THUẬT TOÁN TỐI ƯU ADAM

Đại học Bách Khoa Hà Nội
Giảng viên hướng dẫn: Thân Quang Khoát

04/2020

Nội dung

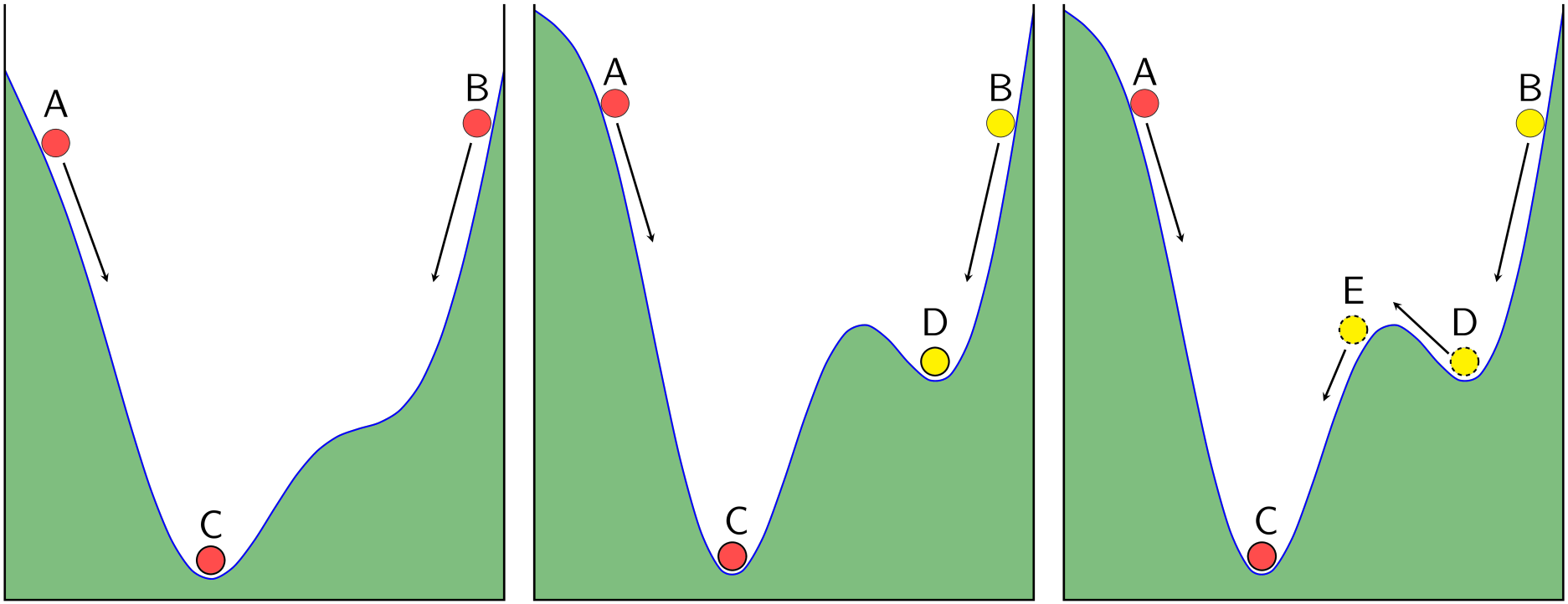
Các thuật toán tối ưu cơ sở

1. Gradient Descent
2. Momentum
3. Nesterow accelerated gradient (NAG)
4. Adagrad
5. Adadelta
6. RMSProp

Gradient Descent

- Đi ngược hướng đạo hàm để tìm điểm cực tiểu.
  - Batch Gradient Descent: $\theta_{t+1} = \theta_t - \eta \nabla_\theta f(\theta_t)$
  - Stochastic Gradient Descent (SGD): $\theta = \theta - \eta . \nabla_\theta J(\theta; x^{(i)}; y^{(i)})$
  - Mini-batch gradient descent: $\theta = \theta - \eta . J(\theta; x^{(i:i+n)}; y^{(i:i+n)})$
- Các vấn đề của Gradient Descent:
  - Chọn learning rate thích hợp.
  - Vượt qua điểm cực tiểu địa phương.

Momentum

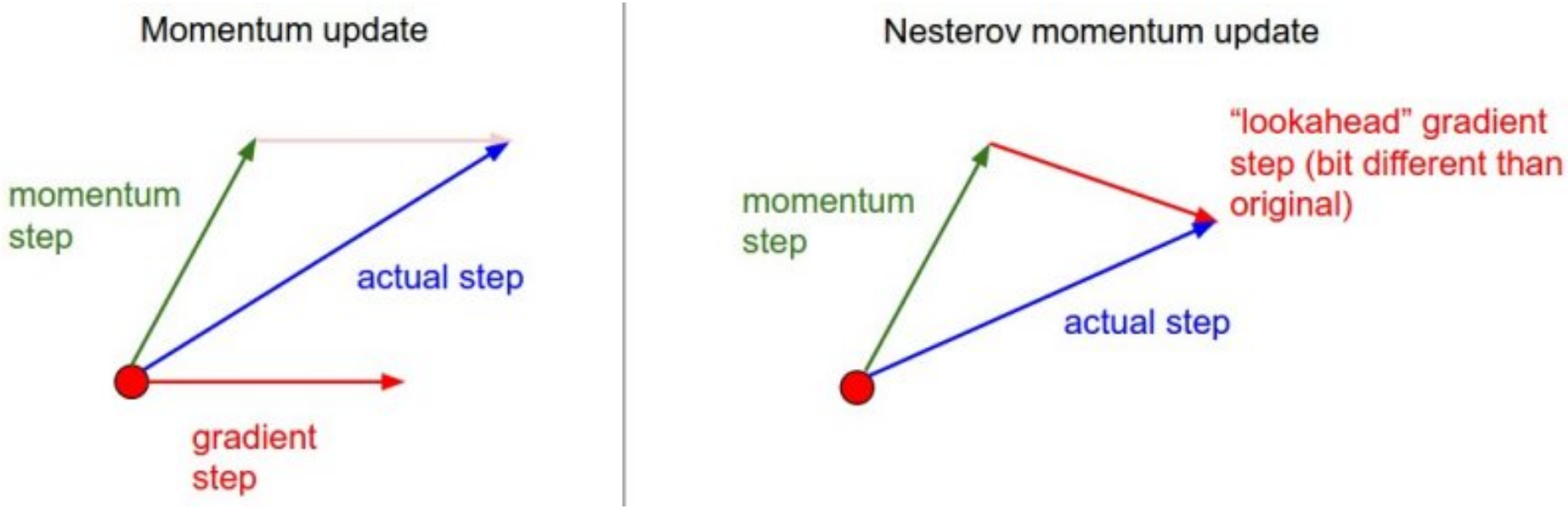


a) GD b) GD c) GD with momentum

- Thêm "đà" (vận tốc v_t) để cố gắng vượt qua cực tiểu địa phương.
- Công thức cập nhật ($\gamma < 1$):

$$v_t = \gamma v_{t-1} + \eta \nabla_\theta J(\theta)$$
$$\theta = \theta - v_t$$

Nesterow accelerated gradient (NAG)



- Dự đoán trước 1 bước đi để hội tụ nhanh hơn.
- Công thức cập nhật:

$$v_t = \gamma v_{t-1} + \eta \nabla_\theta J(\theta - \gamma v_{t-1})$$
$$\theta = \theta - v_t$$

Adagrad

- Gradient descent áp dụng 1 learning rate cho tất cả các tham số. Thực tế, đặc biệt với các bài toán dữ liệu thưa, các tham số cần cập nhật 1 bước xải lớn (nhỏ) khác nhau.
- AdaGrad sử dụng learning rate khác nhau cho mỗi tham số θ_i tại mỗi bước t.
- AdaGrad sinh learning rate cho mỗi tham số θ_i tại bước t dựa vào gradient quá khứ đã được tính toán cho θ_i.

