $3x3$ CNN (convolutional Neural Network) filters at its input layer followed by respective global max pooling layers and a common dense layer with 2-input and 1-output node at the end for class prediction. Based on their initialization one $3x3$ will start learning the horizontal lines detection and another will learn vertical ones. We name them filter-1 and filter-2 respectively. Each filter will have the largest gradients for samples containing the feature it is learning. So horizontal lines will result in the largest gradients for filter-1 while filter-2 will also have some non-zero gradients driving it far from its optimal parameters. This is the component that can act as a destructive monotonous counterpart for these filter-2 parameters. Below the optimal values for filter-1 and 2.

$$filter-1 = \begin{matrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{matrix} \quad filter-2 = \begin{matrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{matrix}$$

Due to the abundance of samples from ”p”, it will frequently cause beneficial but redundant updates for filter-1 while corrupting filter-2 weights. Rarely occurring “q” samples will ignite the filter-2 kernel which are the sparse updates of interest. Evidently, suppressing the p updates for making the q (sparse) gradients effective is the solution for filter-2 to learn properly, which is what we are going to do later in this work. Apart from above, there can be generic noise in gradients during stochastic updates. Presence of any irrelevant feature in the sample (e.g noisy background) will add some noisy component in respective gradients. Choosing a reasonably large mini-batch can be an easy escape from this situation. Noises come uniformly with every possible pattern thus canceling each other’s effect when averaged. Many of the existing works tried to deal with the stochastic coming from per sample noise [][][] others focused on enforcing the sparse gradients by updating the optimization method [][][] or by choosing diverse training samples [][][]. We implemented this problem with a synthetic dataset of lines. We simulated with 95 horizontal line and 5 vertical line and the same model described above. Figure-[4] shows the gradients and propagation of weights for 4 consecutive time steps. Note here filter-1 expects negative gradient at 2nd row while filter-2 expects the same for 2nd column. For other row or columns the opposite is expected. Only the mini-batch of step˙3 contained a sparse update (vertical line). It is clear from the figure that filter-1 is learning the horizontal line detection quite fast. Filter-2 useful gradients at step˙3. At other steps filter-2 is having non opposing gradients which is acting as destructive noise from the monotonous updates. For the last two values of the 3rd row of the of filter-2 gradients the destructive interference is quite high.

