Algorithm 1: The Minibatch Stochastic Gradient Descent Algorithm.

input : Initial value $\boldsymbol{\theta}^{(0)}$, Batch size m, Number of epochs T, Learning rate schedule $\eta_1, \eta_2, \ldots, \eta_T$

output: minimizer $\boldsymbol{\theta}^{(T)}$

for $t = 1$ **to** T **do**
  Split the data into n/m minibatches $\{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_{n/m}\}$
  for $b = 0$ **to** $n/m - 1$ **do**
    $\boldsymbol{\theta}^{(t,b+1)} := \boldsymbol{\theta}^{(t,b)} - \eta_t \cdot \frac{1}{m} \sum_{i \in \mathcal{B}_b} \nabla \ell(y_i, f(x_i; \boldsymbol{\theta}^{(t,b)})$
  end
  $\boldsymbol{\theta}^{(t+1,0)} := \boldsymbol{\theta}^{(t,n/m-1)}$
end

Algorithm 2: The AdaGrad Algorithm.

input : Initial value $\boldsymbol{\theta}^{(0)}$, Batch size m, Number of epochs T, Global learning rate η

output: minimizer $\boldsymbol{\theta}^{(T)}$

for $t = 1$ **to** T **do**
  Split the data into n/m minibatches $\{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_{n/m}\}$
  for $b = 0$ **to** $n/m - 1$ **do**
    Let $\boldsymbol{g} = \frac{1}{m} \sum_{i \in \mathcal{B}_b} \nabla \ell(y_i, f(x_i; \theta^{(t,b)})$.
    Accumulate squared gradient: $\boldsymbol{r} \leftarrow \boldsymbol{r} + \boldsymbol{g} \odot \boldsymbol{g}$
    Compute update: $\boldsymbol{\theta}^{(t,b+1)} := \boldsymbol{\theta}^{(t,b)} - \frac{\eta}{\sqrt{\boldsymbol{r}+\epsilon}} \odot \boldsymbol{g}$ (Division and square root applied element-wise; ϵ is a small number for numerical stability)
  end
  $\boldsymbol{\theta}^{(t+1,0)} := \boldsymbol{\theta}^{(t,n/m-1)}$
end

Algorithm 3: The RMSProp Algorithm.

input : Initial value $\boldsymbol{\theta}^{(0)}$, Batch size m, Number of epochs T, Global learning rate η, Decay rate ρ

output: minimizer $\boldsymbol{\theta}^{(T)}$

```
for t = 1 to T do
    Split the data into n/m minibatches {B_1, B_2, ..., B_{n/m}}
    for b = 0 to n/m − 1 do
        Let g = (1/m) Σ_{i∈B_b} ∇ℓ(y_i, f(x_i; θ^(t,b))).
        Accumulate squared gradient: r ← ρr + (1 − ρ)g ⊙ g
        Compute update: θ^(t,b+1) := θ^(t,b) − η/√(r+ϵ) ⊙ g (Division and
          square root applied element-wise; ϵ is a small number for
          numerical stability)
    end
    θ^(t+1,0) := θ^(t,n/m−1)
end
```

Where the listing writes the steps in plain text, they are, in the page's notation:

- Let $\boldsymbol{g} = \frac{1}{m}\sum_{i\in\mathcal{B}_b} \nabla\ell(y_i, f(x_i;\theta^{(t,b)}))$.
- Accumulate squared gradient: $\boldsymbol{r} \leftarrow \rho\boldsymbol{r} + (1-\rho)\boldsymbol{g}\odot\boldsymbol{g}$
- Compute update: $\boldsymbol{\theta}^{(t,b+1)} := \boldsymbol{\theta}^{(t,b)} - \frac{\eta}{\sqrt{\boldsymbol{r}+\epsilon}} \odot \boldsymbol{g}$
- $\boldsymbol{\theta}^{(t+1,0)} := \boldsymbol{\theta}^{(t,n/m-1)}$

Algorithm 4: The Adam Algorithm.

input : Initial value $\boldsymbol{\theta}^{(0)}$, Batch size m, Number of epochs T, Global learning rate η, Decay rate ρ_1, ρ_2

output: minimizer $\boldsymbol{\theta}^{(T)}$

```
for t = 1 to T do
    Split the data into n/m minibatches {B_1, B_2, ..., B_{n/m}}
    for b = 0 to n/m − 1 do
        Let g = (1/m) Σ_{i∈B_b} ∇ℓ(y_i, f(x_i; θ^(t,b))).
        Update biased 1st moment estimate: v ← ρ_1 v + (1 − ρ_1)g
        Update biased 2nd moment estimate: r ← ρ_2 r + (1 − ρ_2)g ⊙ g
        Correct bias in 1st moment: v̂ ← v/(1 − ρ_1^t)
        Correct bias in 2nd moment: r̂ ← r/(1 − ρ_2^t)
        Compute update: θ^(t,b+1) := θ^(t,b) − η/√(r̂+ϵ) ⊙ v̂
    end
    θ^(t+1,0) := θ^(t,n/m−1)
end
```

Where the listing writes the steps in plain text, they are, in the page's notation:

- Update biased 1st moment estimate: $\boldsymbol{v} \leftarrow \rho_1\boldsymbol{v} + (1-\rho_1)\boldsymbol{g}$
- Update biased 2nd moment estimate: $\boldsymbol{r} \leftarrow \rho_2\boldsymbol{r} + (1-\rho_2)\boldsymbol{g}\odot\boldsymbol{g}$
- Correct bias in 1st moment: $\hat{\boldsymbol{v}} \leftarrow \frac{\boldsymbol{v}}{1-\rho_1^t}$
- Correct bias in 2nd moment: $\hat{\boldsymbol{r}} \leftarrow \frac{\boldsymbol{r}}{1-\rho_2^t}$
- Compute update: $\boldsymbol{\theta}^{(t,b+1)} := \boldsymbol{\theta}^{(t,b)} - \frac{\eta}{\sqrt{\hat{\boldsymbol{r}}+\epsilon}} \odot \hat{\boldsymbol{v}}$