

# Redes Neurais e Aprendizagem Profunda

## ATUALIZAÇÃO DE PESOS RMSPROP / ADAGRAD

---

Zenilton K. G. Patrocínio Jr

[zenilton@pucminas.br](mailto:zenilton@pucminas.br)

# RMSProp

[Hinton et al., 2012]

Os gradientes podem variar muito, embora os parâmetros geralmente tenham a mesma escala

# RMSProp

[Hinton et al., 2012]

Os gradientes podem variar muito, embora os parâmetros geralmente tenham a mesma escala

RMSProp (*Root Mean Squared Propagation*) ajusta os valores de gradientes pelo inverso de uma média móvel

# RMSProp

[Hinton et al., 2012]

Os gradientes podem variar muito, embora os parâmetros geralmente tenham a mesma escala

RMSProp (*Root Mean Squared Propagation*) ajusta os valores de gradientes pelo inverso de uma média móvel

Define-se

$$s^{(t+1)} = \beta s^{(t)} + (1 - \beta)(g^{(t)})^2$$

em que:

- $s^{(t)}$  é o gradiente quadrático médio (media móvel) no passo  $t$ ,

# RMSProp

[Hinton et al., 2012]

Os gradientes podem variar muito, embora os parâmetros geralmente tenham a mesma escala

RMSProp (*Root Mean Squared Propagation*) ajusta os valores de gradientes pelo inverso de uma média móvel

Define-se

$$s^{(t+1)} = \beta s^{(t)} + (1 - \beta)(g^{(t)})^2$$

em que:

- $s^{(t)}$  é o gradiente quadrático médio (media móvel) no passo  $t$ ,
- $\beta \in [0,1]$  é o fator de decaimento da media móvel,

# RMSProp

[Hinton et al., 2012]

Os gradientes podem variar muito, embora os parâmetros geralmente tenham a mesma escala

RMSProp (*Root Mean Squared Propagation*) ajusta os valores de gradientes pelo inverso de uma média móvel

Define-se

$$s^{(t+1)} = \beta s^{(t)} + (1 - \beta)(g^{(t)})^2$$

quadrado elemento a elemento de  $g^{(t)}$ ,

assim  $s^{(t)}$  possui as mesmas dimensões que  $g^{(t)}$

em que:

- $s^{(t)}$  é o gradiente quadrático médio (media móvel) no passo  $t$ ,
- $\beta \in [0,1]$  é o fator de decaimento da media móvel,
- $g^{(t)}$  é o gradiente do *minibatch* no passo  $t$

# RMSProp

[Hinton et al., 2012]

Os gradientes podem variar muito, embora os parâmetros geralmente tenham a mesma escala

RMSProp (*Root Mean Squared Propagation*) ajusta os valores de gradientes pelo inverso de uma média móvel

Define-se

$$s^{(t+1)} = \beta s^{(t)} + (1 - \beta)(g^{(t)})^2$$

quadrado elemento a elemento de  $g^{(t)}$ ,  
assim  $s^{(t)}$  possui as mesmas dimensões que  $g^{(t)}$

em que:

- $s^{(t)}$  é o gradiente quadrático médio (media móvel) no passo  $t$ ,
- $\beta \in [0,1]$  é o fator de decaimento da media móvel,
- $g^{(t)}$  é o gradiente do *minibatch* no passo  $t$

A regra de atualização do **RMSProp** é dada por:

$$W^{(t+1)} = W^{(t)} - \alpha \frac{g^{(t)}}{\sqrt{s^{(t)}}}$$

→ divisão elemento a elemento!

# ADAGRAD

[Duchi et al., 2011]

ADAGRAD (*ADAptive GRADient*) é semelhante ao RMSProp, porém utiliza a **soma acumulada** dos quadrados dos gradientes

# ADAGRAD

[Duchi et al., 2011]

ADAGRAD (*ADAptive GRADient*) é semelhante ao RMSProp, porém utiliza a **soma acumulada** dos quadrados dos gradientes

Define-se

$$c^{(t)} = \sum_{j=1}^t (g^{(j)})^2$$

quadrado elemento a element de  $g^{(t)}$ ,  
assim  $c^{(t)}$  possui as mesmas dimensões que  $g^{(t)}$

# ADAGRAD

[Duchi et al., 2011]

ADAGRAD (*ADAptive GRADient*) é semelhante ao RMSProp, porém utiliza a **soma acumulada** dos quadrados dos gradientes

Define-se

$$c^{(t)} = \sum_{j=1}^t (g^{(j)})^2$$

quadrado elemento a element de  $g^{(t)}$ ,  
assim  $c^{(t)}$  possui as mesmas dimensões que  $g^{(t)}$

