Neural Networks 95 (2017) 110–133

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Learning in the machine: The symmetries of the deep learning channel

Pierre Baldi [a,*], Peter Sadowski [a], Zhiqin Lu [b]

[a] Department of Computer Science, University of California, Irvine, Irvine, CA 92617, United States
[b] Department of Mathematics, University of California, Irvine, Irvine, CA 92617, United States

ARTICLE INFO

Article history:
Received 5 February 2017
Received in revised form 30 July 2017
Accepted 22 August 2017
Available online 5 September 2017

Keywords:
Neural networks
Deep learning
Backpropagation
Local learning
Learning channel
Learning dynamics

ABSTRACT

In a physical neural system, learning rules must be local both in space and time. In order for learning to occur, non-local information must be communicated to the deep synapses through a communication channel, the deep learning channel. We identify several possible architectures for this learning channel (Bidirectional, Conjoined, Twin, Distinct) and six symmetry challenges: (1) symmetry of architectures; (2) symmetry of weights; (3) symmetry of neurons; (4) symmetry of derivatives; (5) symmetry of processing; and (6) symmetry of learning rules. Random backpropagation (RBP) addresses the second and third symmetry, and some of its variations, such as skipped RBP (SRBP) address the first and the fourth symmetry. Here we address the last two desirable symmetries showing through simulations that they can be achieved and that the learning channel is particularly robust to symmetry variations. Specifically, random backpropagation and its variations can be performed with the same non-linear neurons used in the main input–output forward channel, and the connections in the learning channel can be adapted using the same algorithm used in the forward channel, removing the need for any specialized hardware in the learning channel. Finally, we provide mathematical results in simple cases showing that the learning equations in the forward and backward channels converge to fixed points, for almost any initial conditions. In symmetric architectures, if the weights in both channels are small at initialization, adaptation in both channels leads to weights that are essentially symmetric during and after learning. Biological connections are discussed.

1. Introduction

Backpropagation implemented in digital computers has been successful at addressing a host of difficult problems ranging from computer vision (He, Zhang, Ren, & Sun, 2015; Krizhevsky, Sutskever, & Hinton, 2012; Srivastava, Greff, & Schmidhuber, 2015; Szegedy et al., 2015) to speech recognition (Graves, Mohamed, & Hinton, 2013) in engineering, and from high energy physics (Baldi, Sadowski, & Whiteson, 2014; Sadowski, Collado, Whiteson, & Baldi, 2015) to biology (Agostinelli, Ceglia, Shahbaba, Sassone-Corsi, & Baldi, 2016; Di Lena, Nagata, & Baldi, 2012; Zhou & Troyanskaya, 2015) in the natural sciences. Furthermore, recent results have shown that backpropagation is optimal in some sense (Baldi & Sadowski, 2016). However, backpropagation implemented in digital computers is not the real thing. It is merely a digital emulation of a learning process occurring in an idealized physical neural system. Thus thinking about learning in this digital simulation can be useful but also misleading, as it often obfuscates fundamental issues. Thinking about learning in physical neural systems or learning in the machine – biological or other – is useful not only for better understanding how specific or idealized machines can learn, but also to better understand fundamental, hardware-independent, principles of learning. And, in the process, it may occasionally also be useful for deriving new approaches and algorithms to improve the effectiveness of digital simulations and current applications.

Thinking about learning in physical systems first leads to the notion of locality (Baldi & Sadowski, 2016). In a physical system, a learning rule for adjusting synaptic weights can only depend on variables that are available locally in space and time. This in turn immediately identifies a fundamental problem for backpropagation in a physical neural system and leads to the notion of a learning channel. The critical equations associated with backpropagation show that the deep weights of an architecture must depend on non-local information, such as the targets. Thus a channel must exist for communicating this information to the deep synapses—this is the learning channel (Baldi & Sadowski, 2016).

Depending on the hardware embodiment, several options are possible for implementing the learning channel. A first possibility is to use the forward connections in the reverse direction. A second

* Corresponding author.
E-mail address: pfbaldi@uci.edu (P. Baldi).
URL: http://www.ics.uci.edu/~pfbaldi (P. Baldi).

http://dx.doi.org/10.1016/j.neunet.2017.08.008
0893-6080/

possibility is to use two separate channels with different characteristics and possibly different hardware substrates in the forward and backward directions. These two cases will not be further discussed here. The third case we wish to address here is when the learning channel is a separate channel but it is similar to the forward channel, in the sense that it uses the same kind of neurons, connections, and learning rules. Such a learning channel is faced with at least six different symmetry challenges: (1) symmetry of architectures; (2) symmetry of weights; (3) symmetry of neurons; (4) symmetry of derivatives; (5) symmetry of processing; and 6) symmetry of learning rules, where in each case the corresponding symmetry is in general either desirable (5–6) or undesirable (1–4).

In the next sections, we first identify the six symmetry problems and then show how they can be addressed within the formalism of simple neural networks. While biological neural networks remain the major source of inspiration for this work, the analyses derived are more general and not tied to neural computing in any particular substrate.

