Journal of Machine Learning **ISSN**: 2790-2048(e), 2790-203X(p)

Learning a Sparse Representation of Barron Functions with the Inverse Scale Space Flow

Tjeerd Jan Heeringa * [1], Tim Roith [2], Christoph Brune [1], and Martin Burger [2,3]

[1]Mathematics of Imaging & AI, University of Twente, Enschede, The Netherlands.
[2]Helmholtz Imaging, Deutsches Elektronen-Synchrotron DESY, Hamburg 22607, Germany.
[3]Fachbereich Mathematik, Universität Hamburg, Hamburg 20146, Germany.

Abstract. This paper presents a method for finding a sparse representation of Barron functions. Specifically, given an L^2 function f, the inverse scale space flow is used to find a sparse measure μ minimising the L^2 loss between the Barron function associated to the measure μ and the function f. The convergence properties of this method are analysed in an ideal setting and in the cases of measurement noise and sampling bias. In an ideal setting the objective decreases strictly monotone in time to a minimizer with $\mathcal{O}(1/t)$, and in the case of measurement noise or sampling bias the optimum is achieved up to a multiplicative or additive constant. This convergence is preserved on discretization of the parameter space, and the minimizers on increasingly fine discretizations converge to the optimum on the full parameter space.

Keywords:
Barron Space,
Bregman Iterations,
Sparse Neural Networks,
Inverse Scale Space,
Optimization.

Article Info.:
Volume: 4
Number: 1
Pages: 48 - 88
Date: March/2025
doi.org/10.4208/jml.240123

Article History:
Received: 23/01/2024
Accepted: 25/12/2024

Communicated by:
Chenglong Bao

1 Introduction

Most neural networks contain a subnetwork with fewer parameters that performs equally well [36], and some of these subnetworks have been found to generalise equally or even better than their dense counterparts [28, 29]. However, it is a priori hard to determine which parameters of the network will be part of the subnetwork. Hence, various approaches have been developed for finding well performing sparse neural network. They fall roughly in three categories. The first is to add a term to the loss or regularizer that promotes sparsity. An example of this would be the least absolute shrinkage and selection operator (LASSO), in which a ℓ^1 regularizer is added [39]. The second is to train a network first and prune it afterwards, meaning weights are reduced with as little as possible influence on the performance [31]. The third is to start with a sparse architecture, and add or remove neurons during training [22].

One of the methods, which starts from a sparse architecture, is based on the Bregman iteration [33]. This method has been introduced and thoroughly analysed for imaging and compressed sensing [15, 17, 44]. The method works in these settings by progressively adding more detail to the reconstructed images and signals, respectively. A limitation of the original method is that it requires that often requires the problem to be convex.

*Corresponding author. t.j.heeringa@utwente.nl

However, adaptations of the method, e.g. the linearized variant in [5, 13], where the loss is replaced by a first order approximation, allows for a successful application to neural networks. A major success of this method is that it is able to find an auto-encoder without ever explicitly defining an auto-encoder like architecture [12]. This shows that it has major potential for automatic neural network architecture design tasks.

1.1 Related work

Bregman iterations were introduced in [33] and further developed and analysed in [1,6,15, 17–19,43,44] as an algorithm to solve sparsity promoting regularisation tasks in computer vision. Linearized Bregman iterations as introduced in [18, 44] can be seen as a generalization of the mirror descent algorithm [4,32] to the non-differentiable, convex case. More recently, variants of the original algorithm have been applied in the context of machine learning, see, e.g. [12, 13, 40, 41].

Bregman iterations are the implicit Euler discretization of an inverse scale space flow. Going to the continuous limit has helped to find easy implementations for relatively complex functionals like the total variation functional, and has helped to obtain well-justified and simple stopping criteria [14]. In the finite-dimensional case of sparse regularization (and further generalizations) an exact time discretization can be found, which leads to efficient methods [15, 30]. We refer to [6] for recent overview.

Similar to inverse scale space flow being the continuous limit of the Bregman iterations, we have that the Barron spaces are the continuous limit of shallow neural network. It was proven that Barron functions have bounded point evaluations [2,38], Barron functions can be approximated in L^p with rate $\mathcal{O}(m^{-1/p})$ [26], Barron spaces have a represented theorem [34] and that Barron spaces are a kind of integral reproducing kernel Banach spaces (RKBS), a Banach space analogue to reproducing kernel Hilbert spaces (RKHS) [2]. The spaces are parametrized by the activation function of the networks. The Barron spaces associated to most of the commonly used non-periodic activation are embedded in the Barron space with ReLU as activation function [27]. This Barron space together with the Barron spaces associated to the rectified power unit (RePU), the higher-order generalization of the ReLU, are strongly related to bounded variation (BV) spaces [26, 34].

A fundamental open question in machine learning is how to find the best function representing your data. For Barron spaces, this means finding the best measure μ representing the Barron function f. Since the relation between μ and f is linear, this leads to a convex minimization problem. Based on an alternative representation of Barron functions in probability space, the authors in [42] formulated a Wasserstein gradient flow for this problem based on the ideas of [21]. Under several assumptions, including omnidirectional initial conditions and satisfying the Morse-Sard property, this leads to a unique solution π [42]. However, not all Barron functions satisfy the Morse-Sard property, placing a limit on the functions that can be represented with this approach [42]. Although this unique solution π represents the Barron function f, it is not necessarily the probability measure for f with the smallest semi-norm. In order to find sparse neural networks, there is a need for a method that minimizes this semi-norm as well.