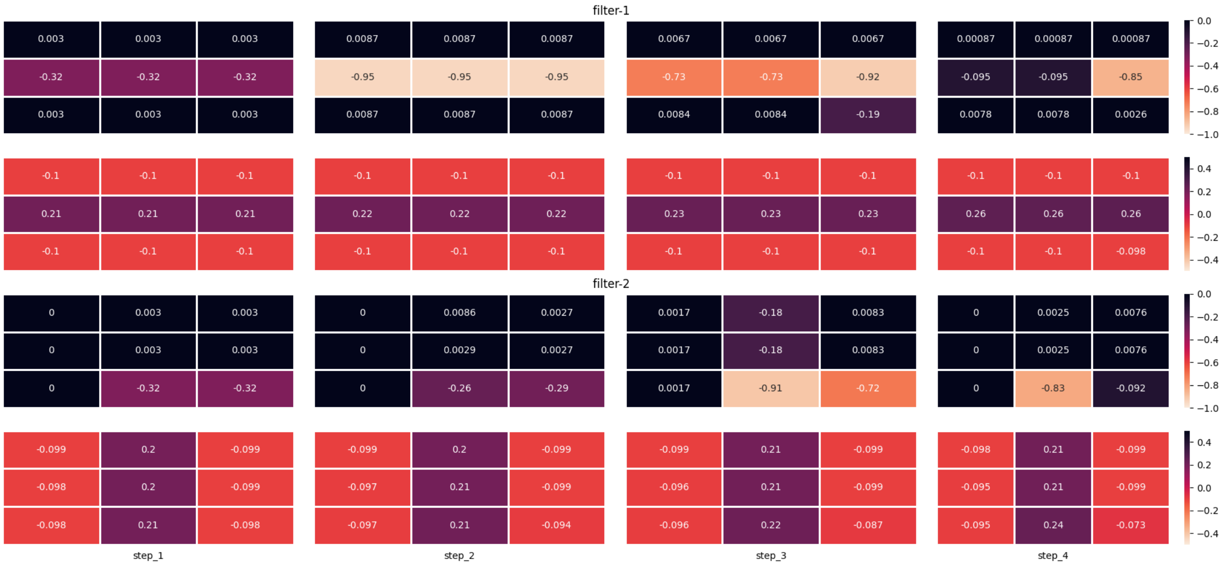

Figure 4: Gradients and propagation of weights for 5 consequitive time steps for filter-1 and filter-2 trained on synthetic line dataset

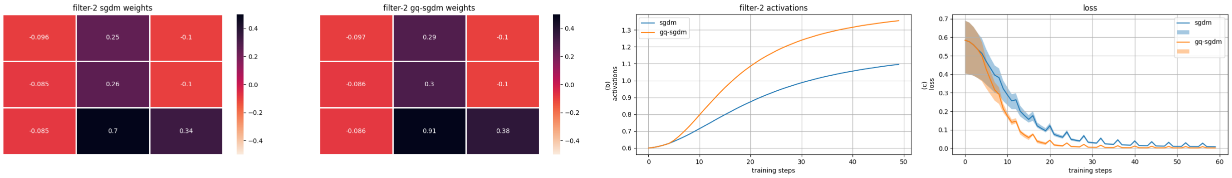

Figure 5: For sgdm and gq-sgdm method (a) weights of filter-2 (b) activations of filter-2 with training steps (c) loss

Online SGD or small mini-batch suffers from the noises from individual samples. If we keep increasing the batch size the monotonous gradients will become dominant enough to suppress the sparse ones.

Assumption 3.1. *From the above hypothesis we can conclude, optimal batch size "B" stays within the following bounds -*
i. Lower bound [ω] : It should be large enough to overcome the stochastic noises.
ii. Upper bound [ψ] : It should be small enough for preserving the sparse updates.

For gradients pulled from "p" and "q" samples - $\mathbb{E}\left(g^q\right) = \frac{1}{q}\sum_{i\ \varepsilon\ q} g_i^q$ *and* $\mathbb{E}\left(g^p\right) = \frac{1}{p}\sum_{i\ \varepsilon\ p} g_i^p$. In destructive cases where $\mathbb{E}(g^q)$, $\mathbb{E}(g^p)$ have opposing signs and $||\mathbb{E}(g^q)/\mathbb{E}(g^p)|| <= p/q$, any uniform mini-batch gradient will diminish the sparse update completely hence will cross the upper bound ψ. The solution is to boost $\mathbb{E}(g^q)/\mathbb{E}(g^p)$, so that we can select a mini-batch large enough to cross the lower bound ω.

Assumption 3.2. *eq[1] imposes a minimal distance limit of 1 standard deviation for a signal to be sparse. Ideally, $|\mathbb{E}(g^q)| >> |\mu+\sigma|$ and $\mathbb{E}(g^p) \approx \mu$. So there exists a ρ for which, $|\mathbb{E}(g_q)| > |\mu_g + \rho\sigma|$, $|\mathbb{E}(g_p)| < |\mu_g + \sigma/\rho|$ where $\rho > 1$ and $\rho \varepsilon \mathbb{Z}$*

From [eq-1] ,

$$\Delta_\rho\left(\mathbb{E}(g^q)\right) = \rho\mathbb{E}(g^q)\ ;\ \ \Delta_\rho\left(\mathbb{E}(g^p)\right) = \mathbb{E}(g^p)/\rho$$