2. The learning channel and the symmetry problems

2.1. Basic notation

Throughout this paper, we consider layered feedforward neural network architectures and supervised learning tasks. We will denote such an architecture by

$$\mathcal{A}[N_0, \ldots, N_h, \ldots, N_L] \tag{1}$$

where N_0 is the size of the input layer, N_h is the size of hidden layer h, and N_L is the size of the output layer. For simplicity, we assume that the layers are fully connected and let w_{ij}^h denote the weight connecting neuron j in layer $h-1$ to neuron i in layer h. The output O_i^h of neuron i in layer h is computed by:

$$O_i^h = f_i^h(S_i^h) \quad \text{where} \quad S_i^h = \sum_j w_{ij}^h O_j^{h-1}. \tag{2}$$

The transfer functions f_i^h are usually the same for most neurons, with typical exceptions for the output layer, and usually are monotonic increasing functions. Typical functions used in artificial neural networks are: the identity, the logistic function, the hyperbolic tangent function, the rectified linear function, and the softmax function.

We assume that there is a training set of M examples consisting of input-target pairs $(I(t), T(t))$, with $t = 1, \ldots, M$. $I_i(t)$ refers to the ith component of the tth training example, and similarly for $T_i(t)$. In addition there is an error function $\mathcal{E}$ to be minimized by the learning process. In general, we will assume standard error functions, such as the squared error in the case of regression problems with identity transfer functions in the output layer, or relative entropy in the case of classification problems with logistic (two-class) or softmax (multi-class) transfer functions in the output layer, although this is not an essential point. The error function is a differentiable function of the weights and its critical points are given by the equations $\partial\mathcal{E}/\partial w_{ij}^h = 0$.

2.2. Local learning

In a physical neural system, learning rules must be local (Baldi & Sadowski, 2016), in the sense that they can only involve variables that are available locally in both space and time, although for simplicity here we will focus primarily on locality in space. Thus typically, in the present formalism, a local learning rule for a deep layer is of the form:

$$\Delta w_{ij}^h = F(O_i^h, O_j^{h-1}, w_{ij}^h) \tag{3}$$

while for the top layer:

$$\Delta w_{ij}^L = F(T_i, O_i^L, O_j^{L-1}, w_{ij}^L) \tag{4}$$

assuming that the targets are local variables for the top layer. Hebbian learning (Hebb, 1949) is a form of local learning. Deep local learning corresponds to stacking local learning rules in a feedforward neural network. Deep local learning using Hebbian learning rules has been proposed by Fukushima (1980) to train the neocognitron architecture, essentially a feed forward convolutional neural network inspired by the earlier neurophysiological work of Hubel and Wiesel (1962). However, in deep local learning, information about the targets cannot be propagated to the deep layers and therefore in general deep local learning cannot find solutions of the critical equations, and thus cannot succeed at learning complex functions in any optimal way.

2.3. The learning channel

Ultimately, for optimal learning, all the information required to reach a critical point of $\mathcal{E}$ must appear in the learning rule of the deep weights. Setting the gradient (or the backpropagation equations) to zero shows immediately that in general at a critical point all the deep synapses must depend on the target or the error information, and this information is not available locally (Baldi & Sadowski, 2016). Thus, to enable efficient learning, there must exist a communication channel to communicate information about the targets or the errors to the deep weights. This is the deep learning channel or, in short, the learning channel. Note that the learning channel is different from the typical notion of "feedback". Although feedback and learning may share the same physical connections, these refer in general to two different processes that often operate at very different time scales, the feedback being fast compared to learning.

In a learning machine, one must think about the physical nature of the channel. A first possibility is to use the forward connections in the reverse direction. This is unlikely to be the case in biological neural systems, in spite of known example of retrograde transmission, as discussed later in Section 6. A second possibility is to use two separate channels with different characteristics and possibly different hardware substrates in the forward and backward directions. As a thought experiment, for instance, one could imagine using electrons in one direction, and photons in the other. Biology can easily produce many different types of cells, in particular of neurons, and conceivably it could use special kinds of neurons in the learning channel, different from all the other neurons. While this scenario is discussed in Section 6, in general it does not seem to be the most elegant or economical solution as it requires different kinds of hardware in each channel. In any case, regardless of biological considerations, we are interested here in exploring the case where the learning channel is as similar as possible to the forward channel, in the sense of being made of the same hardware, and not requiring any special accommodations. However, at the same time, we also want to get rid of any undesirable symmetry properties and constraints, as discussed below. This leads to six different symmetry challenges, four undesirable and two desirable ones.

2.4. The symmetry problems

Symmetry of Architectures [ARC]: Symmetry of architectures refers to having the exact same architecture in the forward and in the backward channel, with the same number of neurons in each hidden layer and the same connectivity. This corresponds to the Bidirectional, Conjoined, and Twin cases defined below. In the Bidirectional and Conjoined case the Symmetry of Architectures is