Gradient Descent Algorithms

ahwan034

August 2024

Link to GitHub:
https://github.com/allisonh0417/Gears-Gradient-Descent
Updated Aug 14, 2024

Algorithm 1 Adaptive Step Size Function

Input: Switch z, Current Error err, Step Size h, Input Gradients A, B, y_{curr}
Output: h, z
if $z = 0$ **then**
    $h_{\text{new}} \leftarrow h \times 1.1$ ▷ Increase step size
else
    $h_{\text{new}} \leftarrow h \times 0.9$ ▷ Decrease step size
end if
$y_{\text{new}} \leftarrow y_{\text{curr}} + h_{\text{new}} \times f(y_{\text{curr}}, A, B)$
$err_{\text{new}} \leftarrow \|A \times y_{\text{new}} - B\|$
if $err_{\text{new}} \leq err$ **then**
    $h \leftarrow h_{\text{new}}$ ▷ Only change step size if new estimated error is smaller than current error
else if $z = 0$ **then**
    $z \leftarrow 1$
else
    $z \leftarrow 0$
end if

Algorithm 2 Nesterov's Accelerated Gradient (NAG) Algorithm

Input: Input Gradients A, B, y_{curr}, Convergence $\epsilon = 10^{-6}$
Initialize: Step Size $h = \frac{1}{3\|A\|^2}$
while $err > \epsilon$ **do**
$y_{\text{prev}} \leftarrow y_{\text{curr}}$
$y_{\text{curr}} \leftarrow y_{\text{prev}} - h \times f(y_{\text{curr}}, A, B)$
$y_{\text{new}} \leftarrow y_{\text{curr}} + \frac{k-1}{k+2} \times (y_{\text{curr}} - y_{\text{prev}})$
$err \leftarrow \|Ay_{\text{curr}} - B\|$
end while

Algorithm 3 Nesterov's Accelerated Gradient (NAG) Algorithm Adaptive

Input: Input Gradients A, B, y_{curr}, Switch $z = 0$, Convergence $\epsilon = 10^{-6}$
Initialize: Step Size $h = \frac{1}{3\|A\|^2}$
while $err > \epsilon$ **do**
$y_{\text{prev}} \leftarrow y_{\text{curr}}$
$y_{\text{curr}} \leftarrow y_{\text{prev}} - h \times f(y_{\text{curr}}, A, B)$
$y_{\text{new}} \leftarrow y_{\text{curr}} + \frac{k-1}{k+2} \times (y_{\text{curr}} - y_{\text{prev}})$
$err \leftarrow \|Ay_{\text{curr}} - B\|$
▷ Adaptive step size adjustment
if $z = 0$ **then**
$h_{\text{new}} \leftarrow h \times 1.005$ ▷ Increase step size
else
$h_{\text{new}} \leftarrow h \times 0.9$ ▷ Decrease step size
end if
$y_{\text{prev\_test}} \leftarrow y_{\text{curr}}$ ▷ Test new potential step size
$y_{\text{curr\_test}} \leftarrow y_{\text{prev\_test}} - h_{\text{new}} \times f(y_{\text{curr}}, A, B)$
$y_{\text{new}} \leftarrow y_{\text{curr\_test}} + \frac{k-1}{k+2} \times (y_{\text{curr\_test}} - y_{\text{prev\_test}})$
$err_{\text{new}} \leftarrow \|Ay_{\text{new\_test}} - B\|$
if $err_{\text{new}} < err$ **then**
$h \leftarrow h_{\text{new}}$
$err \leftarrow err_{\text{new}}$
$y_{\text{curr}} \leftarrow y_{\text{new\_test}}$
$z \leftarrow 0$ ▷ Continue increasing step size
else
$z \leftarrow 1$ ▷ Start decreasing step size
end if
end while

$$x_k = y_{k-1} - s\nabla f(y_{k-1}),$$
$$y_k = x_k + \frac{k-1}{k+2}(x_k - x_{k-1}).$$

If $\epsilon > \|Ax - B\|$, then function converges,
where ϵ is the tolerance level, $\epsilon = 10^{-6}$.

