Random Features For Large-Scale Kernel Machines

Blanca Cano Camarero

Universidad Autónoma de Madrid

March 22, 2023

Overview

1. Objetives
2. Randomness in neural networks: an overview
3. Randomness in neural networks: an overview
   - Feedforwards Networks with Random Weights
   - Recurrent Networks with random weights
   - Theoretical properties of RC Networks
4. Sharp analysis of low-rank kernel matrix approximations
5. Fast and scalable polynomial kernels via explicit feature maps

1. Nystroem Method vs Random Fourier Features: A Theoretical and Empirical Comparison, Advances in Neural Information Processing Systems 2012
2. Random features for kernel approximation: A survey on algorithms, theory, and beyond
3. Williams, C.K.I. and Seeger, M. “Using the Nystroem method to speed up kernel machines”, Advances in neural information processing systems 2001 T. Yang, Y. Li, M. Mahdavi, R. Jin and Z. Zhou
4. Weighted Sums of Random Kitchen Sinks: Replacing minimization with randomization in learning.
5. **Randomness in neural networks: an overview**
6. **Fast and scalable polynomial kernels via explicit feature maps**
7. On the error of random Fourier features
8. **A survey on large-scale machine learning**
9. **Sharp analysis of low-rank kernel matrix approximations**

Paper information

- Title: **Randomness in Neural Networks: An Overview**.
- Authors: Scardapane, Simone and Wang, Dianhui.
- Year: 2017
- Number of cites: 285.
- Scardapane and Wang (2017)

Drawbacks:

- No experimental results.
- Spend many sections in kernel random features we have already seen.

Other articles

- **A review on neural networks with random weights**.
- Authors: Weipeng Cao and Xizhao Wang and Zhong Ming and Jinzhu Gao.
- Year: 2018.
- Number of cites: 308.
- Cao et al. (2018)

Main idea

Randomization is cheaper than optimization

Three families:

- *Feedforward networks with random weights* (RW-FFN).
- Random features for kernel methods.
- RC framework.

Network Architecture

$$f(x) = \sum_{m=1}^{B} \beta_m h_m(x; w_m) = \beta^T h_(x; w_1, \dots, w_B). \tag{1}$$

where

$$h_m(x) = g(a_m^T x + b_m). \tag{2}$$

Randomness options:

Randomnes in coefficients

Random projection: A.

Randomnes in activation function

In the second family of methods, each function is chosen instead as a radial basis function (RBF), typically of Gaussian shape:

$$h_m(x) = \exp\left\{-a_m\|x - c_m\|_2^2\right\}, \tag{3}$$

Randomly chosen the centers c and scaling factors a_m.

Training

Compute β^*

$$\beta^* = argmin_{\beta} \left\{ \frac{1}{2} \|H\beta - y\|_2^2 + \frac{\lambda}{2} \|\beta\|_{norm} \right\} \tag{4}$$

The minimization problem admit some kinds of algebraic or parallelization optimization.

Things to think!

- Why are we not training with gradient descent in the last layer?.
- Why are we not training some intermediate weights?

They are simple ideas, is weird the have not been done before.

Are truly interesting or randomness is bounded?

Universal Approximation Properties

- The basic theoretical result on RW-FNNs was proven in 1995 by Igelnik and Pao, 35 with later corrections made by Li et al.
- The order of approximation error is $\mathcal{O}(C/\sqrt{B})$, where the constant C is independent of B.
- There is always a non-zero probability of an unlucky draw from the probability distribution which will require re-initialization at some stage of approximation. Furthermore, this rapid convergence of order $1/\sqrt{B}$ is assured only up to a given and fixed tolerance.
- Random-weights models still suffer from design choices, translated in free parameters, which are difficult to set optimally with the current mathematical framework, so practically they involve many trials and cross validation to find a good projection space, on top of the selection of the number of hidden [processing elements] and the nonlinear functions.

RANDOM FEATURES FOR KERNEL APPROXIMATION

We have already delved deeper into the topic through the previous survey.

Recurrent Networks with random weights

Some families:
Internal estate

$$f[n] = h(W_i^r x[n] + W_r^r f[n-1] + W_0^r y[n-1]), \tag{5}$$

Or combination past present.

$$f[n] = \lambda f[n] + (1-\lambda) h(W_i^r x[n] + W_r^r f[n-1] + W_0^r y[n-1]), \tag{6}$$

Tn the following steps:

$$y[n] = (w_i^0)^T x[n] + (W_r^0)^T f[n]. \tag{7}$$

Training same that RW-FNN.

