Arquitecturas de Deep Learning

Sebastian Arpon, PhD(c)
Físico y Data Scientist
MetricArts

github.com/sarpon

Temario

- Algoritmos de optimización
  - SGD
  - SGD+Momentum
  - RMSProp
  - ADAM


BRACE YOURSELF
OPTIMIZATION IS COMING

SGD

STOCHASTIC GRADIENT DESCENT

SGD

- Algoritmo de optimización basado en Gradient Descent, pero en vez de usar toda la data simultáneamente, usa pequeños grupos (batch) para realizar cada iteración.

Algorithm 8.1 Stochastic gradient descent (SGD) update at training iteration k

Require: Learning rate ϵ_k
Require: Initial parameter $\boldsymbol{\theta}$
  while stopping criterion not met **do**
    Sample a minibatch of m examples from the training set $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ with corresponding targets $\boldsymbol{y}^{(i)}$.
    Compute gradient estimate: $\hat{\boldsymbol{g}} \leftarrow +\frac{1}{m}\nabla_{\boldsymbol{\theta}} \sum_i L(f(\boldsymbol{x}^{(i)};\boldsymbol{\theta}), \boldsymbol{y}^{(i)})$.
    Apply update: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \epsilon\hat{\boldsymbol{g}}$.
  end while

SGD

- El parámetro ϵ es conocido como taza de aprendizaje (Learning Rate) y se necesitan las siguientes condiciones para que el algoritmo termine.

$$\sum_{k=1}^{\infty} \epsilon_k = \infty,$$

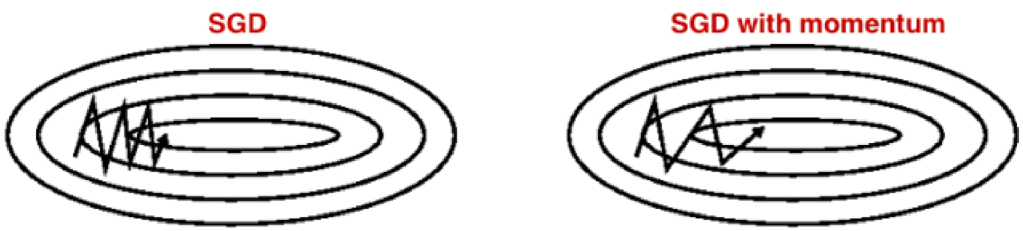
$$\sum_{k=1}^{\infty} \epsilon_k^2 < \infty.$$

Batch

- El batch es usado principalmente por la incapacidad computacional de calcular sobre todos nuestros datos.
- Si bien el uso de batch ayuda a alivianar la carga computacional tiene sus complejidades. Por ejemplo, la dirección del gradiente puede cambiar mucho iteración a iteración lo cual puede hacer que el algoritmo tome mas tiempo del necesario en converger.

Idea para mejorar convergencia

- Que pasa si en vez de solo utilizar el gradiente que calculamos actualmente, utilizamos los gradientes de las iteraciones anteriores para calcular la dirección de descenso. De acá nace la idea del momentum.



Taken from the Coursera Course Introduction to Deep Learning (by Higher School of Economics)

SGD+MOMENTUM

SGD+MOMENTUM

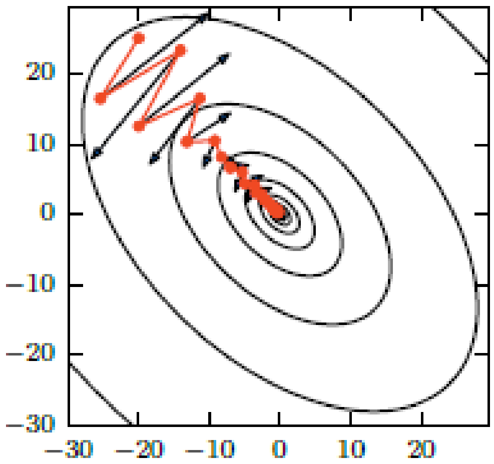
- En este algoritmo la actualización de la dirección de decrecimiento depende de las direcciones de descenso que se calcularon en iteraciones previas.