Adagrad

- Gọi:
  - g_t là gradient tại bước t.
  - $g_{t,i}$ là đạo hàm riêng của hàm mục tiêu theo tham số θ_i tại bước thứ t: $g_{t,i} = \nabla_\theta J(\theta_{t,i})$
  - $G_t \in \mathbb{R}^{dxd}$ là một ma trận đường chéo mà các phần tử trên đường chéo i, i là tổng bình phương của các gradient theo θ_i cho đến bước thứ t
  - G_t gồm tổng bình phương các gradient quá khứ của tất cả các tham số θ dọc theo đường chéo
- Công thức cập nhật: $\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii}+\varepsilon}} . g_{t,i}$
- Hay: $\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t+\varepsilon}} \odot g_t$

AdaDelta

- Adagrad tích luỹ gradient ở mẫu số lớn dần lên trong quá trình training. Điều này làm cho learning ratethu hẹp lại và dần trở lên vô cùng nhỏ, tại thời điểm đó, thuật toán không còn có thể học được nữa.
- Thay vì tích luỹ bình phương gradient quá khứ, Adadelta hạn chế tích luỹ quá khứ.
- Đưa ra khái niệm running average $E[g^2]_t$ tại thời điểm t chỉ phụ thuộc vào trung bình trước và gradient hiện tại: $E[g^2]_t = \gamma E[g^2]_{t-1} + (1-\gamma)g_t^2$
- Công thức cập nhật:

$$\Delta\theta_t = -\frac{RMS[\Delta\theta]_{t-1}}{RMS[g]_t} g_t$$

$$\theta_{t+1} = \theta_t + \Delta\theta_t$$

RMSProp

- RMSProp là phương pháp điều chỉnh learning rate được đưa ra bởi Geoff Hinton. RMSProp và Adadelta phát triển độc lập cùng lúc xuất phát từ nhu cầu giải quyết triệt để learning rate giảm dần của Adagrad.
- Công thức cập nhật:

$$E[g^2]_t = 0.9[g^2]_{t-1} + 0.1g_t^2$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \varepsilon}} g_t$$

Thuật toán tối ưu Adam

1. Thuật toán Adam
2. Bias correction
3. Sự hội tụ của Adam

Thuật toán Adam

Require: α: Stepsize;
Require: $\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates for the moment estimates;
Require: $f(\theta)$: Stochastic objective function with parameters θ;
Require: θ_0: Initial parameter vector;
$m_0 \leftarrow 0$; $v_0 \leftarrow 0$; $t \leftarrow 0$;
while θ_t *not converged* **do**
$t \leftarrow t + 1$;
$g_t \leftarrow \nabla_\theta(\theta_{t-1})$;
$m_t \leftarrow \beta_1.m_{t-1} + (1 - \beta_1).g_t$;
$v_t \leftarrow \beta_2.v_{t-1} + (1 - \beta_2).g_t^2$;
$\hat{m}_t \leftarrow m_t/(1 - \beta_1^t)$;
$\hat{v}_t \leftarrow v_t/(1 - \beta_2^t)$;
$\theta_t \leftarrow \theta_{t-1} - \alpha.\hat{m}_t/(\sqrt{\hat{v}_t} + \varepsilon)$;
end
Result: Trả về tham số θ_t

Bias correction

- Adam ước lượng moment cấp 1 và moment cấp 2 của gradient để tính toán learning rate.
- Để ước lượng các moment, Adam tận dụng *exponentially moving average* của gradient và bình phương gradient, với *decay rate* là β_1 và β_2.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

Với m_t và v_t được gọi là *moving averages*
- Mục tiêu :

$$\mathbb{E}[m_t] = \mathbb{E}[g_t]$$

$$\mathbb{E}[v_t] = \mathbb{E}[g_t^2]$$

Bias correction

Ta khai triển v_t như sau (tương tự với m_t):

$$
\begin{aligned}
v_0 &= 0 \\
v_1 &= \beta_2 v_0 + (1-\beta_2) g_1^2 = (1-\beta_2) g_1^2 \\
v_2 &= \beta_2 v_1 + (1-\beta_2) g_2^2 = \beta_2 (1-\beta_2) g_1^2 + (1-\beta_2) g_2^2 \\
&\dots \\
v_t &= \beta_2 v_{t-1} + (1-\beta_2) g_t^2 = (1-\beta_2) \sum_{i=1}^{t} \beta_2^{t-i} . g_i^2
\end{aligned}
$$

Bias correction

- Ta có:

$$\begin{aligned}\mathbb{E}[v_t] &= \mathbb{E}[(1-\beta_2)\sum_{i=1}^{t}\beta_2^{t-i}.g_i^2] \\ &= \mathbb{E}[g_t^2].(1-\beta_2)\sum_{i=1}^{t}\beta_2^{t-i} + \zeta \\ &= \mathbb{E}[g_t^2].(1-\beta_2^t) + \zeta\end{aligned}$$

- Cần điều chỉnh giá trị ước lượng của v_t để cho kỳ vọng của nó xấp xỉ moment cấp 2. Sở dĩ có sự sai lệch đại lượng $(1-\beta_2^t)$ là do ta đã khởi tạo $v_0 = 0$.
- Công thức hiệu chỉnh:

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

Sự hội tụ của Adam

- Kingma và LeiBa đã có những sai sót trong phân tích sự hội tụ của thuật toán Adam. Trong bài báo của mình, các tác giả sử dụng các ký hiệu sau:
  - $f_1(\theta), f_2(\theta), ..., f_T(\theta)$ là một dãy các hàm chi phí và lồi tại vòng lặp tương ứng $1, 2, ..., T$.
  - $g_t := \nabla f_t(\theta_t)$ và gọi $g_{t,i}$ là phần tử thứ i của vector gradient g_t
  - $g_{1:t,i} = [g_{1,i}, g_{2,i}, ..., g_{t,i}] \in \mathbb{R}^t$ và $\gamma := \frac{\beta_1^2}{\sqrt{\beta_2}}$
  - θ^* là tham số tối ưu của bài toán, θ_t là tham số dự đoán được tại vòng lặp thứ t
- Tổng sai lệch của mô hình qua T vòng lặp:

$$R(T) = \sum_{t=1}^{T} [f_t(\theta_t) - f_t(\theta^*)]$$

Sự hội tụ của Adam

Định lý

Giả sử rằng các hàm f_t là có gradient bị chặn, $||\nabla f_t(\theta)||_2 \leq G$, $||\nabla f_t(\theta)||_\infty \leq G_\infty$ với mọi tham số $\theta \in \mathbb{R}^d$, và khoảng cách giữa 2 tham số θ_t bất kỳ được sinh ra bởi Adam là bị chặn, $||\theta_n - \theta_m||_2 \leq D$, $||\theta_m - \theta_n||_\infty \leq D_\infty$ với mọi m, n bất kỳ thuộc $\{1, ..., T\}$, và $\beta_1, \beta_2 \in [0, 1)$ thoả mãn $\frac{\beta_1^2}{\sqrt{\beta_2}} < 1$. Đặt $\alpha_t = \frac{\alpha}{\sqrt{t}}$ và $\beta_{1,t} = \beta_1 \lambda^{t-1}, \lambda \in (0, 1)$. Thì ta có:

$$R(T) \leq \frac{D^2}{2\alpha(1-\beta_1)} \sum_{i=1}^{d} \sqrt{T\hat{v}_{T,i}} + \frac{\alpha(1+\beta_1)G_\infty}{(1-\beta_1)\sqrt{1-\beta_2}(1-\gamma)^2} \sum_{i=1}^{d} ||g_{1:T,i}||_2$$
$$+ \sum_{i=1}^{d} \frac{D_\infty^2 G_\infty \sqrt{1-\beta_2}}{2\alpha(1-\beta_1)(1-\lambda)^2}$$