[(Friedlander & Schmidt, 2012)] derived the following convergence equation for gradient descent with error in gradient calculation -

$$f(x_k+1) - f(x^*) \leq (1-\mu/L)\left[f(x_k) - f(x^*)\right] + \frac{1}{2L}\|e_k\|^2 \tag{7}$$

assuming, $f(x)$ is strongly convex with positive μ and L-Lipschitz continuous. Hypothetically, a batch containing "q" type samples would be fully effective if it can preserve the sparse magnitude i.e $\mathbb{E}(g^b) = \mathbb{E}(g^q)$ without any interference. So, the error $||e_k||$ in [7] can be modeled as deviation of $\mathbb{E}(g^b)$ from $\mathbb{E}(g^q)$. Note, there can be another component of this error coming from stochistic noise. We are aiming to improve accuracy on optimal and beyond optimal batch size above the lower bound ω of [3.1]. So this component can be safely overlooked for this study.

$$||e_k|| = ||\mathbb{E}(g^q) - \mathbb{E}\left(g^b\right)|| \ , \mathit{where}\ ||\mathbb{E}(g^q)|| > ||\mathbb{E}\left(g^b\right)|| > ||\mathbb{E}(g^p)||$$

$$\mathbb{E}\left(g^b\right) = 1/B\left(\sum_{i\ \varepsilon\ q} g_i^q + \sum_{i\ \varepsilon\ p} g_i^p\right) = 1/B\ \left(q\mathbb{E}(g^q) + p\mathbb{E}(g^p)\right)$$

$$\begin{aligned}
&\mathit{case}\ 1: && ||\mathbb{E}(g^q)/\mathbb{E}(g^p)|| \gg\ p/q; \mathbb{E}\left(g^b\right) \to \mathbb{E}(g^q)\ ; && ||e_k|| = 0\\
&\mathit{case}\ 2: && ||\mathbb{E}(g^q)/\mathbb{E}(g^p)|| =\ p/q;\ \mathbb{E}\left(g^b\right) = 0 \qquad ; && ||e_k|| = ||\mathbb{E}(g^q)||\\
& && && \mathit{where},\ \mathbb{E}(g^q) * \mathbb{E}(g^p) < 0\\
&\mathit{case}\ 3: && ||\mathbb{E}(g^q)/\mathbb{E}(g^p)|| \ll\ p/q;\ \mathbb{E}\left(g^b\right) \to \mathbb{E}(g^p); && ||e_k|| = ||\mathbb{E}(g^q) - \mathbb{E}(g^p)||
\end{aligned}$$

Note, in case-2 we demonstrated the worst outcome only. Case-2 and 3 exceeds the upper bound ψ of assumption[3.1] If assumption[eq-3.2] holds, applying Δ_ρ to $\mathbb{E}(g^q)/\mathbb{E}(g^p)$ both the cases can be turned into case-1 for a large enough ρ.

$$\Delta_\rho\left(||\mathbb{E}(g^q)/\mathbb{E}(g^p)||\right) = \rho^2\ ||\mathbb{E}(g^q)/\mathbb{E}(g^p)||$$

$\mathit{if}\ \rho\ \mathit{is\ high\ enough}$,

$$\rho^2\ ||\mathbb{E}(g^q)/\mathbb{E}(g^p)|| \gg\ p/q$$

$$\Delta_\rho\left(\mathbb{E}\left(g^b\right)\right) \to \mathbb{E}(g^q);\ ||e_k|| \to 0$$

