Input: ϵ_0, initial learning rate
Input: α, decay rate of learning rate
Input: β_1, 1st order moment for plain gradients
Input: β_2, 2nd order moment for squared gradients
Input: ζ, small constant to avoid zero division
Input: m, minibatch size
Input: k, epoch size
Input: θ, initial weights
Input: $\mathbf{X}$, training dataset inputs
Input: $\mathbf{y}$, training dataset targets
Initialize: $s \leftarrow 0$, accumulation variable for historical gradients
Initialize: $r \leftarrow 0$, accumulation variable for historical squared gradients
Initialize: $t \leftarrow 0$, counter of each gradient update
Initialize: $j \leftarrow 1$, current epoch
while $j \leq k$ **do**
  update learning rate $\epsilon_j \leftarrow \epsilon_0 + \alpha(\epsilon_{j-1} - \epsilon_0)$
  while *stopping criteria is not satisfied* **do**
    $\{\mathbf{x}^1 ... \mathbf{x}^m\}$, $\{\mathbf{y}^1 ... \mathbf{y}^m\} \leftarrow$ get a sample from $\mathbf{X}$ and $\mathbf{y}$ randomly
    calculate estimation of gradient $\hat{g} \leftarrow \frac{1}{m}\sum_{i=1}^{m} L(f(\mathbf{x}^i;\theta), \mathbf{y}^i)$
    accumulate historical graidents $s \leftarrow \beta_1 s + (1-\beta_1)\hat{g}$
    accumulate historical squared graidents $r \leftarrow \beta_2 r + (1-\beta_2)\hat{g} \odot \hat{g}$
    $t \leftarrow t + 1$
    apply bias correction to 1st order momentum $\hat{s} = \frac{s}{1-\beta_1^t}$
    apply bias correction to 2nd order momentum $\hat{r} = \frac{r}{1-\beta_2^t}$
    calculate step size $\Delta\theta \leftarrow -\epsilon_j \frac{\hat{s}}{\sqrt{\zeta + \hat{r}}}$
    update weights $\theta \leftarrow \theta + \Delta\theta$
  $j \leftarrow j + 1$ go to next epoch