Sự hội tụ của Adam

- Các bổ đề chứng minh chưa chính xác:

Bổ đề

Giả sử $||g_t||_2 \leq G, ||g_t||_\infty \leq G_\infty$, *thì:*

$$\sum_{t=1}^{T} \sqrt{\frac{g_{t,i}^2}{t}} \leq 2G_\infty ||g_{1:T,i}||_2$$

Bổ đề

Giả sử $\gamma < 1$ *và* $||g_t||_2 \leq G$, $||g_t||_\infty \leq G_\infty$, *thì:*

$$\sum_{t=1}^{T} \frac{\hat{m}_{t,i}^2}{\sqrt{t\hat{v}_{t,i}}} \leq \frac{2}{1-\gamma} \frac{1}{\sqrt{1-\beta_2}} ||g_{1:T,i}||_2$$

Sự hội tụ của Adam

- Trong báo cáo của mình tại ICLR 2018, Sashank J. Reddi, Satyen Kate và SanjivKumar đã chứng minh thuật toán Adam không hội tụ và đề xuất ra các mở rộng, cải tiến.
- Các tác giả đưa ra khung chung cho các phương pháp adaptive (thích nghi) để cung cấp cái nhìn sâu sắc về sự khác biệt giữa các phương pháp adaptive:

Input: $x_1 \in \mathbb{F}$, step size $\{\alpha_t\}_t^T$, sequence of functions $\{\phi_t\psi\}$;
for $t = 1$ **to** T **do**
  $g_t = \nabla f_t(x_t)$;
  $m_t = \phi_t(g_1, ..., g_t)$ and $V_t = \psi_t(g_1, ..., g_t)$;
  $\hat{x}_{t+1} = x_t - \alpha_t m_t / \sqrt{V_t}$;
  $x_{t+1} = \prod_{\mathbb{F}, \sqrt{V_t}}(\hat{x}_{t+1})$;
end

Algorithm 2: Generic Adaptive Method Setup

Sự hội tụ của Adam

- Với gradient descent:

$$\phi_t(g_1, ..., g_t) = g_t$$

$$\psi_t(g_1, ..., g_t) = \mathbb{I}$$

- Với AdaGrad:

$$\phi_t(g_1, ..., g_t) = g_t$$

$$\psi_t(g_1, ..., g_t) = \frac{diag(\sum_{i=1}^{t} g_i^2)}{t}$$

- Với Adam:

$$\phi_t(g_1, ..., g_t) = (1 - \beta_1) \sum_{i=1}^{t} \beta_1^{t-1} g_i$$

$$\psi_t(g_1, ..., g_t) = (1 - \beta_2) diag(\sum_{i=1}^{t-1} g_i^2)$$

Sự hội tụ của Adam

- Các tác giả thấy rằng vấn đề chính dẫn đến sự không hội tụ của Adam nằm ở đại lượng sau:

$$\Gamma_{t+1} = \left(\frac{\sqrt{V_{t+1}}}{\alpha_{t+1}} - \frac{\sqrt{V_t}}{\alpha_t} \right)$$

- Đại lượng này đo lường sự thay đổi của nghịch đảo learning rate cho các phương pháp thích nghi theo thời gian.
- Dễ thấy rằng với SGD và AdaGrad thì $\Gamma_t \geq 0$ với mọi $t \in [T]$. Tuy nhiên, điều này là chưa hẳn với các phương pháp sử dụng trung bình trượt theo cấp số nhân như Adam và RMSProp. Γ_t có thể bất ổn định với $t \in [T]$. Vi phạm này có thể dẫn đến sự không hội tụ của Adam và RMSProp.

Sự hội tụ của Adam

- Xét f_t là các hàm tuyến tính và $\mathbb{F} = [-1, 1]$. Xét dãy hàm:

$$f_t(x) = \begin{cases} Cx, & \text{Nếu t mod } 3 = 1 \\ -x, & \text{Các trường hợp khác} \end{cases}$$

Với $C \geq 2$. Xét thuật toán Adam với các tham số sau:

$$\beta_1 = 0, \beta_2 = \frac{1}{1 + C^2}, \alpha_t = \frac{\alpha}{\sqrt{t}}, \alpha < \sqrt{1 - \beta_2}$$

- Kết quả chứng minh: $R(T) \geq (2C - 4)T/3$
- Vì $C \geq 2$, nên mất mát sẽ lớn dần và $R_T/T \nrightarrow 0$ khi $T \to \infty$

Các mở rộng của Adam

1. **AdaMax**
2. **Nadam**
3. **AMSGrad**

Adamax

- Sử dụng chuẩn L^2 thành chuẩn L^p cho quy tắc cập nhật của Adam.
- Các biến thể như vậy không ổn định cho p lớn.
- Tuy nhiên, trong trường hợp đặc biệt khi $p \to \infty$ thì ta lại thu được một thuật toán ổn định đáng ngạc nhiên.
- Trong chuẩn L^p, tại bước thứ t tỉ lệ nghịch với $v_t^{1/p}$, với:

$$v_t = \beta_2^p v_{t-1} + (1-\beta_2^p)|g_t|^p = (1-\beta_2^p)\sum_{i=1}^{t} \beta_2^{p(t-i)}.|g_i|^p$$

Adamax

Require: α: Stepsize;
Require: $\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates ;
Require: $f(\theta)$: Stochastic objective function with parameters θ;
Require: θ_0: Initial parameter vector;
$m_0 \leftarrow 0$ (Initialize 1^{st} moment vector) ;
$u_0 \leftarrow 0$ (Initialize the exponentially weighted infinit norm) ;
$t \leftarrow 0$ (Initialize timestep) ;
while θ_t *not converged* **do**
    $t \leftarrow t + 1$;
    $g_t \leftarrow \nabla_\theta(\theta_{t-1})$;
    $m_t \leftarrow \beta_1.m_{t-1} + (1 - \beta_1).g_t$ (Update biased first moment estimate);
    $u_t \leftarrow max(\beta_2.u_{t-1}, |g_t|)$ (Update the exponentially weighted infinity norm);
    $\theta_t \leftarrow \theta_{t-1} - (\alpha/(1 - \beta_1^t)).m_t/u_t$ (Update parameters) ;
end

Adamax

Lưu ý rằng thuật ngữ phân rã ở đây được tham số hoá là β_2^p thay vì β_2. Bây giờ, xét $p \to \infty$ và định nghĩa $u_t = lim_{p\to\infty}(v_t)^{1/p}$, thì:

$$
\begin{aligned}
u_t = lim_{p\to\infty}(v_t)^{1/p} &= \lim_{p\to\infty} \left((1-\beta_2^p) \sum_{i=1}^{t} \beta_2^{p(t-i)} . |g_i|^p \right)^{1/p} \\
&= \lim_{p\to\infty} (1-\beta_2^p)^{1/p} \left(\sum_{i=1}^{t} \beta_2^{p(t-i)} . |g_i|^p \right)^{1/p} \\
&= \lim_{p\to\infty} \left(\sum_{i=1}^{t} \beta_2^{p(t-i)} . |g_i|^p \right)^{1/p} \\
&= max(\beta_2^{t-1}|g_1|, \beta_2^{t-2}|g_2|, ..., \beta_2|g_{t-1}|, |g_t|)
\end{aligned}
$$

