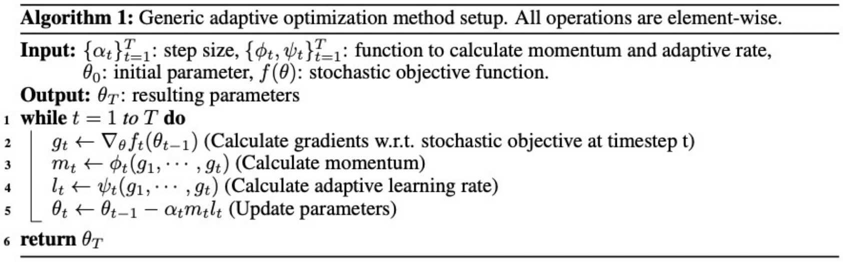
This paper talks about the improvements in already existing optimization algorithms such as ADAM or RMSProp using a method known as "**warmup**" wherein the training happens with a small learning rate for the first few epochs. The author claims that the lack of samples in the early stage of training learning rate has an undesirably large variance which leads to suspicious/ bad local optima. The paper introduces a new method known as Rectified Adam with the implementation of these warmups and states the motivation to do so.



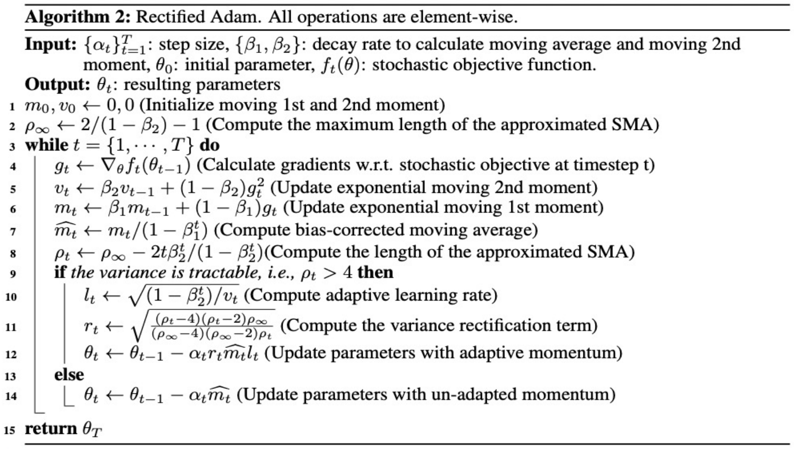
```
Algorithm 1: Generic adaptive optimization method setup. All operations are element-wise.
Input: ${\alpha_t}_{t=1}^T$: step size, ${\phi_t, \psi_t}_{t=1}^T$: function to calculate momentum and adaptive rate,
       $\theta_0$: initial parameter, $f(\theta)$: stochastic objective function.
Output: $\theta_T$: resulting parameters
while $t = 1$ to $T$ do
    $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Calculate gradients w.r.t. stochastic objective at timestep t)
    $m_t \leftarrow \phi_t(g_1, \cdots, g_t)$ (Calculate momentum)
    $l_t \leftarrow \psi_t(g_1, \cdots, g_t)$ (Calculate adaptive learning rate)
    $\theta_t \leftarrow \theta_{t-1} - \alpha_t m_t l_t$ (Update parameters)
return $\theta_T$
```

This is the previously available structure of an adaptive learning rate optimization algorithm. By specifying different phi's and theta's different algorithms are obtained.

The paper then experiments with two different models that are slightly modified versions of ADAM .

The first method is adam2k in which the learning rate is updated in the first 2000 iterations while momentum and other parameters are fixed, and the remaining iterations take place according to conventional ADAM. The second slightly modified version is where the value of epsilon is changed from 10^{-8} to 10^{-4}. Adam-2k: Additional 2k samples helped avoid the convergence problem of vanilla-Adam. Also, the additional samples prevent the gradient distribution from being distorted which shows that if there is sufficient data in the early stages the convergence problem can be avoided.

Adam-eps: Prevents the gradient distribution from being distorted. The seriousness of the convergence problem is much lesser compared to vanilla-Adam which proves that reducing the variance of the adaptive learning rate can solve the convergence problem.



```
Algorithm 2: Rectified Adam. All operations are element-wise.
Input: ${\alpha_t}_{t=1}^T$: step size, ${\beta_1, \beta_2}$: decay rate to calculate moving average and moving 2nd
       moment, $\theta_0$: initial parameter, $f_t(\theta)$: stochastic objective function.
Output: $\theta_t$: resulting parameters
$m_0, v_0 \leftarrow 0, 0$ (Initialize moving 1st and 2nd moment)
$\rho_\infty \leftarrow 2/(1-\beta_2) - 1$ (Compute the maximum length of the approximated SMA)
while $t = \{1, \cdots, T\}$ do
    $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Calculate gradients w.r.t. stochastic objective at timestep t)
    $v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g_t^2$ (Update exponential moving 2nd moment)
    $m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t$ (Update exponential moving 1st moment)
    $\widehat{m_t} \leftarrow m_t/(1-\beta_1^t)$ (Compute bias-corrected moving average)
    $\rho_t \leftarrow \rho_\infty - 2t\beta_2^t/(1-\beta_2^t)$(Compute the length of the approximated SMA)
    if the variance is tractable, i.e., $\rho_t > 4$ then
        $l_t \leftarrow \sqrt{(1-\beta_2^t)/v_t}$ (Compute adaptive learning rate)
        $r_t \leftarrow \sqrt{\frac{(\rho_t-4)(\rho_t-2)\rho_\infty}{(\rho_\infty-4)(\rho_\infty-2)\rho_t}}$ (Compute the variance rectification term)
        $\theta_t \leftarrow \theta_{t-1} - \alpha_t r_t \widehat{m_t} l_t$ (Update parameters with adaptive momentum)
    else
        $\theta_t \leftarrow \theta_{t-1} - \alpha_t \widehat{m_t}$ (Update parameters with un-adapted momentum)
return $\theta_T$
```

rt has a similar form to the heuristic linear warmup. i.e. setting r! as min t,T" /T"

This confirms the observation from earlier which shows warmup reduces variance

RAdam deactivates $\psi(.)$ when variance is divergent, thus avoiding instability, RAdam is independent of model architectures and can be combined with other stabilization techniques. On resnet18, resnet20, and cifar10 datasets it is observed that RAdam outperforms Adam in all three datasets. rt does make it slower than Adam in the first few epochs but converges faster after that. RAdam improves the robustness of model training by making the model less sensitive to the learning rate parameter.

References: *Liu, Liyuan, et al. On the variance of the adaptive learning rate and beyond. ICLR 2020.*