This paper talks about Variance reduction using a momentum-based method (STORM). Variance reduction technique typically requires - carefully tuned learning rates and use of "mega-batches" to achieve improved results. STORM (The algorithm of this paper) - Does not require any batches and uses adaptive learning rate which enables simpler implementation and lesser hyper parameter tuning. Their technique for removing the batches uses a variant of momentum to achieve variance reduction in non-convex optimization.

SVRG algorithms have improved the convergence rate to critical points of non-convex SGD from $O(1/T^{1/4})$ to $O(1/T^{3/10})$ to $O(1/T^{1/3})$. Despite this improvement, SVRG has not seen as much success in practice in non-convex machine learning problems. Two potential issues that still prevail are the use of non-adaptive learning rates and reliance on giant batch sizes. This is where STORM steps in. STORM that stands for STOchastic Recursive Momentum, achieves variance reduction using a variant of the momentum term. Hence, our algorithm does not require a gigantic batch to compute. Storm achieves the optimal convergence rate of $O(1/T^{1/3})$, and it uses an adaptive learning rate schedule that will automatically adjust to the variance values of ∇f(xt, ξt).

The algorithm of storm is as follows:

Algorithm 1 STORM: STOchastic Recursive Momentum

Input: Parameters k, w, c, initial point $\boldsymbol{x}_1$
Sample ξ_1
$G_1 \leftarrow \|\nabla f(\boldsymbol{x}_1, \xi_1)\|$
$\boldsymbol{d}_1 \leftarrow \nabla f(\boldsymbol{x}_1, \xi_1)$
$\eta_0 \leftarrow \frac{k}{w^{1/3}}$
for $t = 1$ **to** T **do**
  $\eta_t \leftarrow \frac{k}{(w+\sum_{i=1}^{t} G_t^2)^{1/3}}$
  $\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{x}_t - \eta_t \boldsymbol{d}_t$
  $a_{t+1} \leftarrow c\eta_t^2$
  Sample ξ_{t+1}
  $G_{t+1} \leftarrow \|\nabla f(\boldsymbol{x}_{t+1}, \xi_{t+1})\|$
  $\boldsymbol{d}_{t+1} \leftarrow \nabla f(\boldsymbol{x}_{t+1}, \xi_{t+1}) + (1 - a_{t+1})(\boldsymbol{d}_t - \nabla f(\boldsymbol{x}_t, \xi_{t+1}))$
end for
Choose $\hat{\boldsymbol{x}}$ uniformly at random from $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$. (In practice, set $\hat{\boldsymbol{x}} = \boldsymbol{x}_T$).
return $\hat{\boldsymbol{x}}$

In words, Theorem 1 guarantees that STORM will make the norm of the gradients converge to 0 at a rate of $O(\frac{\ln T}{\sqrt{T}})$ if there is no noise, and in expectation at a rate of $\frac{2\sigma^{1/3}}{T^{1/3}}$ in the stochastic case. We remark that we achieve both rates automatically, without the need to know the noise level nor the need to tune stepsizes. Note that the rate when $\sigma \neq 0$ matches the optimal rate [3], which was previously only obtained by SVRG-based algorithms that require a "mega-batch" [8, 31].

STORM finds critical points in stochastic, smooth, non-convex problems. It also removes the need for batch-size and incorporates adaptive learning rates. Storm is substantially easier to tune. On CIFAR-10 with a ResNet32 architecture, it seems to be optimizing the objective in fewer iterations than baseline algorithms which has been shown in this paper experimentally.

References: *Cutkosky, Ashok, and Francesco Orabona. Momentum-based variance reduction in non-convex sgd. NeurIPS 2020.*