Ta có công thức đệ quy đơn giản tương ứng là:

$$u_t = max(\beta_2 . u_{t-1}, |g_t|)$$

Nadam

- Nadam (Nesterov-accelerated Adaptive Moment Estimation) là một thuật toán kết hợp Adam và NAG. Để phát triển Nadam, ta phải điều chỉnh giá trị momentum m_t trong Adam.
- Công thức cập nhật:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \left(\beta_1 \hat{m}_t + \frac{(1-\beta_1) g_t}{1-\beta_1^t} \right)$$

AMSGrad

- Như đã thảo luận bên trên, trong một số trường hợp, các phương pháp ước lượng learning rate thích ứng không hội tụ đến giải pháp tối ưu, thậm chí kết quả còn kém hơn SGD với momentum.
- Để giải quyết vấn đề này, thuật toán AMSGrad được đề xuất. Thuật toán sử dụng giá trị lớn nhất của các bình phương gradient trong quá khứ, thay vì EMA để cập nhật tham số v_t được định nghĩa tương tự như trong thuật toán Adam:

$$\hat{v}_t = max(\hat{v}_{t-1}, v_t)$$

AMSGrad

- Công thức cập nhật:

$$
\begin{aligned}
m_t &= \beta_1 m_{t-1} + (1-\beta_1) g_t \\
v_t &= \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \\
\hat{v}_t &= max(\hat{v}_{t-1}, v_t) \\
\theta_{t+1} &= \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} m_t
\end{aligned}
$$

Thực nghiệm

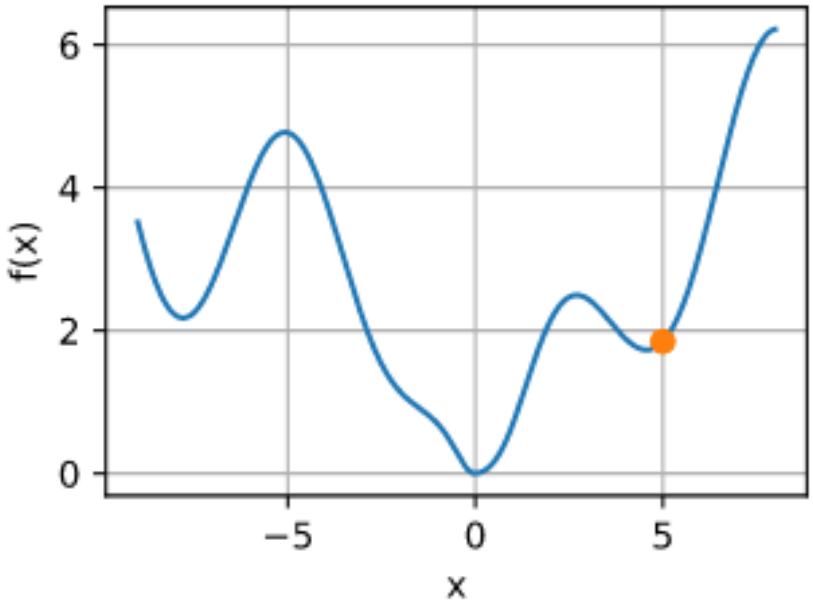
1. **Thực nghiệm Adam với các thuật toán cơ sở**
2. **Thực nghiệm Adam với AMSGrad**

Thực nghiệm Adam với các thuật toán cơ sở

- Thực nghiệm trên:
  - $f(x) = log(1 + (\mid x \mid)^{2+sinx}))$
  - $f(x) = (2 + \frac{sin50x}{50})(arctanx)^2$
  - Neural network với bộ dữ liệu MNIST

Thực nghiệm Adam với các thuật toán cơ sở

- Khởi tạo $x_0 = 2, \alpha = 0.01, \beta_1 = 0.9, \beta_2 = 0.99$:

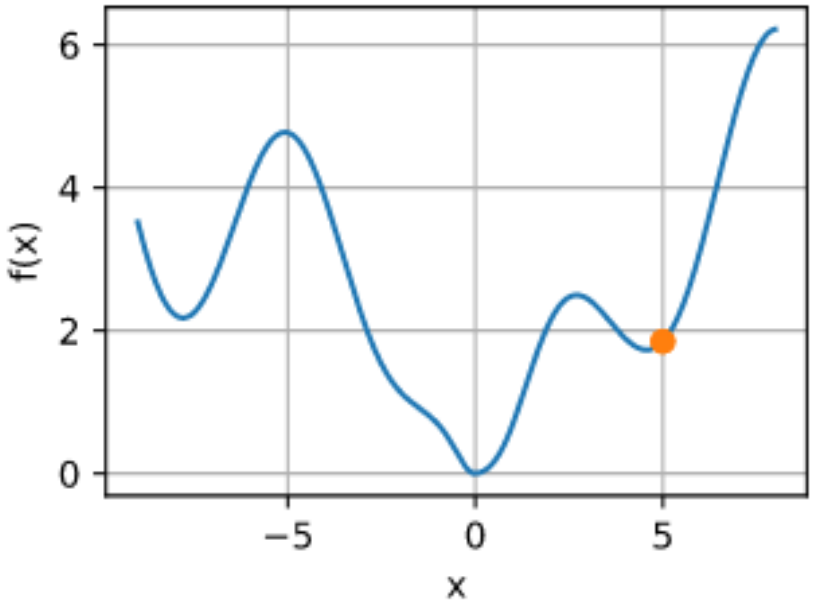



```
epoch  868 x: [-1.06588016e-08]
Gradient descent [-1.06588016e-08]
epoch 265 x [-2.17762022e-06]
Momentum [-2.17762022e-06]
epoch 267 x [-8.78597066e-07]
Nag [-8.78597066e-07]
epoch 9999 x [0.33193342]
Adagrad [0.33193342]
epoch 243 x [-4.99176985e-07]
RMSprop [-4.99176985e-07]
Adam: epoch 569 x [-7.69715167e-07]
Adam [-7.69715167e-07]
```

- Kết luận: Các thuật toán đều hội tụ tốt.

Thực nghiệm Adam với các thuật toán cơ sở

- Khởi tạo $x_0 = 8, \alpha = 0.01, \beta_1 = 0.9, \beta_2 = 0.99$:



