**Algorithm 1:** Here is the Gravity, our proposed optimization method with a kinematic approach. $\mathcal{N}$ is the normal distribution with a mean of $\mu$ and a standard deviation of $\sigma$. Also, $G$ is the gradient of the objective function*, J, w.r.t W.* The symbol $\oslash$ is the element-wise division (Hadamard division). This algorithm has three hyper-parameters whose recommended values are $l = 0.1$, $\alpha = 0.01$, $\beta = 0.9$. For easier implementation of the Gravity optimizer, its python implementation using TensorFlow's high-level API, Keras, is available in the Gravity GitHub repository.

**Require:** $l$: Learning Rate
**Require:** $\alpha$: Govern initial Step size
**Require:** $\beta$: Moving Average Parameter $\in [0,1]$
**Require:** $t_{max}$: maximum number of update steps

for each weight matrix $W^i$:
  $\mu \leftarrow 0$
  $\sigma \leftarrow \alpha / l$
  ${V_0}^i \leftarrow \mathcal{N}(\mu, \sigma)$
$t \leftarrow 0$
**while** $t < t_{max}$ **:**
  $t \leftarrow t + 1$
  $\hat{\beta} \leftarrow (\beta t + 1)/(t + 2)$
  for each weight matrix $W^i$**:**
    $G \leftarrow \partial J / \partial w$
    $m \leftarrow 1 \oslash max\big(abs(G)\big)$
    $\zeta \leftarrow G \oslash (1 + (G \oslash m)^2)$
    ${V_t}^i \leftarrow \hat{\beta} {V_{t-1}}^i + \big(1 - \hat{\beta}\big)\zeta$
    $W^i \leftarrow W^i - l {V_t}^i$