Assim,  $c^{(t)}$  representa o gradiente quadrático **acumulado** no passo  $t$

# ADAGRAD

[Duchi et al., 2011]

ADAGRAD (*ADAptive GRADient*) é semelhante ao RMSProp, porém utiliza a **soma acumulada** dos quadrados dos gradientes

Define-se

$$c^{(t)} = \sum_{j=1}^t (g^{(j)})^2$$

quadrado elemento a elemento de  $g^{(t)}$ ,  
assim  $c^{(t)}$  possui as mesmas dimensões que  $g^{(t)}$

Assim,  $c^{(t)}$  representa o gradiente quadrático **acumulado** no passo  $t$

A regra de atualização do **ADAGRAD** é dada por:

$$W^{(t+1)} = W^{(t)} - \alpha \frac{g^{(t)}}{\sqrt{c^{(t)}}}$$

→ divisão elemento a elemento!

OBS:  $c^{(t)}$  tende a crescer linearmente em função de  $t$ , assim o ADAGRAD reduz sua taxa de aprendizado efetiva ao longo do tempo em  $1/\sqrt{t}$

# RMSprop × ADAGRAD

```
# RMSProp
cache = decay_rate * cache + (1 - decay_rate) * dx**2
x += - learning_rate * dx / (np.sqrt(cache) + 1e-7)
```

```
# Adagrad update
cache += dx**2
x += - learning_rate * dx / (np.sqrt(cache) + 1e-7)
```

# RMSprop x ADAGRAD

```
# RMSProp
cache = decay_rate * cache + (1 - decay_rate) * dx**2
x += - learning_rate * dx / (np.sqrt(cache) + 1e-7)
```

Diferença

```
# Adagrad update
cache += dx**2
x += - learning_rate * dx / (np.sqrt(cache) + 1e-7)
```

# RMSProp x ADAGRAD

- Como ambos normalizam magnitudes de gradiente, o RMSProp e o ADAGRAD funcionam muito bem em conjuntos de dados com grande variação na magnitude de gradientes

# RMSProp x ADAGRAD

- Como ambos normalizam magnitudes de gradiente, o RMSProp e o ADAGRAD funcionam muito bem em conjuntos de dados com grande variação na magnitude de gradientes
- O exemplo mais comum de aplicação são dados textuais, nos quais os gradientes variam de 4 a 5 ordens de magnitude

# RMSProp x ADAGRAD

- Como ambos normalizam magnitudes de gradiente, o RMSProp e o ADAGRAD funcionam muito bem em conjuntos de dados com grande variação na magnitude de gradientes
- O exemplo mais comum de aplicação são dados textuais, nos quais os gradientes variam de 4 a 5 ordens de magnitude
- O uso do RMSProp / ADAGRAD pode acelerar o aprendizado de modelos de texto em 2-3 ordens de magnitude

# RMSProp x ADAGRAD

- Como ambos normalizam magnitudes de gradiente, o RMSProp e o ADAGRAD funcionam muito bem em conjuntos de dados com grande variação na magnitude de gradientes
- O exemplo mais comum de aplicação são dados textuais, nos quais os gradientes variam de 4 a 5 ordens de magnitude
- O uso do RMSProp / ADAGRAD pode acelerar o aprendizado de modelos de texto em 2-3 ordens de magnitude
- Porém, menos eficaz quando existem fortes dependências entre as características

# RMSProp × ADAGRAD

- O RMSProp é uma abordagem heurística, enquanto o ADAGRAD possui limites formais na sua taxa de convergência, embora apenas para problemas convexos

# RMSProp × ADAGRAD

- O RMSProp é uma abordagem heurística, enquanto o ADAGRAD possui limites formais na sua taxa de convergência, embora apenas para problemas convexos
- A taxa de aprendizado no RMSProp é fixa ao longo do tempo, sendo mais adequado para tarefas de treinamento de longa duração

# RMSProp × ADAGRAD

- O RMSProp é uma abordagem heurística, enquanto o ADAGRAD possui limites formais na sua taxa de convergência, embora apenas para problemas convexos
- A taxa de aprendizado no RMSProp é fixa ao longo do tempo, sendo mais adequado para tarefas de treinamento de longa duração
- A magnitude da soma dos gradientes quadráticos do ADAGRAD cresce linearmente com o tempo  $t$ ; portanto, a taxa de aprendizado no ADAGRAD reduz em  $1/\sqrt{t}$ , o que é bastante agressivo (podendo ser bom para modelos curtos e fáceis de treinar, porém rápido demais para treinos de longa duração)

# RMSProp × ADAGRAD