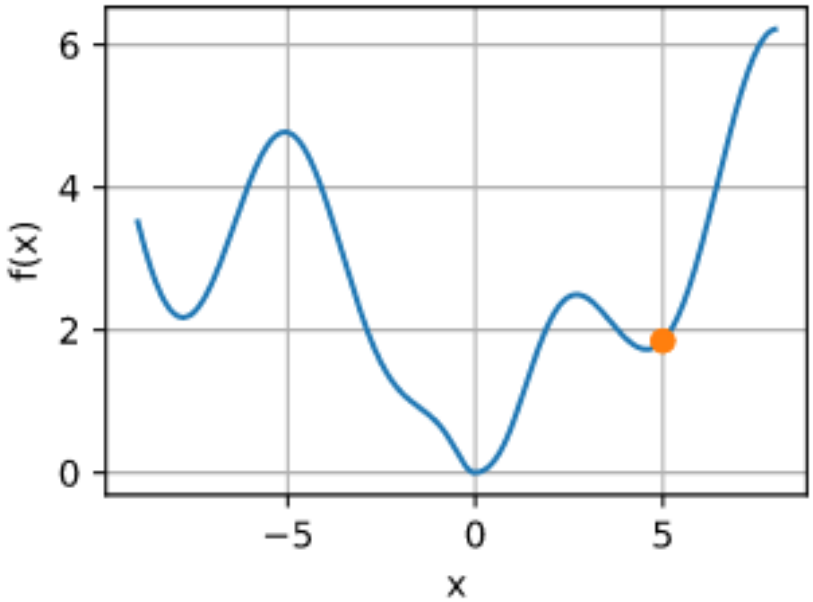

```
epoch  1433 x: [4.56611984]
Gradient descent [4.56611984]
epoch 296 x [4.56612179]
Momentum [4.56612179]
epoch 302 x [4.56611949]
Nag [4.56611949]
epoch 9999 x [5.64208075]
Adagrad [5.64208075]
epoch 382 x [4.56611898]
RMSprop [4.56611898]
Adam: epoch 1110 x [4.56611936]
Adam [4.56611936]
```

-

Kết luận: Các thuật toán đều hội tụ tới điểm cực tiểu địa phương, không tới được điểm cực tiểu toàn cục.

Thực nghiệm Adam với các thuật toán cơ sở

- Khởi tạo Khởi tạo $x_0 = 8, \alpha = 2, \beta_1 = 0.9, \beta_2 = 0.99$:

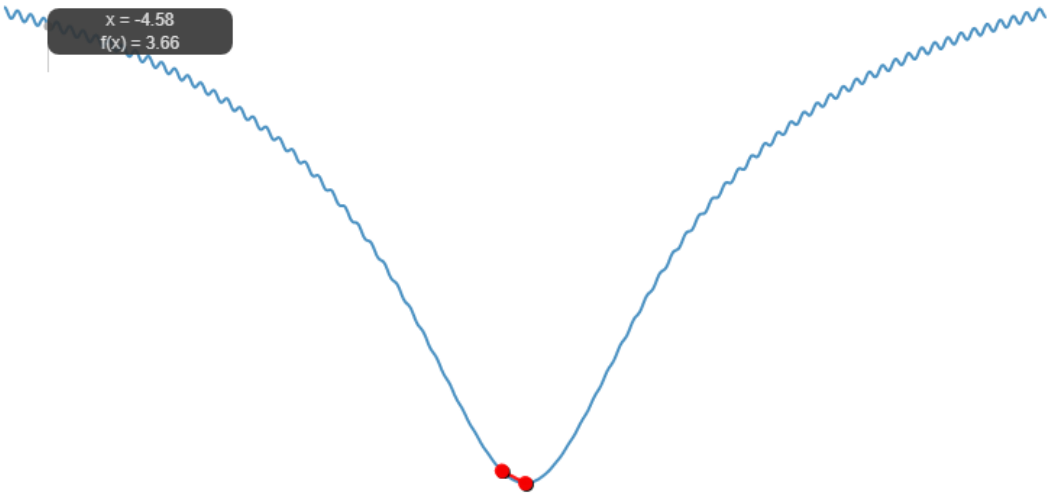



```
epoch  9999 x: [5.0106917]
Gradient descent [5.0106917]
epoch 9999 x [31166.58848496]
Momentum [31166.58848496]
epoch 9999 x [12.18847365]
Nag [12.18847365]
epoch 12 x [4.56611898]
Adagrad [4.56611898]
epoch 9999 x [0.90725303]
RMSprop [0.90725303]
Adam: epoch 303 x [-8.4867482e-07]
Adam [-8.4867482e-07]
```

- Kết luận: Thuật toán Adam hội tụ được đến cực tiểu toàn cục trong trường hợp này khi tăng α lên 2.

Thực nghiệm Adam với các thuật toán cơ sở

- Khởi tạo $x_0 = 4.03, \alpha = 0.06, \beta_1 = 0.9, \beta_2 = 0.99$:

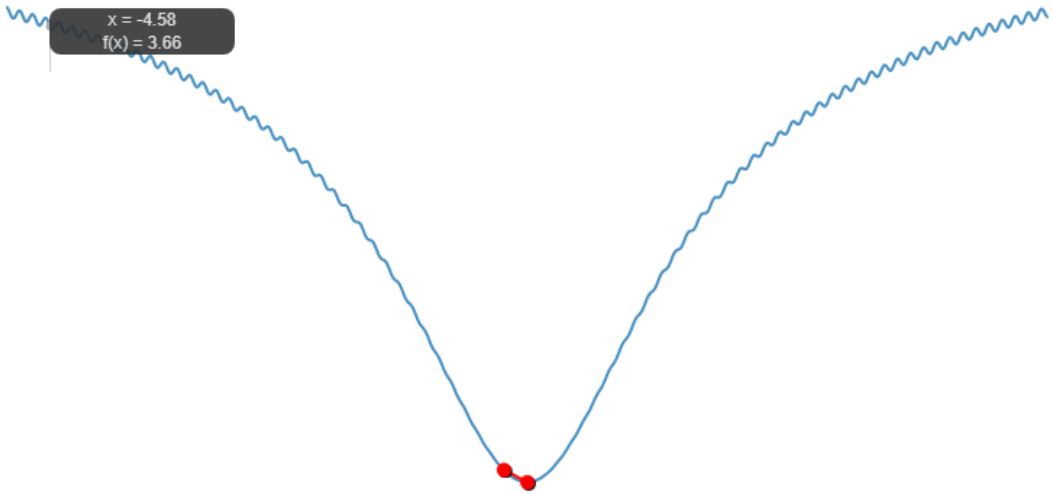



```
epoch  9999 x: [2.08558813]
Gradient descent [2.08558813]
epoch 317 x [-6.59583539e-07]
Momentum [-6.59583539e-07]
epoch 179 x [-1.52504801e-06]
Nag [-1.52504801e-06]
epoch 50 x [3.98626395]
Adagrad [3.98626395]
epoch 158 x [-4.97920382e-07]
RMSprop [-4.97920382e-07]
Adam: epoch 237 x [3.98626386]
Adam [3.98626386]
```

- Kết luận: Các thuật toán khác hội tụ đến cực tiểu toàn cục nhưng Adam và Adagrad không hội tụ được đến cực tiểu toàn cục.

Thực nghiệm Adam với các thuật toán cơ sở

- Tăng learning rate lên 2:

