

Week 10 Quiz 1回目 1

1] stochastic gradient descent を用いて logistic regression classifier を training している。直近の 500 example (本) の平均の cost ($\theta, (x^{(i)}, y^{(i)})$) をくり返し回数を横軸にプロットすると、中々りと増加していくと判明した。

① 問題ではない。想定されたことだ

X それはダメだよ

② 平均する example の数を増やせ

X 平均するだけでは平均が減少傾向は直らないので

③ α を小さく

O 正しい

④ α を大きく

X

2] stochastic ~ について

① 利点の w は 並列化できることで、batch ~ よりもずっと速くなる

X

② 巨大な training set の時、batch ~ よりもずっと速くなる

O 一部の training set を使うから、当然速くなる

③ 前にデータを 32×4 にシャッフルする

O

④ 収束を確実にするために、くり返しの度に $J_{train}(\theta)$ を計算するが普通は、 J が減少し続けている

X 1000回くり返して1回 J を計算する程度でよい

X [3] online learning に あてはまるものは?

- ① ビデオでは、くり返し1つの training example ^{↑ 1つずつのこと} を得て、stochastic ~ を実行して、次の example の処理に移る。
* 1個ずつ example がくるので stochastic ~ を使う意味がある。
○ ^{か正解} stochastic ~ は 非常に多量の training set がある場合に1回を使うだけ。
- ② 各ステップで 新しい example (x, y) を得て learning をその example に対して行う (本質的に stochastic ~ と同じ)、その example は捨て、次へ。
○
- ③ 欠点は、training example を保存するのに memory と disk space が ~~必要~~。
X データは stream として得られる
- ④ 利点の1つは、学習率 α を選ぶ必要がないこと。
X

[4] 非常に大量の training set。map-reduce で並列化できるアルゴリズムは?

- X ① stochastic ~ を用いた logistic regression
- ② batch ~ =
- X ③ stochastic ~ Linear regression
- ④ training set 中の feature の平均を求めた

X 5 map-reduce にあてはまる記述は?

① gradient descent で map-reduce を使う時、通常各 map-reduce 2 台からの gradient を加算するのは 1 台のマシンをつかう。

(各マシンのパラメータを更新するため)

X 割り当てられたデータを加算するのは各 map-reduce 2 台。
部分和を集めて加算するのは 1 台だけ。

② ネットワークによる遅延と、他のオーバヘッドのため、 N 台で map-reduce を動作させても N -fold (N 重) 以下の speed up とする

O

③ N 台で map-reduce すると、少なくとも N 重の speed up が実現される

X

④ 1 core の 1 台のコンピュータを使っていると、map-reduce は役に立たない。

O overhead が発生する。

Week 10 Quiz 2回目 4

1 1回目と同じ

- X ① もっと少ない example で平均を計算してアップデートする
- X ② トレーニングセットの example を減らす
- O ③ α を減らす。
- X ④ 収束が保証されているので、このようにすることは起きない。

X 2 stochastic ~ について

- X ① $J(\theta)$ は減ることか保証されている
- X ② 収束させるためには、 ϵ 区間の度に $J_{\text{train}}(\theta)$ を計算して、減少を確認する
- X ③ numerical gradient descent で bug check して、やはり stochastic ~ は偏微分がからむ。
- O ④ 前々、データをランダムにシャッフルする

3 online learning

- O ① 毎ステップ新しい example (x, y) を得て、その example を学習 (基本的に stochastic gradient descent を) し、その example に μ だけ。
↑ stochastic ~ は子集
- X ② α を選択する必要が在るのかもしれない
- X ③ 2点法 example data memory と disk space が必要
- O ④ ϵ 区間 1 個の training example を得て、その example を stochastic を使って、次の example に行く。

Week 10 Quiz 2回目 5

4 非常に大きな training set. map-reduceで並列化すべきなのは?

① Logistic regression with Batch gradient descent

② = stochastic -

③ Linear =

④ 全ての features の平均

5 map-reduce

① 1台のマシンのみ, 24 core CPU or 24F CPU.

② gradient descent で, 1台で gradient を計算する
「部分和を計算すること」を言っているように

③ N 台で走らせるためにはデータを N^2 に分割する

④ 並列化のためは, データに関数をかけた和の形にする。

[注] 1台のマシンの1コアでも map-reduce は役に立つ → ①が正解
データのアクセスを局所化する事でキャッシュが hit しやすい?

Week 10 Quiz 3回目 6

2 stochastic ~

① 各くりかえしでは 1個の training データしか使わない

2 stochastic ~

x ① $J(\theta)$ は ~~step~~ step ごとに減少するわけではない

x ② training set から小さい割合だけ選ぶ

x ③ 毎 2 行ごと、1つの training example を用いる

o 正しい

o ④ 利点は 1個の example を見ただけで θ を更新できる