Algorithm 8.2 Stochastic gradient descent (SGD) with momentum

Require: Learning rate ϵ, momentum parameter α
Require: Initial parameter $\boldsymbol{\theta}$, initial velocity $\boldsymbol{v}$
while stopping criterion not met **do**
Sample a minibatch of m examples from the training set $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ with corresponding targets $\boldsymbol{y}^{(i)}$.
Compute gradient estimate: $\boldsymbol{g} \leftarrow \frac{1}{m} \nabla_{\boldsymbol{\theta}} \sum_i L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}), \boldsymbol{y}^{(i)})$.
Compute velocity update: $\boldsymbol{v} \leftarrow \alpha \boldsymbol{v} - \epsilon \boldsymbol{g}$.
Apply update: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \boldsymbol{v}$.
end while

SGD+MOMENTUM

- En la medida el parámetro de aprendizaje (ϵ) sea mayor que el parámetro de momemtum (α) significa que el gradiente obtenido actualmente importa menos que los obtenidos en iteraciones anteriores.
- A tipicamente toma valores de 0.5, 0.9 o 0,99.



RMSProp

ROOT MEAN SQUARE PROPAGATION

RMSProp

- A diferencia de los algoritmos anteriores RMSProp cambia la taza de aprendizaje de cada dirección del gradiente de forma independiente.

Algorithm 8.5 The RMSProp algorithm

Require: Global learning rate ϵ, decay rate ρ
Require: Initial parameter $\boldsymbol{\theta}$
Require: Small constant δ, usually 10^{-6}, used to stabilize division by small numbers
  Initialize accumulation variables $\boldsymbol{r} = 0$
  while stopping criterion not met **do**
    Sample a minibatch of m examples from the training set $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ with corresponding targets $\boldsymbol{y}^{(i)}$.
    Compute gradient: $\boldsymbol{g} \leftarrow \frac{1}{m} \nabla_{\boldsymbol{\theta}} \sum_i L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}), \boldsymbol{y}^{(i)})$.
    Accumulate squared gradient: $\boldsymbol{r} \leftarrow \rho \boldsymbol{r} + (1 - \rho) \boldsymbol{g} \odot \boldsymbol{g}$.
    Compute parameter update: $\Delta \boldsymbol{\theta} = -\frac{\epsilon}{\sqrt{\delta + \boldsymbol{r}}} \odot \boldsymbol{g}$. ($\frac{1}{\sqrt{\delta + \boldsymbol{r}}}$ applied element-wise)
    Apply update: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \Delta \boldsymbol{\theta}$.
  end while

Acerca de los cambios en las tazas

- Se puede ver que en la medida que el cuadrado del gradiente sea pequeño aumentaremos la taza de aprendizaje en esa dirección y viceversa.

ADAM

ADAPTIVE MOMENT ESTIMATION

ADAM

- Este algoritmo es como un RMSProp con esteroides.
- Sigue el mismo principio que el anterior, pero el concepto de re escalamiento y hasta cierto punto el de momentum.

Algorithm 8.7 The Adam algorithm

Require: Step size ϵ (Suggested default: 0.001)
Require: Exponential decay rates for moment estimates, ρ_1 and ρ_2 in $[0, 1)$. (Suggested defaults: 0.9 and 0.999 respectively)
Require: Small constant δ used for numerical stabilization (Suggested default: 10^{-8})
Require: Initial parameters $\boldsymbol{\theta}$
  Initialize 1st and 2nd moment variables $\boldsymbol{s} = \mathbf{0}$, $\boldsymbol{r} = \mathbf{0}$
  Initialize time step $t = 0$
  while stopping criterion not met **do**
    Sample a minibatch of m examples from the training set $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ with corresponding targets $\boldsymbol{y}^{(i)}$.
    Compute gradient: $\boldsymbol{g} \leftarrow \frac{1}{m} \nabla_{\boldsymbol{\theta}} \sum_i L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}), \boldsymbol{y}^{(i)})$
    $t \leftarrow t + 1$
    Update biased first moment estimate: $\boldsymbol{s} \leftarrow \rho_1 \boldsymbol{s} + (1 - \rho_1) \boldsymbol{g}$
    Update biased second moment estimate: $\boldsymbol{r} \leftarrow \rho_2 \boldsymbol{r} + (1 - \rho_2) \boldsymbol{g} \odot \boldsymbol{g}$
    Correct bias in first moment: $\hat{\boldsymbol{s}} \leftarrow \frac{\boldsymbol{s}}{1 - \rho_1^t}$
    Correct bias in second moment: $\hat{\boldsymbol{r}} \leftarrow \frac{\boldsymbol{r}}{1 - \rho_2^t}$
    Compute update: $\Delta \boldsymbol{\theta} = -\epsilon \frac{\hat{\boldsymbol{s}}}{\sqrt{\hat{\boldsymbol{r}}} + \delta}$ (operations applied element-wise)
    Apply update: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \Delta \boldsymbol{\theta}$
  end while