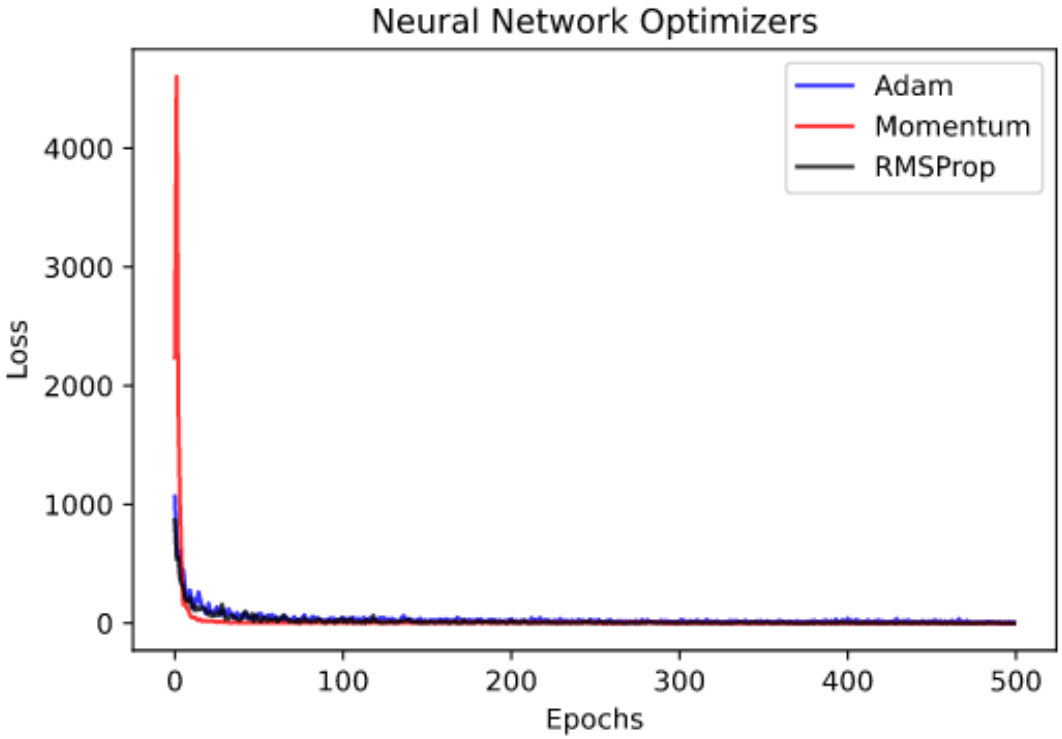



```
epoch  9999 x: [-456.60608974]
Gradient descent [-456.60608974]
epoch 9999 x [930.83453448]
Momentum [930.83453448]
epoch 9999 x [-119.53706345]
Nag [-119.53706345]
epoch 148 x [-5.04527632e-07]
Adagrad [-5.04527632e-07]
epoch 9999 x [329.12282844]
RMSprop [329.12282844]
Adam: epoch 346 x [-5.26884698e-07]
Adam [-5.26884698e-07]
```

- Kết luận: Vì learning rate quá lớn nên trong 10000 epoch momentum chưa kịp hội tụ còn adam đã hội tụ tốt.

Thực nghiệm Adam với các thuật toán cơ sở

Thí nghiệm so sánh các thuật toán tối ưu sử dụng bộ dữ liệu chữ viết tay MNIST . Bài toán phân loại các vector ảnh 784 chiều vào 10 lớp
Sử dụng 1 neural network với 2 hidden layer 256 unit .
Hyper-parameter : $\beta_1 = 0.9, \beta_2 = 0.99, \alpha = 0.01$, batch_size=128 , epoch $= 500$



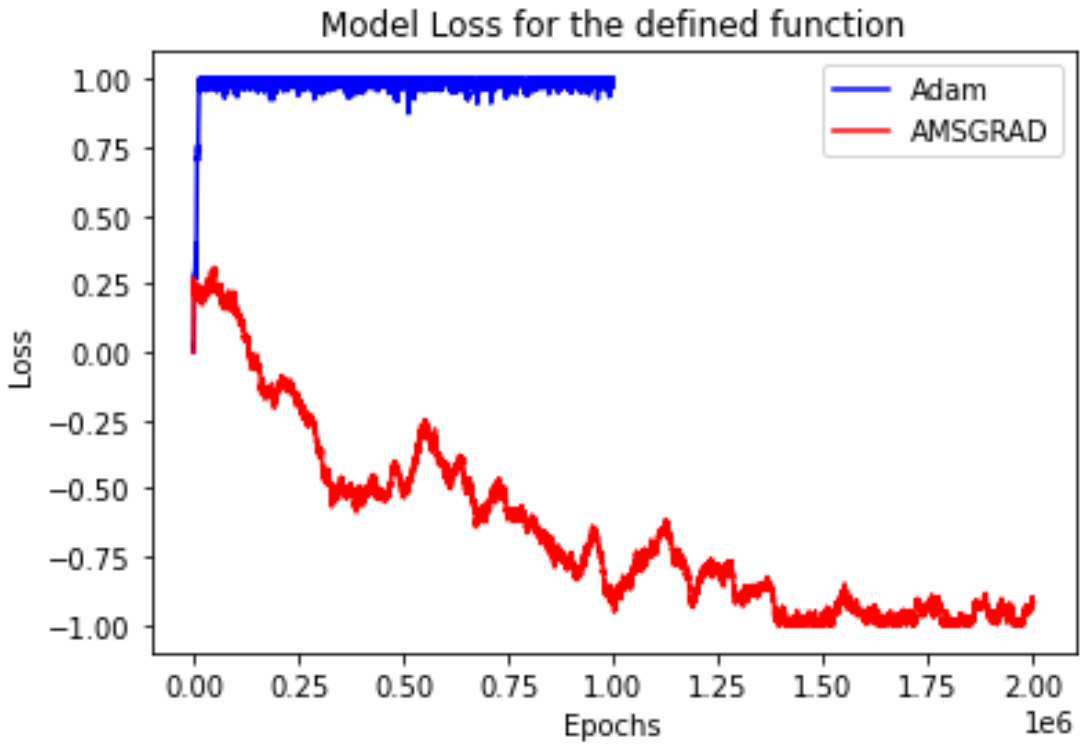
Thực nghiệm Adam với AMSGrad

1. **Synthetic**
2. **Logistic Regression**
3. **Neural Network**
4. **VGG**

Thực nghiệm Adam với AMSGrad

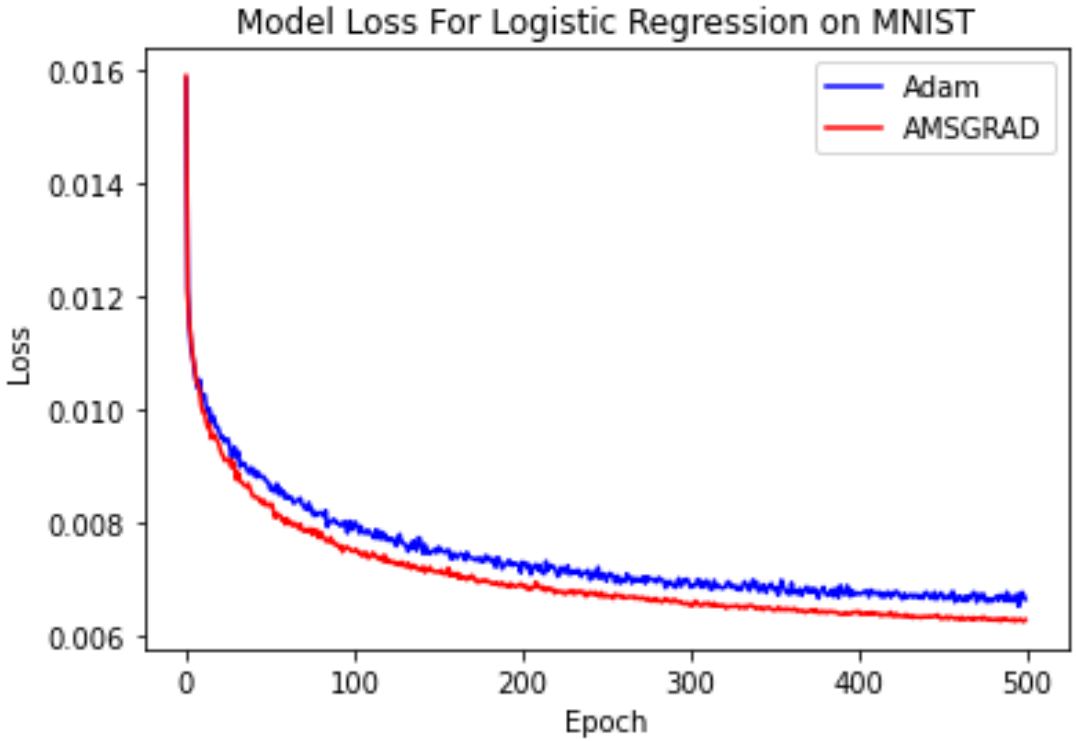
$$f_t(x) = \begin{cases} 1010x & \text{for } t \bmod 101 = 1 \\ -10x & \text{otherwise} \end{cases}$$

Với $\mathcal{F} = [-1; 1]$, có thể thấy giải pháp tối ưu là x = -1 vì tại điểm này sẽ cho ra regret nhỏ nhất. Do đó, x = -1 là điểm hội tụ tối ưu nhất.



