Нейросетевой подход к поиску циркулянтных предобуславливателей для систем с теплицевыми матрицами

Осень 2024

Содержание

1 Постановка задачи

Имеется СЛАУ с теплицевой (пока симметричной матрицей):

$$\mathbf{Tx} = \mathbf{f},$$

Требуется найти *удачный* правый циркулянтный предобуславливатель $\mathbf{C}^{-1}$. Под удачным понимается предобуславливатель, ускоряющий сходимость итерационных методов, например, GMRES.

2 Подбор функционала

Для начала мы хотим определить функционал / функционалы, которые будем использоваться для обучения сети. Архитектуру пока отодвигаем в сторону, будем учить многослойный персептрон, который гарантированно «выучит все».

Существует несколько общих мотиваций, из которых можно строить функционалы:

- На скорость сходимости в итерационных методах влияет спектр матрицы $\mathbf{I} - \mathbf{TC}^{-1}$

  - Зажимаем его в ноль равномерно, оптимизуем нормы: спектральный радиус, вторая норма (∞-норма вектора сингулярных чисел), норма Фробениуса (2-норма вектора сингулярных чисел).
  - На самом деле мы не против небольшого количества выборосов среди собственных значений, то есть нас интересует (устраивает) малость спектра матрицы $\mathbf{I}-\mathbf{TC}^{-1}-\mathbf{R}$, где $\mathbf{R}$ – матрица малого ранга. В этом случае подходит ядерная норма (1-норма вектора сингулярных чисел), но она дорогая для вычисления. Звучит, как будто ее можно использовать как регуляризатор, но не на постоянной основе, а каждый батч / матрицу с вероятностью p штрафовать за нее.
  - Кроме того, пробуем оптимизироваться на следующий функционал: K-число обусловленности, которое определяется как

    $$\frac{(\mathrm{tr}A/n)^n}{\det A}$$

    Этот функционал возникает в теоретических выкладках и идейно штрафует собственные значений за общий разброс. Кроме того, например, если $A = \varepsilon I$, то

    $$\frac{(\mathrm{tr}A/n)^n}{\det A} = \frac{(n \cdot \varepsilon/n)^n}{\varepsilon^n} = 1$$

    То есть решение задачи оптимизации с таким функционалом не единственно и более того значение функционала совпадает на матрицах, имеющих очень различные свойства с точки зрения сходимости GMRES. Как вариант – использовать его с регуляризатором. Другим важным моментом является то, что этот функционал следует вычислять не напрямую, а через логарифмы, так как что в числителе, что в знаменателе стоят произведения многих чисел, но логарифм определителя «хорошо» бы работал если бы матрицы были положительно определенными, а это существенное, не совсем понятно как соблюдаемое ограничение.

- Можно учиться на быструю разрешимость конкретным методом, например, GMRES'ом. В этом подходе видятся несколько проблем:

– Что считать функцией потерь? Невязку на n-ом шаге? Число итераций до сходимости? Штрафовать за число итераций или за распределение числа итераций?

– Кажется, что учить «с нуля» такую сеть будет сложно, поскольку понятие «хороший спектр» для сходимости GMRES немного размыто, мы скорее можем указать конкретные случаи, когда мы знаем, что сходимость должна быть быстрой, но есть ощущение (возможно, с подтверждением из линала), что в начале обучения сеть будет много «путаться» и в итоге медленно учиться, возможно, можно выделить несколько «голов» сети и надеяться, что каждая выучит свою зависимость.

- Можно использовать другие хорошие варианты расположения собственных значений, например, известно, что GMRES хорошо сходится, если спектр матрицы «примерно зануляет» некоторый многочлен малой степени, отсюда если как-то кластеризовать собственные значения, то мы можем надеяться на быструю сходимость GMRES. Проблема в том, что считать сами собственные значения матрицы $\mathbf{TC}^{-1}$ дорого и нужно как-то из самой матрицы доставать информацию о кластерах собственных значений:

  – Идея: круги Гершгорина. По элементам матрицы за недорого можно достать круги на комплексной плоскости, содержащие собственные значения. Давайте, например, в каждом таком круге сэмплировать равномерно по 100 точек, кроме того, сэмплировать степень многочлена из распределения, например, Пуассона, решать задачу полиномиальной регрессии и штрафовать модель за ошибку решения. Задача регрессии решается аналитически, причем размер матрицы, которую нужно обращать – число признаков, а размерность пространства признаков – степень многочлена.

$$z_1, \ldots, z_N, \sum_i \|a_0 + z_i a_1 + \cdots + z_i^k a_k\|^2 \to \min_{|a_1|+\cdots+|a_k|\neq 0}$$

  Сформлированная задачка плохая с точки зрения оптимизации, поэтому давайте искать решение не просто каким-то многочленом, а со старшим коэффициентом, равным 1:

$$\sum_i \|a_0 + z_i a_1 + \cdots + z_i^k\|^2 \to \min$$

  Это просто линейная регрессия.

3 Преобразование Фурье?

Мы знаем, что существует связь между спектром теплицевой матрицы $\mathbf{T}$ и частичной суммой ряда Фурье. Отсюда кажется, что вход нейронной сети есть гармоники. Отсюда возникает интересная связь: если учить персептрон на гармониках, это будет «соответствовать» обучению сверточной сети во временной области, так как

$$\mathcal{F}^{-1}(\hat{\mathbf{W}}\hat{\mathbf{x}}) = \mathcal{F}^{-1}(\hat{\mathbf{W}}) * \mathcal{F}^{-1}(\hat{\mathbf{x}}) = \mathbf{W} * \mathbf{x} = \int \mathbf{W}(t-\tau)\mathbf{x}(\tau)d\tau$$

То есть на уровне махания руками учить:

$$\mathbf{T} \to \text{Perceptron}(\boldsymbol{\Theta}) \to \mathbf{C}^{-1}$$

Равносильно тому, что учить

$$\mathbf{T} \to \mathcal{F}^{-1} \to \text{CNN}(\boldsymbol{\Theta}) \to \mathcal{F} \to \mathbf{C}^{-1}$$

Равносильно с точки зрения выразительной способности, но

- сверточные сети можно учить и применять на данных разных размерностей
- сверточные сети могут дать тот же скор при меньшем числе параметров