$\mathit{Here}\ ,\ 1 < \rho \leqslant \zeta$

$$\mathit{For}\ \rho = \zeta\ ,\ \Delta_\zeta \mathbb{E}\left(g^b\right) = \mathbb{E}(g^q)$$

$$\Delta\zeta\left(1/B\ \left(q\mathbb{E}(g^q) + p\mathbb{E}(g^p)\right)\right) = \mathbb{E}(g^q)$$

$$q\zeta\mathbb{E}(g^q) + p\mathbb{E}(g^p)/\zeta = B\mathbb{E}(g^q)$$

$$\zeta^2 q\mathbb{E}(g^q) - \zeta B\mathbb{E}(g^q) + p\mathbb{E}(g^p) = 0$$

$$\zeta = \frac{B\mathbb{E}(g^q) + \sqrt{B^2\left(\mathbb{E}(g^q)\right)^2 - 4q\mathbb{E}(g^q)\,p\mathbb{E}(g^p)}}{2q\mathbb{E}(g^q)}$$

3.3 Enhancing Intra-Batch sparsity

In the above analysis we see that a data sample can be specifically responsible for teaching the model some particular sub-task/tasks (e.g horizontal or vertical line detection) to achieve the collective objective (e.g line detection). Each sub-task can be learnt by a subset of the parameter space (like the two convolutional filters in our case). If the sample contains a rare feature, blending its gradient with regular ones can make the model to not learn the associated sub-goal. The problem here is, as the group gets larger the internal sub-tasks will start to differ and interfere with each other destructively as we have seen in Figure [4]. It can create enough interference with sparse updates to get it nullified by the monotonous counterparts, pursuing different sub-goals. It is the reason for failure of too high mini-batch size to reach good accuracy []. To handle this we further cluster data samples within a large batch. It is done based on the similarity in the inherent feature space of each sample. So the sub objectives within each cluster stays aligned. Consequently, samples with sparse objectives are expected to stay in similar clusters.

For a particular parameter W of a model. The mini-batch size is B and the number of classes is C. At every update we will get a BxC matrices of loss hence a matrix of BxC gradients. Online SGD

will take the overall mean of $m*n$ gradients to update W. We do the following instead -
- Cluster the n samples into k groups, replace each cluster with their center weighted by group population, resulting in KxC gradients.
- Take average across the class dimension, resulting in K gradients for weight W.
- Calculate the K distances Δ element wise for each of the K gradients.
- Take the weighted mean as follows - $G* \;=\; \sum_{i=1}^{k} \Delta_i \;*\; G_i$

Image search or re-identification algorithms[][] uses feature map F from an intermediate dense layer as the matching criteria. Inspiring from that we extract the vector F from an intermediate dense layer of the model. For a batch total B the samples are clustered into k clusters using the corresponding feature vectors. Now within each of these clusters we squeeze out the unnecessary information by applying our "sparsity amplification operator" on the cluster center.

Algorithm 1 Algorithm (GQ)

```
Init $\theta_0, m_0$
$Q = [0,0,...], L = length(Q)$
set $\alpha, \beta, \rho, k$
while $epoch$ do
    $\mu^t \;=\frac{1}{L}\sum_{i=t-1}^{t-L} g_i$ ; $\sigma^t \;=\; \frac{1}{L}\sum_{i=t-1}^{t-L}(g_i \;-\; \mu_t)$
    for $x_b^t, y_b^t$ in Batches do
        $z_b^t \leftarrow \mathbb{F}_{fp}(x_b^t, y_b^t, \theta^t)$, ; $z_b^t$= feature vector $\mathbb{F}_{fp}$=Forward pass.
        $g_b^t \leftarrow \mathbb{F}_{bp}(x_b^t, y_b^t, \theta^t)$, ; $g_b^t$ = batch gradients, $\mathbb{F}_{bp}$ = backward pass.
        $g_1, g_2 ... g_k = \mathbb{C}(z_b^t, g_b^t, k)$ ; $\mathbb{C}$ = Apply KMeans on $z_b^t$ with K=k, split $g_b^t$ based on it.
        $g^* = \frac{1}{B}\sum_{i=1}^{k} len(g_k) * \Delta_\rho(mean(g_k), \mu^t, \sigma^t)$
        $\theta_{t+1} \leftarrow \theta_t + \alpha g^*$,
        $Q \leftarrow g_t$
    end for
end while
```