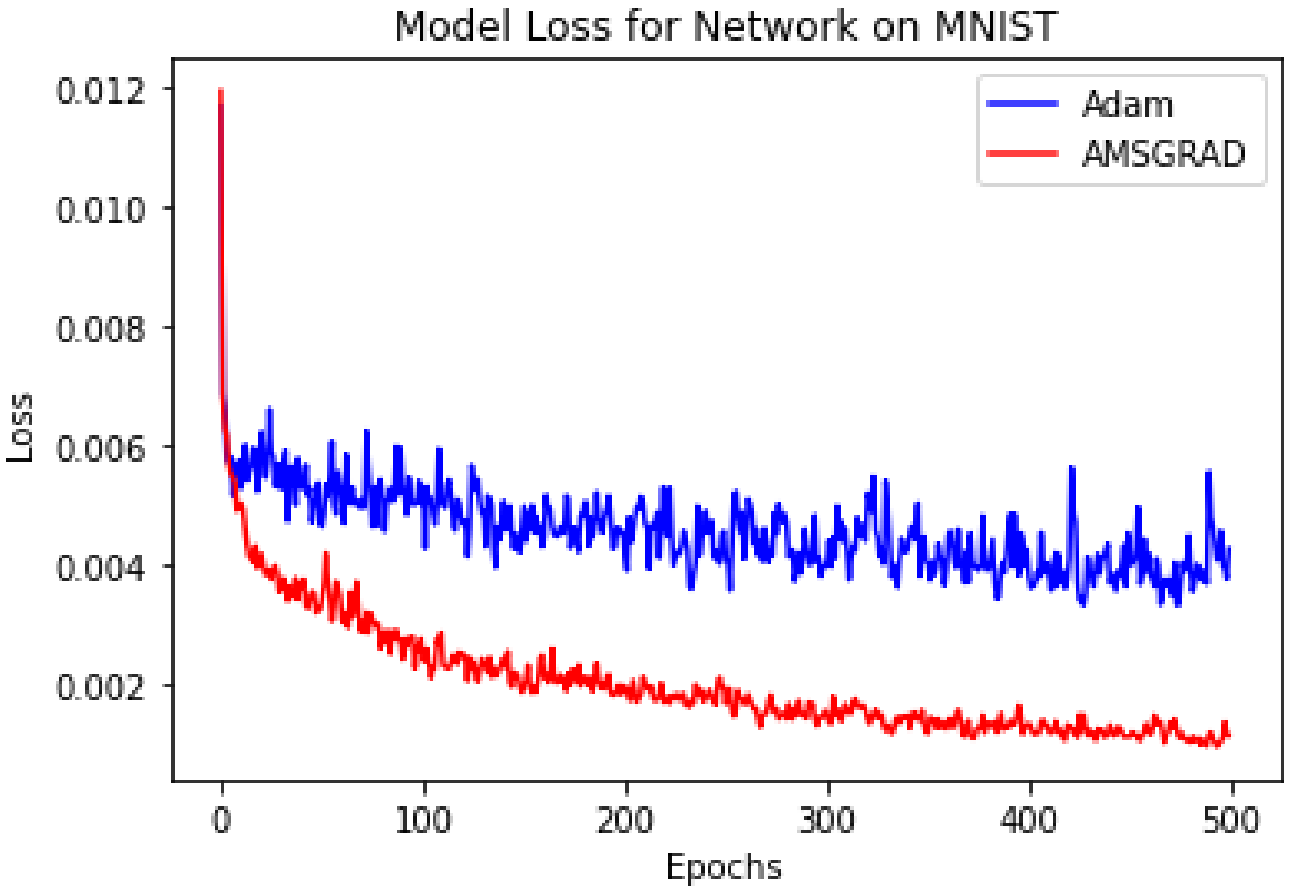
Thực nghiệm Adam với AMSGrad

- Phân loại bộ dữ liệu MNIST bằng Logistic Regression.
- Hyper-parameters: $\beta_1 = 0.9$, $\beta_2 = 0.99$,
$\alpha = 0.01$, batch size $= 128$, epochs $= 500$.



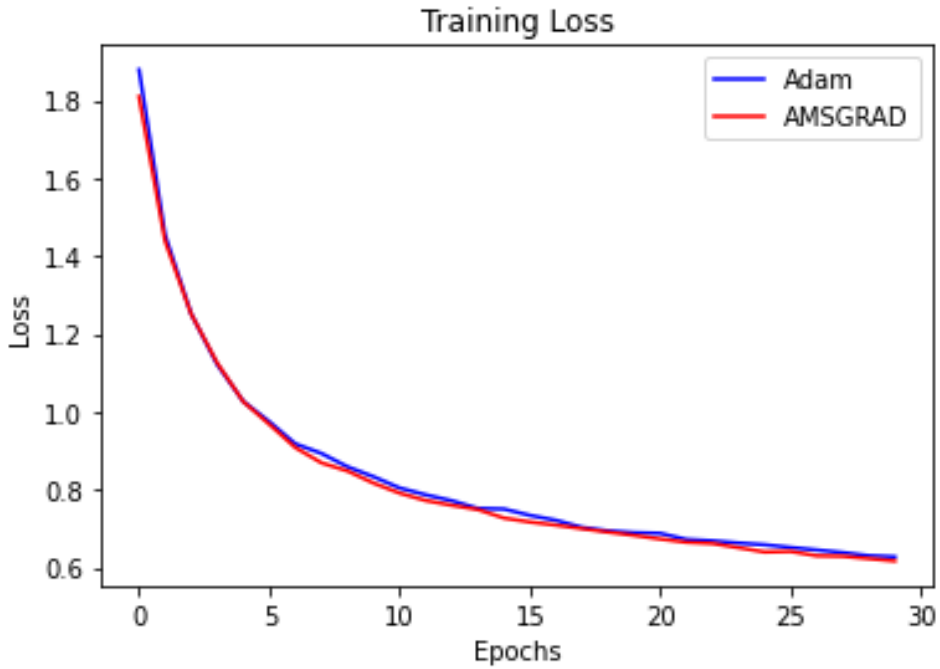
Thực nghiệm Adam với AMSGrad

- Tương tự thí nghiệm trước, nhưng thêm một hidden layer với 100 neuron.