Algorithm 4 Adaptive Step Size Algorithm

Input: Gradients A, B, y_{curr}, Switch z, Current Error err_{global}
Output: h, z
while $err_{\text{new}} \geq err_{\text{global}}$ **and** $t < 5$ **do**
  if $z = 0$ **then** ▷ Increase step size
    $h_{\text{new}} \leftarrow h \times 1.1$
    $y_{\text{new\_test}} \leftarrow y_{\text{curr}} + h_{\text{new}} \times f(y_{\text{curr}}, A, B)$
    $err_{\text{new}} \leftarrow \|A \times y_{\text{new\_test}} - B\|$
    if $err_{\text{new}} < err_{\text{local}}$ **then**
      $err_{\text{local}} \leftarrow err_{\text{new}}$
      $h \leftarrow h_{\text{new}}$
      $err_{\text{prev}} \leftarrow err_{\text{new}}$
    else if $err_{\text{new}} < err_{\text{global}}$ **then**
      $h \leftarrow h_{\text{new}}$
      break
    else if $err_{\text{prev}} < err_{\text{new}}$ **then**
      $z \leftarrow 1$
      $err_{\text{prev}} \leftarrow err_{\text{new}}$
    else
      $err_{\text{prev}} \leftarrow err_{\text{new}}$
    end if
  else ▷ Decrease step size
    $h_{\text{new}} \leftarrow h \times 0.9$
    $y_{\text{new\_test}} \leftarrow y_{\text{curr}} + h_{\text{new}} \times f(y_{\text{curr}}, A, B)$
    $err_{\text{new}} \leftarrow \|A \times y_{\text{new\_test}} - B\|$
    if $err_{\text{new}} < err_{\text{local}}$ **then**
      $err_{\text{local}} \leftarrow err_{\text{new}}$
      $h \leftarrow h_{\text{new}}$
      $err_{\text{prev}} \leftarrow err_{\text{new}}$
    else if $err_{\text{new}} < err_{\text{global}}$ **then**
      $h \leftarrow h_{\text{new}}$
      break
    else if $err_{\text{prev}} < err_{\text{new}}$ **then**
      $z \leftarrow 0$
      $err_{\text{prev}} \leftarrow err_{\text{new}}$
    else
      $err_{\text{prev}} \leftarrow err_{\text{new}}$
    end if
  end if
end while
$h_{\text{new}} \leftarrow h$

Algorithm 5 5th Order Gear's Method

while err $> \epsilon$ **do**
  $y_{\text{prev}} \leftarrow y_{\text{curr}}$
  switch (num) ▷ case equals n-th Order
    case 1: $y_{\text{curr}} \leftarrow y_{\text{prev}} + h_1 \cdot f(y_{\text{prev}}, A, B)$
    case 2: $y_{\text{curr}} \leftarrow \frac{4}{3} y_{\text{prev}} - \frac{1}{3} y_{\text{prev\_0}} + \frac{2}{3} h_1 \cdot f(y_{\text{prev}}, A, B)$
    case 3: $y_{\text{curr}} \leftarrow \frac{18}{11} y_{\text{prev}} - \frac{9}{11} y_{\text{prev\_1}} + \frac{2}{11} y_{\text{prev\_0}} + \frac{6}{11} h_1 \cdot f(y_{\text{prev}}, A, B)$
    case 4: $y_{\text{curr}} \leftarrow \frac{48}{25} y_{\text{prev}} - \frac{36}{25} y_{\text{prev\_2}} + \frac{16}{25} y_{\text{prev\_1}} - \frac{3}{25} y_{\text{prev\_0}} + \frac{12}{25} h_1 \cdot f(y_{\text{prev}}, A, B)$
    case 5: $y_{\text{curr}} \leftarrow \frac{300}{137} y_{\text{prev}} - \frac{300}{137} y_{\text{prev\_3}} + \frac{200}{137} y_{\text{prev\_2}} - \frac{75}{137} y_{\text{prev\_1}} + \frac{12}{137} y_{\text{prev\_0}} + \frac{60}{137} h_1 \cdot f(y_{\text{prev}}, A, B)$
  err $\leftarrow \|A \cdot y_{\text{curr}} - B\|$
  if err $>$ last_valid_err **then**
    $[h_1, z] \leftarrow$ new_step$(z, \text{err}, h_1, A, B, y_{\text{curr}})$
    num $\leftarrow 1$
  else
    last_valid_err $\leftarrow$ err
    num $\leftarrow \min(\text{num} + 1, 5)$
  end if
end while