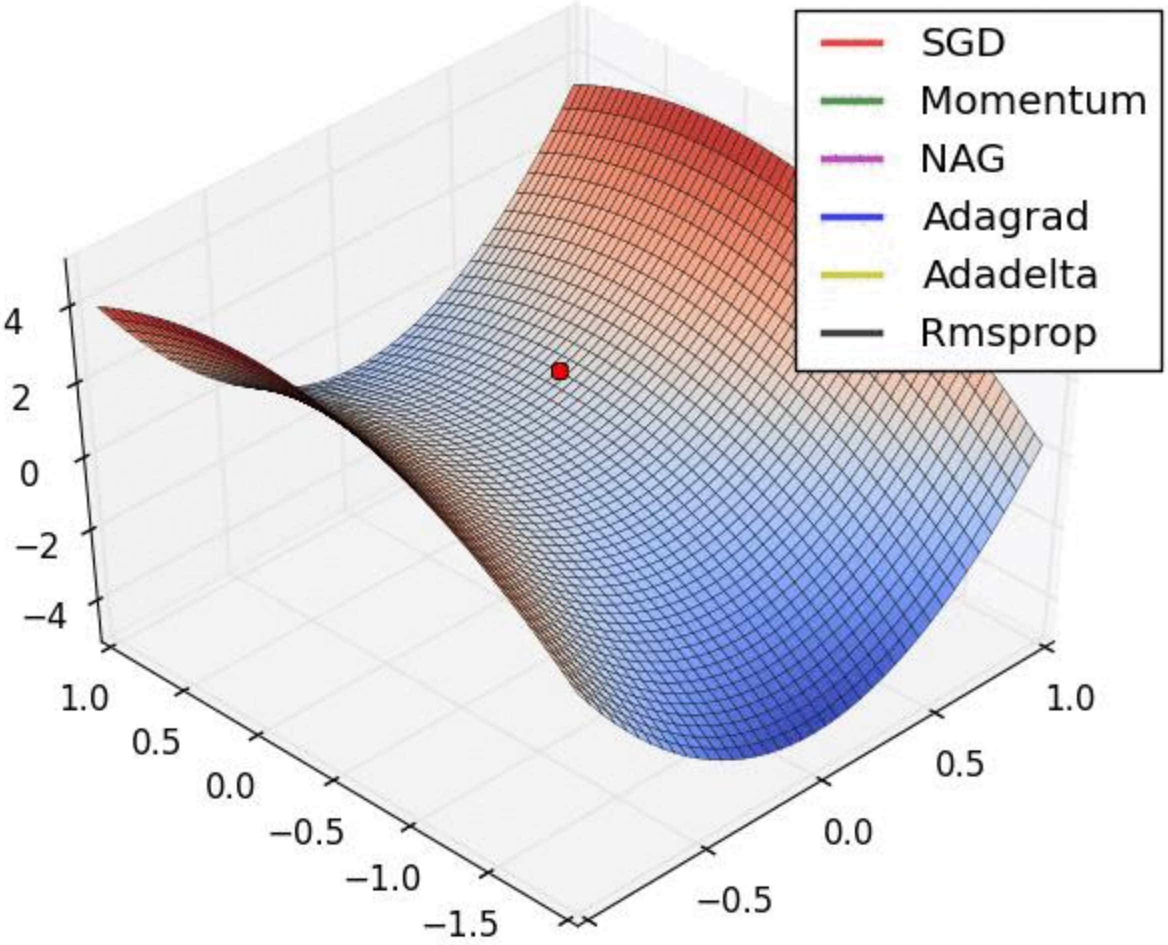
Acerca de ADAM

- El re escalamiento que hace a los pesos permite que el decrecimiento de el learning rate no sea tan agresivo.