4. Experimental Results

We have experimented on CIFAR10, MNIST dataset with SGDM(Stochistic Gradient Descent with Momentum) , sgdm boosted with our GQ(grad queue) method namely GQ-SGDM. For GQ-SGDM we have tried with one single cluster and multiple clusters for different batch sizes. For reuters dataset we have used ADAM optimizer and shown comparison with ADAM boosted with GQ namely GQ-ADAM. In every cases grad queue boosted method out performed vanilla optimizers and higher number of cluster out performs single cluster for large batches. For beyound optimal batch sizes we have used number of clusters equals to its ratio with optimal size.

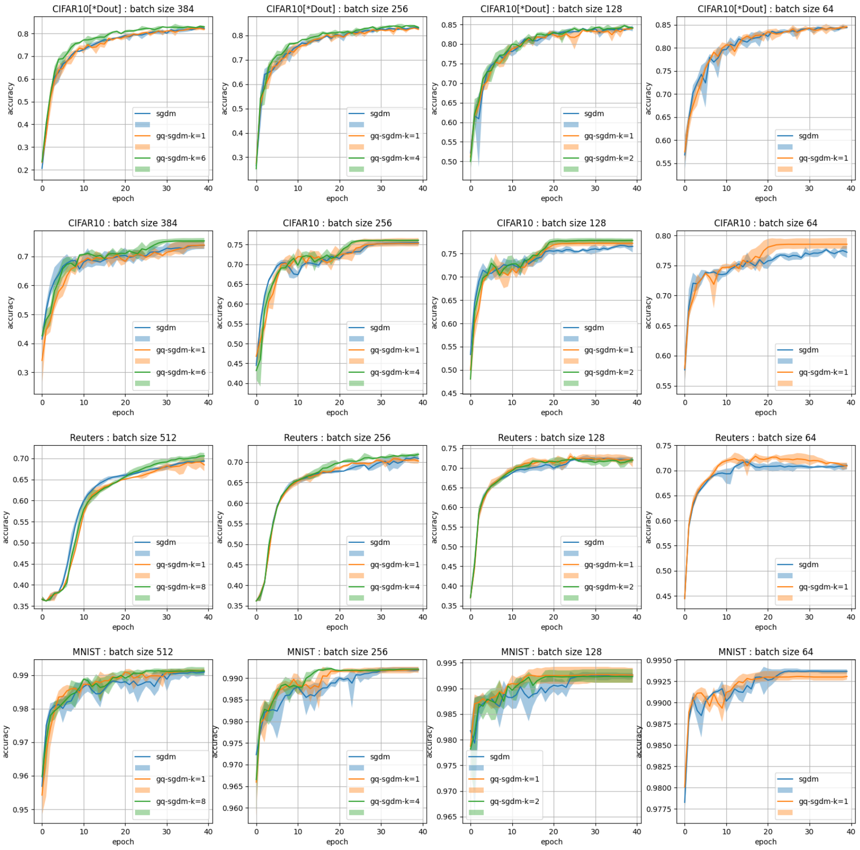

Figure 6: Results on CIFAR10 , Reuters dataset

5. Conclusion

We demonstrate the potential of emphasizing on gradients having more potential on training time. We achieve it by maintaining a active queue of gradients with time. Using our approach classification task was performed in several widely used datasets in computer vision and natural language processing domain. Robust techniques to extract the impactful gradients can be explored further while memory and run-time can be optimized for making the method widely applicable.

Appendix A.