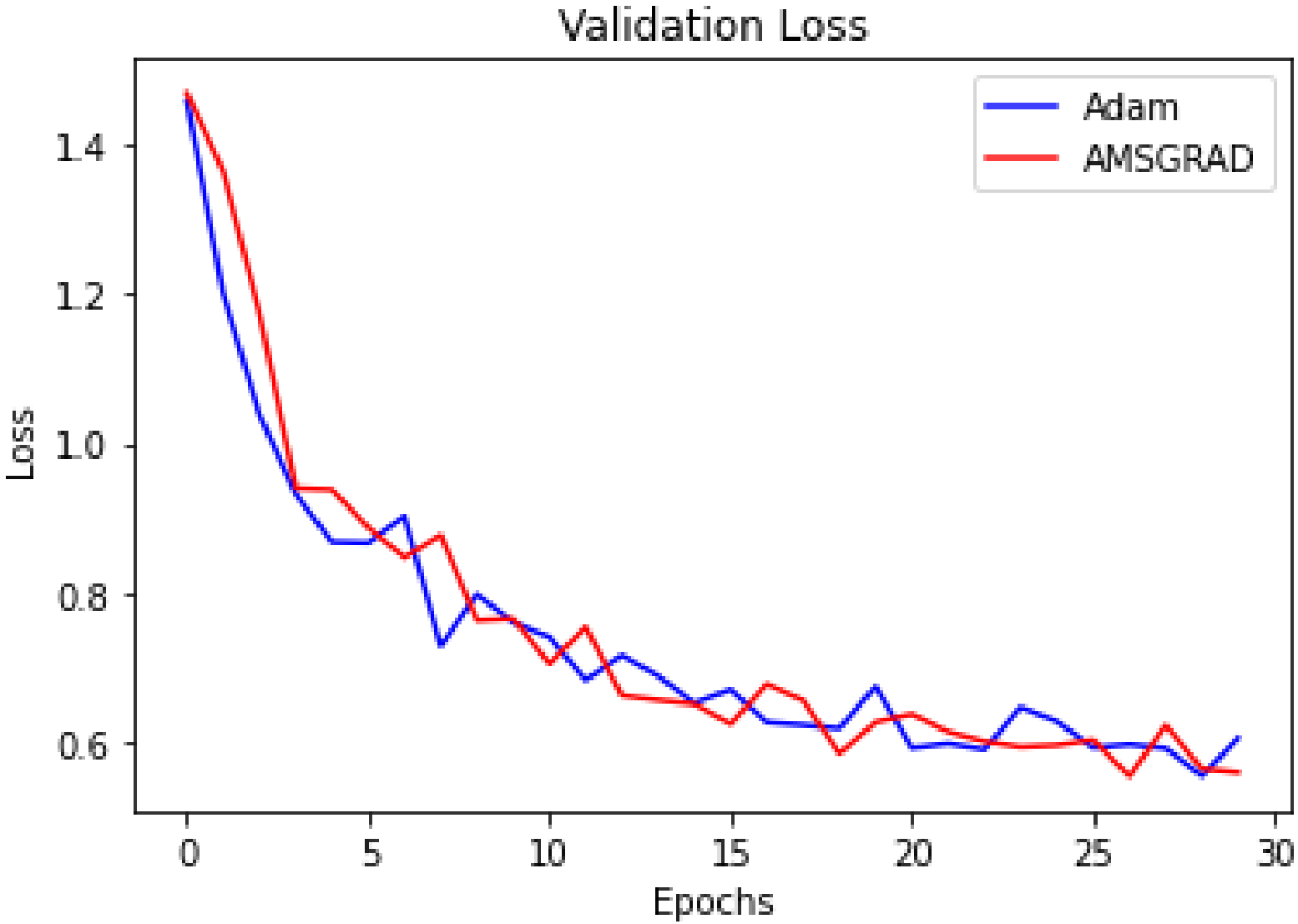


Thực nghiệm Adam với AMSGrad

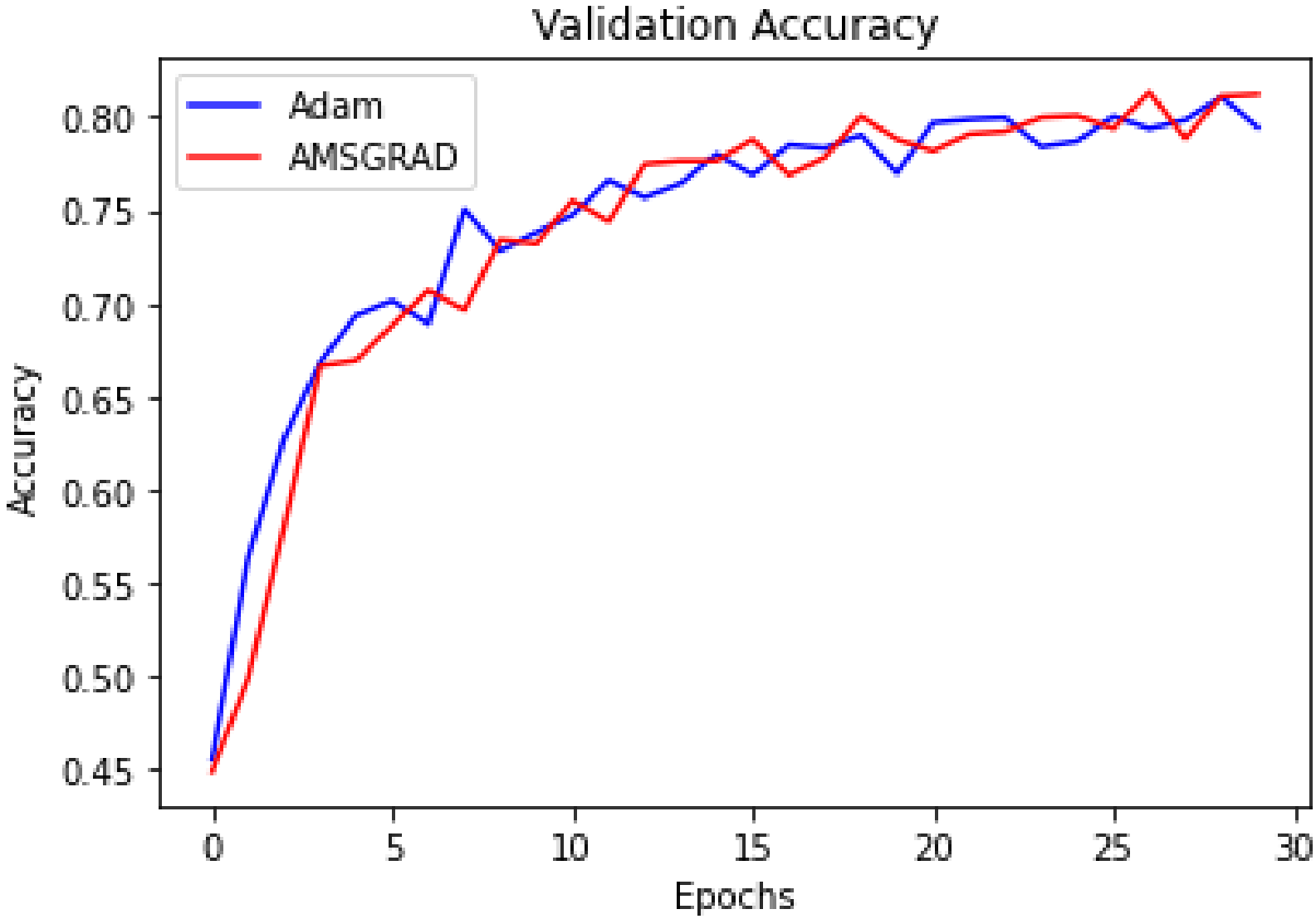
- Sử dụng mạng VGG cho bộ dữ liệu Cifar10 với Hyper parameters : $\beta_1 = 0.9,\ \beta_2 = 0.99,\ \alpha = 0.001,$ batch size $= 128,$ epochs $= 30$



Thực nghiệm Adam với AMSGrad



Thực nghiệm Adam với AMSGrad



Thank you!

Cảm ơn mọi người và thầy đã lắng nghe!