Theoretical properties of RC Networks

- Initialization with uniform distribution in $[-1, 1]$ is not a good idea in dynamic system (could create oscillation or chaotic behaviors) due to vanish.
- Echo state property can be guaranteed almost always by rescaling the matrix W so that its spectral radius is less than 1. (Criterio is a heuristic)

Sharp analysis of low-rank kernel matrix approximations

- Bach (2013)
- Title: Sharp analysis of low-rank kernel matrix approximations.
- Author: Francis Bach.
- Year: 2013.
- Number of cites 316.

Contribution interesting:

- A kind of survey:
  - kernel optimization,
  - analysis of column sampling approximation,
  - randomized dimension reduction,
  - theoretical analysis of predictive performance of kernel methods.
- Number of columns in Nyström.

Contributions

- The rank p can be chosen to be linear in the degrees of freedom associated with the problem, a quantity which is classically used in the statistical analysis of such methods.
- They present in Section 4.4 simple algorithms that have sub-quadratic running time complexity, and, for the square loss, provably exhibit the same predictive performance as classical algorithms than run in quadratic time (or more).
- They provide in Section 4.3 explicit examples of optimal values of the regularization parameters and the resulting degrees of freedom, as functions of the decay of the eigenvalues of the kernel matrix, shedding some light in the joint computational/statistical trade-offs for choosing a good kernel.
- They show that with kernels with fast spectrum decays (such as the Gaussian kernel), computational limitations may prevent exploring the relevant portions of the regularization paths, leading to underfitting.

Degrees of freedom

Degrees of freedom:

$$trK^2(K + n\lambda I)^{-2}. \tag{8}$$

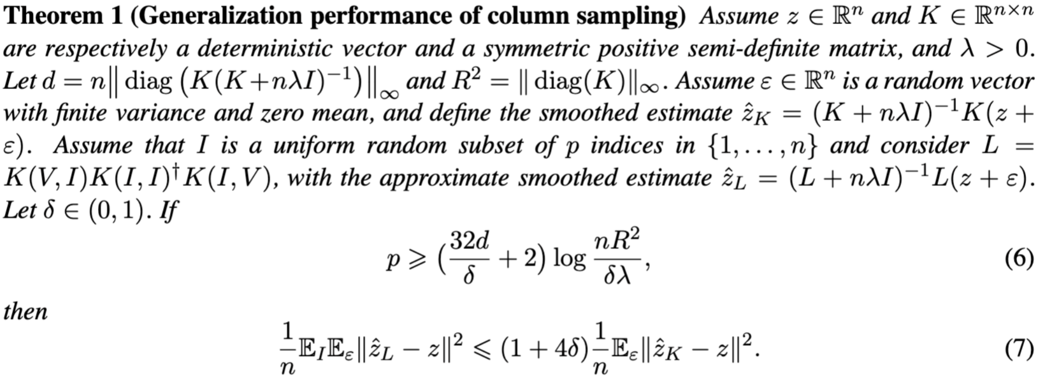
They define *maximal marginal degrees of freedom* d as

$$d = n\|diag(K(K + n\lambda I)^{-1}\|_\infty. \tag{9}$$

d provides an upper-bound on the regular degrees of freedom.

Predictive performance of column sampling

Consider sampling p columns (without replacement) from the original n columns. We consider the column sampling approximation and provide sufficient conditions (a lower-bound on p) to obtain the same predictive performance than with the full kernel matrix.



Theorem 1 (Generalization performance of column sampling) *Assume $z \in \mathbb{R}^n$ and $K \in \mathbb{R}^{n\times n}$ are respectively a deterministic vector and a symmetric positive semi-definite matrix, and $\lambda > 0$. Let $d = n\big\| \operatorname{diag}\big(K(K+n\lambda I)^{-1}\big)\big\|_\infty$ and $R^2 = \|\operatorname{diag}(K)\|_\infty$. Assume $\varepsilon \in \mathbb{R}^n$ is a random vector with finite variance and zero mean, and define the smoothed estimate $\hat{z}_K = (K+n\lambda I)^{-1}K(z+\varepsilon)$. Assume that I is a uniform random subset of p indices in $\{1,\dots,n\}$ and consider $L = K(V,I)K(I,I)^\dagger K(I,V)$, with the approximate smoothed estimate $\hat{z}_L = (L+n\lambda I)^{-1}L(z+\varepsilon)$. Let $\delta \in (0,1)$. If*

$$p \geqslant \Big(\frac{32d}{\delta} + 2\Big)\log\frac{nR^2}{\delta\lambda}, \qquad (6)$$

then

$$\frac{1}{n}\mathbb{E}_I\mathbb{E}_\varepsilon\|\hat{z}_L - z\|^2 \leqslant (1+4\delta)\frac{1}{n}\mathbb{E}_\varepsilon\|\hat{z}_K - z\|^2. \qquad (7)$$

Some observations

- Provides a relative approximation guarantee (small values of δ).
- p seen to by large in practise ...
- Avoiding terms in $1/\lambda$.
- Regularization effects.