Proof of Lemma 3.1
Using the momentum equation with $m_0 = 0$, $k > 0$ and $k \varepsilon \mathbb{Z}$.

$$\begin{aligned}
m_x &= \beta\, m_{x-1} \ + g_x \ = \ \beta^2 m_{x-2} \ + \ \beta g_{x-1} \ + \ g_x \\
m_x &= \ \beta^x m_0 \ + \ \beta^{x-1} g_1 \ + \beta^{x-2} g_2 \ + \ \ldots \ + \ \beta g_{x-1} \ + \ g_x \\
m_{N-1} &= \beta^{N-1} m_0 \ + \ \beta^{N-2}\, g_1 \ + \ \beta^{N-3}\, g_2 \ + \ \ldots \ + \ \beta g_{N-2} \ + \ g_{N-1} \\
&= \ \beta^{N-1} m_0 \ + \ u \ \left[\beta^{N-2} \ + \ \beta^{N-3} \ + \ \ldots \ + \ \beta \ + \ 1\right] \ = \beta^{N-1} m_0 \ + \ u \ \frac{\beta^{N-1} - 1}{\beta - 1}
\end{aligned}$$

$$\begin{aligned}
m_{N-1} = \ \beta^{N-1} m_0 \ + \ u\, \beta_{N-1} \ , \ \ taking \ \ \beta_x \ = \frac{\beta^x - 1}{\beta - 1} \\
m_x = \ \beta^x m_0 \ + \ u\, \beta_x \ , \ \ taking \ \ \beta_x \ = \frac{\beta^x - 1}{\beta - 1}
\end{aligned} \tag{8}$$

Eq. [8] can be used as a general formula for expanding the momentum values with time step, starting from initial value m_0 and expanding for the next x steps while u is repeated every time.

$$\begin{aligned}
m_N \ &= \beta m_{N-1} \ + \ C \ = \ \beta^N m_0 \ + \ u\beta\beta_{N-1} \ + \ C \\
&= \ u\beta\beta_{N-1} \ + \ C \ , \ with\ m_0 = 0 \\
m_{2N-1} &= \ \beta^{N-1} m_N \ + \ u\beta_{N-1} \\
m_{2N} \ &= \ \beta m_{2N-1} \ + \ C \ = \ \beta^N m_N \ + \ u\beta\beta_{N-1} \ + \ C \\
&= \ \left(\beta^N + 1\right)\left(u\beta\beta_{N-1} \ + \ C\right) \\
m_{3N} \ &= \beta\left(\beta^{N-1}\left(\left(\beta^N + 1\right)\left(u\beta\beta_{N-1} \ + \ C\right)\right) + \ \beta_{N-1} u\right) \ + \ C \\
&= \ \left(\beta^{2N} + \beta^N + 1\right)\left(\beta\beta_{N-1} u + C\right) \\
&= \beta_3^N \ \left(u\beta\beta_{N-1} + \ C\right) \\
m_{kN} \ &= \beta_k^N \ \left(u\beta\beta_{N-1} + \ C\right)
\end{aligned}$$

Proof of Lemma 3.2

$$\sigma = \sqrt{\mathbb{E}\,(x^2)\ -\ \mathbb{E}(x)}\ \ =\sqrt{\frac{(L-1)u^2+C^2}{L}\ -\ \left(\frac{(L-1)u\ +\ C}{L}\right)^2}\ \ =\ \sqrt{(L-1)}\ \left(\frac{u-C}{L}\right)$$

$$\Delta(u)\ =\ \frac{u-\mu}{\sigma}u = \frac{u\ -\ \frac{(L-1)u\ +\ C}{L}}{\sigma}u\ = \frac{u}{\sqrt{(L-1)}}$$

$$\Delta_\rho(u) = max\left(\frac{1}{\sqrt{(L-1)}}, \frac{1}{\rho}\right)\ u$$

$$taking, \phi\ = max\left(\frac{1}{\sqrt{(L-1)}}, \frac{1}{\rho}\right)$$

$$since, \rho > 1\ and\ L > 3\ ;\ \phi < 1$$

Proof of Lemma 3.3

Let’s consider a zero initiated queue of length L is being filled with values from [3]. Boosting with eq. [1] is possible if the queue is filled after initial L steps.

