

ADAM optimalizációs eljárás

Kormányos Andor

Komplex Rendszerek Fizikája Tanszék

2022. november 2.

Legmeredekebb ereszkedés gépi tanulásban¹

Emlékeztető:

- iteratív algoritmus
- válasszunk egy tetszőleges kezdőértéket az \mathbf{a} paramétervektor elemeinek
- repeat until convergence {
 $a_j := a_j - \alpha \frac{\partial}{\partial a_j} J(\mathbf{a}; \mathbf{x}^{(i)}, y^{(i)})$
}

Alapesetben

- az α learning rate állandó
- minden a_j esetén ugyanazt az α -t használjuk

¹Andrew Ng, Coursera alapján

Legmeredekebb ereszkedés gépi tanulásban¹

Emlékeztető:

- iteratív algoritmus
- válasszunk egy tetszőleges kezdőértéket az \mathbf{a} paramétervektor elemeinek
- repeat until convergence {
 $a_j := a_j - \alpha \frac{\partial}{\partial a_j} J(\mathbf{a}; \mathbf{x}^{(i)}, y^{(i)})$
}

Alapesetben

- az α learning rate állandó
- minden a_j esetén ugyanazt az α -t használjuk

Tovább lépés:

- α változtatása ahogy minimumkeresés előrehalad (pl. kezdetben nagyobb, majd ahogy közeledünk a minimumhoz egyre kisebb)
- különböző a_j esetén különböző α_j használata

¹Andrew Ng, Coursera alapján

A gradient descent módszer kiterjesztése: **Adaptive Movement Estimation Algorithm: ADAM**

- a gradiens vektor átlagával kapcsolatos mennyiséget
- gradiens vektor szórásával kapcsolatos mennyiség átlagát számoljuk

ADAM módszer

A gradient descent módszer kiterjesztése: **Adaptive Movement Estimation Algorithm: ADAM**

- a gradiens vektor átlagával kapcsolatos mennyiséget
- gradiens vektor szórásával kapcsolatos mennyiség átlagát számoljuk

Bevezetünk három új “hyperparamétert”

- $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$
- ezeket az értékek problémák széles körére működnek

Emellett persze inicializálni kell az α értékét is, pl $\alpha = 0.01$

Előkészület:

- \mathbf{a} paramétervektor elemeit egy $\mathbf{a}[j]$ vektorban tárolhatjuk
- hasonlóan, a $\frac{\partial}{\partial a_j} J(\mathbf{a})$ értékét egy $\text{derivJ}[j]$ vektor elemeiként
- $a_p[j]$ és $\text{derivJ}_p[j]$ jelöli ezek értékét a p -ik iterációs lépésben

ADAM módszer

Előkészület:

- \mathbf{a} paramétervektor elemeit egy $\mathbf{a}[j]$ vektorban tárolhatjuk
- hasonlóan, a $\frac{\partial}{\partial a_j} J(\mathbf{a})$ értékét egy $\text{derivJ}[j]$ vektor elemeiként
- $\mathbf{a}_p[j]$ és $\text{derivJ}_p[j]$ jelöli ezek értékét a p -ik iterációs lépésben

Bevezetünk két új vektort is inicializáljuk őket 0 értékkel:

- $\mathbf{m}_{p=0}[j] = 0$ és $\mathbf{v}_{p=0}[j] = 0$
- \mathbf{m} és \mathbf{v} lényegében az *exponenciális mozgóátlagot* és az *exponenciális mozgó szórást* tartalmazzák
- az iteráció során a korábbi értékek egyre kevésbé számítanak a pillanatnyi átlag/szórás számításakor

Előkészület:

- \mathbf{a} paramétervektor elemeit egy $\mathbf{a}[j]$ vektorban tárolhatjuk
- hasonlóan, a $\frac{\partial}{\partial \mathbf{a}_j} J(\mathbf{a})$ értékét egy $\text{derivJ}[j]$ vektor elemeiként
- $\mathbf{a}_p[j]$ és $\text{derivJ}_p[j]$ jelöli ezek értékét a p -ik iterációs lépésben

Bevezetünk két új vektort is inicializáljuk őket 0 értékkel:

- $\mathbf{m}_{p=0}[j] = 0$ és $\mathbf{v}_{p=0}[j] = 0$
- \mathbf{m} és \mathbf{v} lényegében az *exponenciális mozgóátlagot* és az *exponenciális mozgó szórást* tartalmazzák
- az iteráció során a korábbi értékek egyre kevésbé számítanak a pillanatnyi átlag/szórás számításakor

Szükség lesz még két segédváltozó vektorra: $\hat{\mathbf{m}}[j]$, $\hat{\mathbf{v}}[j]$

```
repeat until convergence {  
  calculate derivJp[j]  
  mp[j] = β1mp-1[j] + (1.0 - β1)derivJp[j]  
  vp[j] = β2vp-1[j] + (1.0 - β2)(derivJp[j])2  
   $\hat{m}[j] = \frac{m_p[j]}{1.0 - (\beta_1)^p}$   
   $\hat{v}[j] = \frac{v_p[j]}{1.0 - (\beta_2)^p}$   
  ap[j] := ap-1[j] - α  $\frac{\hat{m}[j]}{\sqrt{\hat{v}[j] + \epsilon}}$   
}
```

Látható, hogy a paraméterterben az effektív lépéshossz nagysága a p -ik lépésben $\Delta_p[j] = \alpha \frac{\hat{m}[j]}{\sqrt{\hat{v}[j] + \epsilon}}$

- megmutatható, hogy a legtöbb esetben $|\Delta_p[j]| \leq \alpha$
- a $\frac{\hat{m}[j]}{\sqrt{\hat{v}[j]}}$ felfogható egyfajta “jel-zaj aránynak” (signal-to-noise ratio)
- pl ha derivált előjele gyakran változik, akkor $\frac{\hat{m}[j]}{\sqrt{\hat{v}[j]}} < 1$
- ha a jel-zaj arány kicsi, akkor kisebb lépésekkel megy előre a keresés

Eredeti kézirat: <https://arxiv.org/abs/1412.6980>