Optimal choice of the regularization parameter

- Based on eigenvalues and n.
- Provide some simulations.

Fast and scalable polynomial kernels via explicit feature maps

- Pham and Pagh (2013)
- Cited by 296.
- Year 2013.

Util for us

- Good approach for polynomial kernel

$$k(x,y) = (\langle x,y\rangle + c)^p. \tag{10}$$

Content

- They introduce a fast and scalable randomized tensor product technique for approximating polynomial kernels,
- Accelerating the training of kernel machines.
- By exploiting the connection between tensor product and fast convolution of Count Sketches, our approximation algorithm works in time O(n(d+D log D)) for n training samples in d-dimensional space and D random features.

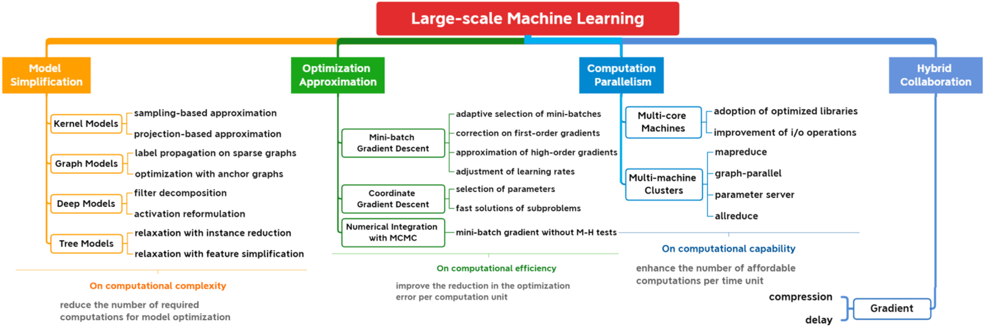
Objetivos semana siguiente

- Disminuir coste computacional SVM (Kernel ridge, GP) - Deep Redes neuronales Ideas simplificación: - RF - Nyström ——————- Contar todas las cosas y hacer experimentos.

Large scale machine Learning

- Cite Wang et al. (2020),
- cited by 60.

Article sections



Large-scale Machine Learning

Model Simplification

- Kernel Models
  - sampling-based approximation
  - projection-based approximation
- Graph Models
  - label propagation on sparse graphs
  - optimization with anchor graphs
- Deep Models
  - filter decomposition
  - activation reformulation
- Tree Models
  - relaxation with instance reduction
  - relaxation with feature simplification

On computational complexity

reduce the number of required computations for model optimization

Optimization Approximation

- Mini-batch Gradient Descent
  - adaptive selection of mini-batches
  - correction on first-order gradients
  - approximation of high-order gradients
  - adjustment of learning rates
- Coordinate Gradient Descent
  - selection of parameters
  - fast solutions of subproblems
- Numerical Integration with MCMC
  - mini-batch gradient without M-H tests

On computational efficiency

improve the reduction in the optimization error per computation unit

Computation Parallelism

- Multi-core Machines
  - adoption of optimized libraries
  - improvement of i/o operations
- Multi-machine Clusters
  - mapreduce
  - graph-parallel
  - parameter server
  - allreduce

On computational capability

enhance the number of affordable computations per time unit

Hybrid Collaboration

- Gradient
  - compression
  - delay

References I

Francis Bach. Sharp analysis of low-rank kernel matrix approximations, 2013.

Weipeng Cao, Xizhao Wang, Zhong Ming, and Jinzhu Gao. A review on neural networks with random weights. *Neurocomputing*, 275:278–287, 2018. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2017.08.040. URL `https://www.sciencedirect.com/science/article/pii/S0925231217314613`.

Ninh Pham and Rasmus Pagh. Fast and scalable polynomial kernels via explicit feature maps. pages 239–247, 08 2013. doi: 10.1145/2487575.2487591.

Simone Scardapane and Dianhui Wang. Randomness in neural networks: An overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 7, 01 2017. doi: 10.1002/widm.1200.

Meng Wang, Weijie Fu, Xiangnan He, Shijie Hao, and Xindong Wu. A survey on large-scale machine learning, 2020.