For $0 < t < L$, from eq. [8] with $m_0 = 0$, $m_L = \beta u$ and $\Delta_\rho(u) = u$

For $t > L$, using Lemma 3.2, $\Delta_\rho(u) = \phi u$

From eq. [8] -

$$\begin{aligned}
&Starting\quad from\ t = 0\ \ and\ expanding\ up\ to\ x^{th}\ step,\\
&m_x\quad\quad = \beta^x m_0\ +\ \beta_x u\ \ [here,\ x > L]\\
&Starting\quad from\ L^{th}\ (\ where,\ x > L)\ step\ \ and\ expanding\ for\ remaining\ x-L\ steps,\\
&m_x\quad\quad = \beta^{x-L} m_L\ +\ u\beta_{x-L}\\
&m_{N-1}\quad = \beta^{(N-1-L)} m_L\ +\ u\beta_{N-1-L}\ =\ \beta^{(N-1-L)} u\beta_L\ +\ u\beta_{N-1-L}\ \ [here,\ N-1\ >\ L]\\
&\Delta_\rho(m_{N-1})\ =\ \Delta_\rho\left(\ u\beta^{(N-1-L)}\beta_L\ +\ u\beta_{N-1-L}\ \right)\ = \beta^{N-1-L} u\beta_L\ +\ \beta_{(N-1-L)}\frac{u}{\rho}
\end{aligned}$$

$$taking,\ \ \gamma^0_x\ = \beta^{N-1-L}\beta_L + \ \beta_{(N-1-L)}\tfrac{1}{\rho}\ \ ,\ for\ 0 < t\ <\ L$$

$$and\ \ \ \gamma_x\ = \phi\beta^{N-1-L}\beta_L + \ \beta_{(N-1-L)}\tfrac{1}{\rho}\ \ ,\ for\ L < t\ ;$$

$$\begin{aligned}
\Delta_\rho(m_{N-1}) &=\ \gamma^0_{N-1}\ u\\
\Delta_\rho(m_N) &=\ u\beta\gamma^0_{N-1}\ +\ \rho C\\
\Delta_\rho(m_{2N-1}) &=\ \beta^{N-1} m_N\ +\ u\gamma_{N-1}\\
\Delta_\rho(m_{2N}) &=\ \beta\left(\beta^{N-1} m_N\ +\ u\gamma_{N-1}\ \right)\ +\ \rho C\ =\ \beta^N\left(u\beta\gamma^0_{N-1}\ +\ \rho C\right)\ + (\ u\beta\gamma_{N-1}\ +\ \rho C)\\
\Delta_\rho(m_{3N-1}) &=\ \beta^{N-1} m_{2N}\ +\ u\gamma_{N-1}\\
\Delta_\rho(m_{3N}) &=\ \beta\left(\beta^{N-1} m_{2N}\ +\ u\gamma_{N-1}\ \right)\ +\ \rho C\ = \beta^{2N}\left(u\beta\gamma^0_{N-1}\ +\ \rho C\right)\ +\ \left(\beta^N +\ 1\right)(u\beta\gamma_{N-1}\ +\ \rho C)\\
\Delta_\rho(m_{kN}) &=\ \beta^{(k-1)N}\left(u\beta\gamma^0_{N-1}\ +\ \rho C\right)\ +\ \left(\beta^{(k-2)N} + \beta^{(k-3)N}\ + ... + 1\right)(u\beta\gamma_{N-1}\ +\ \rho C)\\
&=\ \ \beta^{(k-1)N}\left(u\beta\gamma^0_{N-1}\ +\ \rho C\right)\ +\ \beta^N_{k-1}(u\beta\gamma_{N-1}\ +\ \rho C)
\end{aligned}$$

References

Agarwal, C., D'souza, D., & Hooker, S. (2022). Estimating example difficulty using variance of gradients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10368–10378.