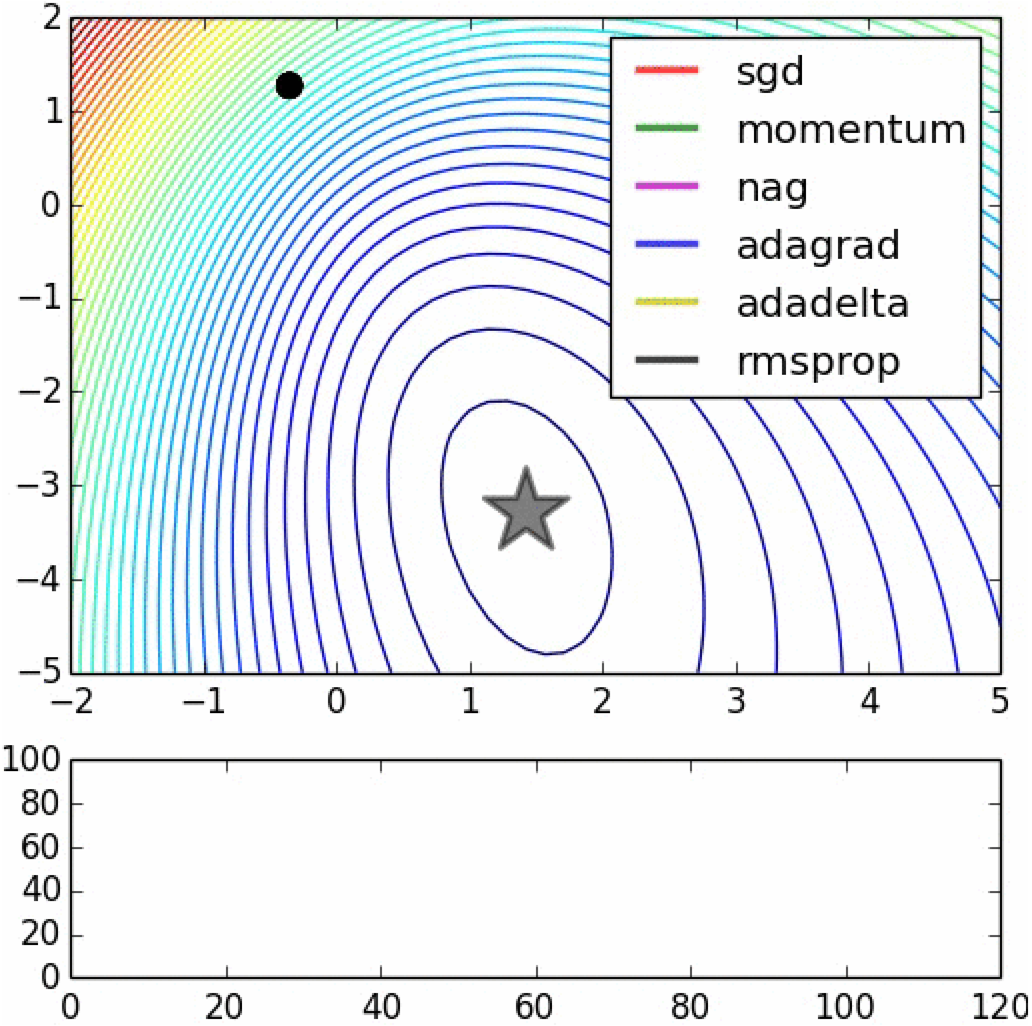
gifs

EL MOMENTO DE LOS MEMES

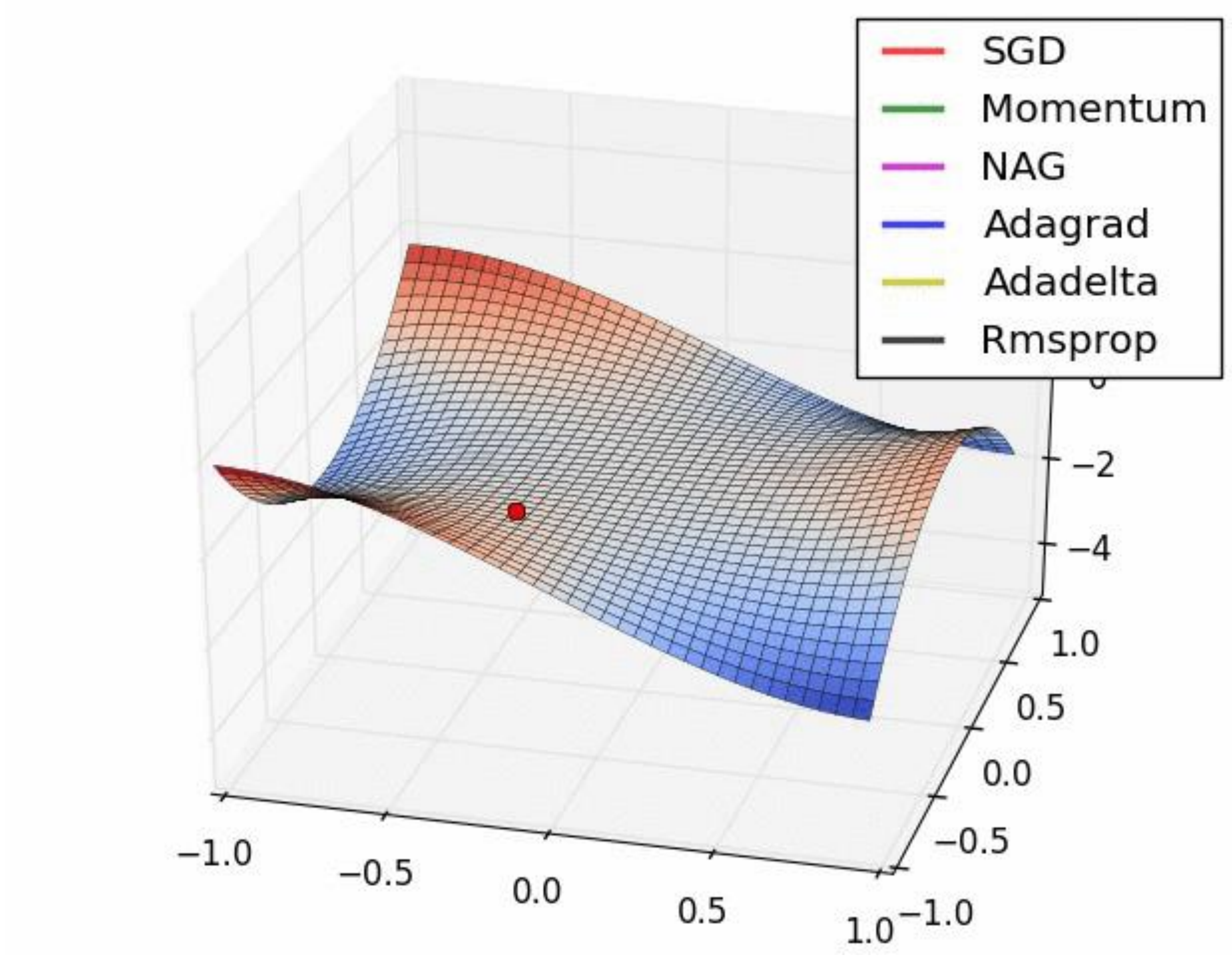
Comparemos algoritmos



Comparemos algoritmos



Comparemos algoritmos



Comparemos algoritmos