Kernelized Synaptic Weight Matrices

Lorenz K. Muller [1] **Julien N.P. Martel** [1] **Giacomo Indiveri** [1]

bstract

In this paper we introduce a novel neural network architecture, in which weight matrices are reparametrized in terms of low-dimensional vectors, interacting through kernel functions. layer of our network can be interpreted as introducing a (potentially infinitely wide) linear layer between input and output. We describe the theory underpinning this model and validate it with concrete examples, exploring how it can be used to impose structure on neural networks in diverse applications ranging from data visualization to recommender systems. We achieve state-of-the-art performance in a collaborative filtering task (MovieLens).

1. Introduction

Neural Networks have a large number of free parameters and often training algorithms need to choose between a range of near optimal value assignments for those parameters. This choice can be difficult, because optimality with respect to a training set does not guarantee good behavior on unseen, similar data (especially when there are many free parameters). This defines overfitting. To address this problem, regularization techniques are widely used in Neural Network optimization to help training procedures find generalizable solutions.

In this paper, we show that by expressing the weights of a neural network layer as the kernel interaction of low-dimensional vectors of free parameters, we can embed the weight matrix of a layer in some (potentially high-dimensional) feature-space; the embedding is controlled by the choice of the kernel function. This technique provides a structural way to regularize weight matrices.

[1]Institute of Neuroinformatics, University of Zurich and ETH Zurich, Switzerland. Correspondence to: Lorenz K. Muller <lorenz@ini.ethz.ch>.

Proceedings of the 35^{th} *International Conference on Machine Learning*, Stockholm, Sweden, PMLR 80, 2018.

1.1. Related Work

There exist many approaches that reparametrize the weight matrix of a neural network. (Schmidhuber, 1997) and (Gomez & Schmidhuber, 2005) learned weights for a neural network by training either small programs or another neural network respectively to generate them. The approach of training a neural network to generate weights for another network can take many forms and reoccurs for example in the work of (Stanley et al., 2009; Ha et al., 2016; Fernando et al., 2016). In (Koutnik et al., 2010) the weight matrix of a neural network is decomposed by a discrete cosine transform (DCT) and learning is performed directly on the DCT parameters. Several recent papers propose different types of reparameterizations using various forms of matrix product decomposition (Denil et al., 2013; Moczulski et al., 2015; Tai et al., 2015).

Many methods have been proposed to ensure that at end of the training of a neural network the weights fulfill some desired properties: Weight-decay (or L_2 regularization) (Krogh & Hertz, 1992) makes large entries in the weight matrix costly, encouraging 'simple' models; drop-out (Srivastava et al., 2014) (the inclusion of multiplicative Bernoulli noise) performs model averaging for appropriate network architectures, prevents co-adaptation of different weights, and can be interpreted as letting networks approximate deep Gaussian processes (Gal & Ghahramani, 2016); low-rank decomposition of weight matrices (Sainath et al., 2013) (analogous to separable convolutions in ConvNets (Jaderberg et al., 2014)) is primarily used for computational speed-up and memory footprint reduction, but also has a regularizing effect. s we shall see in this work, low-rank decomposition is a special case of the method we propose in this paper.

In contrast to these approaches, our method allows the imposition of low-dimensional structure onto the network. We will show on several examples that the low dimensional embedding of the network weights does not only reduce the number of free parameters in a network, but has the advantage of increasing its interpretability as well as allowing for structural regularization.

1.2. Core Idea

Instead of assigning an individual, free weight-parameter between each input and output neuron of a neural network