Al Hasib, I. M., Sultana, S. S., Nyeen, I. Z., & Sabur, M. A. (2023). Boosting auxiliary task guidance: a probabilistic approach. *IAES International Journal of Artificial Intelligence*, *12*(1), 96.

Alain, G., Lamb, A., Sankar, C., Courville, A., & Bengio, Y. (2015). Variance reduction in sgd by distributed importance sampling. *arXiv preprint arXiv:1511.06481*.

Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., & Sivic, J. (2016). Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5297–5307.

Bengio, Y., Louradour, J., Collobert, R., & Weston, J. (2009). Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48.

Csiba, D., & Richtárik, P. (2018). Importance sampling for minibatches. *The Journal of Machine Learning Research*, *19*(1), 962–982.

De, S., Yadav, A., Jacobs, D., & Goldstein, T. (2016). Big batch sgd: Automated inference using adaptive batch sizes. *arXiv preprint arXiv:1610.05792*.

Du, Y., Czarnecki, W. M., Jayakumar, S. M., Farajtabar, M., Pascanu, R., & Lakshminarayanan, B. (2018). Adapting auxiliary losses using gradient similarity. *arXiv preprint arXiv:1812.02224*.

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization.. *Journal of machine learning research*, *12*(7).

Efe, U., Ince, K. G., & Alatan, A. (2021). Dfm: A performance baseline for deep feature matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4284–4293.

Elibol, M., Lei, L., & Jordan, M. I. (2020). Variance reduction with sparse gradients. *arXiv preprint arXiv:2001.09623*.

Faghri, F., Duvenaud, D., Fleet, D. J., & Ba, J. (2020). A study of gradient variance in deep learning. *arXiv preprint arXiv:2007.04532*.

Fan, Y., Tian, F., Qin, T., Bian, J., & Liu, T.-Y. (2017). Learning what data to learn. *arXiv preprint arXiv:1702.08635*.

Fang, C., Li, C. J., Lin, Z., & Zhang, T. (2018). Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. *Advances in Neural Information Processing Systems*, *31*.

Friedlander, M. P., & Schmidt, M. (2012). Hybrid deterministic-stochastic methods for data fitting. *SIAM Journal on Scientific Computing*, *34*(3), A1380–A1405.

Goyal, P., Dollar, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., & He, K. (2017). Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*.

Hinton, G., Srivastava, N., & Swersky, K. (2012). Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, *14*(8), 2.

Jiang, L., Zhou, Z., Leung, T., Li, L.-J., & Fei-Fei, L. (2018). Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International conference on machine learning*, pp. 2304–2313. PMLR.

Johnson, R., & Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, *26*.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., & Tang, P. T. P. (2016). On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kumar, M., Packer, B., & Koller, D. (2010). Self-paced learning for latent variable models. *Advances in neural information processing systems*, *23*.

Liu, Y., Gao, Y., & Yin, W. (2020). An improved analysis of stochastic gradient descent with momentum. *Advances in Neural Information Processing Systems*, *33*, 18261–18271.

Loshchilov, I., & Hutter, F. (2015). Online batch selection for faster training of neural networks. *arXiv preprint arXiv:1511.06343*.

Wang, Z., Ji, K., Zhou, Y., Liang, Y., & Tarokh, V. (2018). Spiderboost: A class of faster variance-reduced algorithms for nonconvex optimization. *arXiv*, *2018*.

Wang, Z., Ji, K., Zhou, Y., Liang, Y., & Tarokh, V. (2019). Spiderboost and momentum: Faster variance reduction algorithms. *Advances in Neural Information Processing Systems*, *32*.

Yin, D., Pananjady, A., Lam, M., Papailiopoulos, D., Ramchandran, K., & Bartlett, P. (2018). Gradient diversity: a key ingredient for scalable distributed learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 1998–2007. PMLR.

Zhdanov, F. (2019). Diverse mini-batch active learning. *arXiv preprint arXiv:1901.05954*.