Ghost Units Yield Biologically Plausible Backprop in Deep Neural Networks

Thomas Mesnard∗(thomas.mesnard@gmail.com)
Montreal Institute for Learning Algorithms, University of Montreal
Montreal H3T 1J4, Quebec, Canada

Gaëtan Vignoud∗(gaetan.vignoud@gmail.com)
Montreal Institute for Learning Algorithms, University of Montreal
Montreal H3T 1J4, Quebec, Canada

Walter Senn (senn@pyl.unibe.ch)
Department of Physiology, University of Bern
Bern 3012, Switzerland

Yoshua Bengio (yoshua.umontreal@gmail.com)
Montreal Institute for Learning Algorithms, University of Montreal
Montreal H3T 1J4, Quebec, Canada

Abstract

In the past few years, deep learning has transformed artificial intelligence research and led to impressive performances in various difficult tasks. However, it is still unclear how the brain can perform credit assignment as efficiently as backpropagation does in deep neural networks. In this paper, we introduce a model that relies on a new type of neurons, Ghost Units, that enable the network to backpropagate errors and do efficient credit assignment in deep structures. While requiring very few biological assumptions, it is able to achieve great performances on classification tasks. Error backpropagation occurs through the network dynamics itself and thanks to biologically plausible local learning rules. In particular, it does not require separate feedforward and feedback circuits. Therefore, this model is a step towards understanding how learning and memory are computed in cortical multilayer structures, but also raises interesting questions about neuromorphic hardware applications.

Keywords: bio backprop; credit assignment; feedback alignment;

Introduction

Recently, deep learning has revolutionized artificial intelligence. Thanks to backpropagation (Almeida, 1987; Pineda, 1987), deep neural networks take advantage of the multilayer structure of the network to learn high level features relevant for the tasks it is trained to perform.

However, how the brain is able to perform credit assignment in deep structures is still an open question. The core algorithm used to train deep neural networks (i.e backpropagation) is seen by the neuroscience community as being biologically implausible because the implementation used in deep learning relies on assumptions that cannot be met in the brain (Bengio, Lee, Bornschein, Mesnard, & Lin, 2015; Neftci, Augustine, Paul, & Detorakis, 2017).

In this paper, we are considering a recurrent network composed of pyramidal units (PU) that can be identified with the units that are used in a multilayer Perceptron. These cells integrate feedforward activity coming from the lower layers and feedback activity coming from the upper layers. However, in order to backpropagate the error, we introduce a new type of interneurons, referred to as ghost units (GU). Their goal is to predict and cancel feedback from pyramidal units in the upper layer by integrating the same feedforward input but without receiving any feedback from the upper layers. This property enables the network to converge quickly during the feedforward path, in spite of the presence of recurrent connections, by canceling feedback coming from the upper layers thanks to the ghost units. This cancellation effect of the ghost units allows top-down corrective feedback to be correctly backpropagated when targets are provided in the nudging phase. This gives the network the capacity to perform credit assignment in the multilayer structure by simply following its dynamics and updating the weights according to local contrastive Hebbian plasticity learning rules. Links can be drawn with (Sacramento, Costa, Bengio, & Senn, 2018; Jaderberg et al., 2016) where local credit assignment is also performed, with (Sacramento et al., 2018) having introduced the idea of canceling the downstream feedback with the inhibitory lateral feedback in order to leave out only the backpropagated error as remaining from the feedback signal. Here however, we show that it is possible to approximate the backpropagated gradient without leading to an exponential decay of its magnitude as it is propagated through more layers.

Backpropagation thanks to Ghost Units

Architecture

We consider here a biologically plausible implementation for backpropagation of a directed acyclic graph of feedforward connections with network input $x$, target output $y$ and a scalar $L(\hat{y}, y)$ which somehow compares the network output $\hat{y}$ with the target output $y$. 


∗Both authors contributed equally to this paper. Alphabetical order.
Each node of this graph is associated with one or more pyramidal units (PU) whose activity is denoted by \( s_i \) (the state of unit \( i \)), which will be referred to as unit \( i \). Pyramidal units have an output non-linearity \( \rho \) which maps their activity \( s_i \) to their firing rate \( \rho(s_i) \). Both feedforward and feedback connections are considered in this model. The main feedforward synaptic weights \( W^f_{ij} \) correspond to the influence of presynaptic unit \( j \) on the downstream post-synaptic unit \( i \). The feedback weights \( W^b_{ij} \) encapsulate the effect of downstream unit \( i \) on upstream unit \( j \). Note that a single feedforward unit (called pyramidal unit) could in reality be implemented by multiple pyramidal units with similar input and output connectivity, allowing the network to reduce the spiking noise.

We also consider a lateral network of ghost units (GU), which can be identified as inhibitory interneurons. These neurons are represented by a scalar variable \( g_i \) for each unit \( i \). A ghost neuron in a layer is connected to the pyramidal units of the previous layer only, through two matrices \( V^f_{ij} \) (for connections from the pyramidal unit \( j \) to ghost unit \( i \) of the previous) and \( V^b_{ij} \) (for the feedback connections of the ghost unit \( i \) to the pyramidal unit \( j \)). These neurons aim to reproduce the feedback activity of the pyramidal units during the forward phase, which enable the network to compute directly the gradient during the nudging phase. These neurons are considered as inhibitory when projecting to the previous layer (which means that their contribution to the activity of the pyramidal units is \( -V^b_{ij}(g_i) \)).

Training is decomposed in a prediction phase and a nudging phase, following (Scellier & Bengio, 2016). During the prediction phase, the network evolves thanks to its recurrent dynamics with only inputs provided. During the nudging phase, both inputs and targets are presented to the network. A top-down error signal \(-\nabla L(\hat{y}, y)\) pushes the output units \( s_k \) towards a value corresponding to a smaller loss \( L \). We show that the combination of lateral recurrent and feedback connections propagates this error into the network in a way that closely approximates backpropagation, so long as some assumptions are satisfied, regarding the ability of feedback connections to mimic feedforward connections (approximate symmetry) and of lateral connections to learn to cancel the feedback connections when there is no nudging.

**Dynamics of the units**

We distinguish three types of inputs arriving into each pyramidal unit \( i \):

- \( b_i = \sum_j W^f_{ij} \rho(s_j) \) is the bottom-up input on the feedforward path from the pyramidal units
- \( t_i = \sum_j W^b_{ij} \rho(s_j) \) is the top-down feedback onto unit \( s_i \) from its successors.
- \( c_i = \sum_j V^b_{ij} \rho(g_j) \) is the top-down feedback onto unit \( s_i \) from the ghost units of its successors.

We denote \( e_i = t_i - c_i \) being the error signal, which we propose will indicate in which direction \( s_i \) should move to reduce \( L \). The pyramidal units are evolving through:

\[
\tau \dot{s}_i = -s_i + b_i + t_i - c_i = -s_i + b_i + e_i \quad (1)
\]

The ghost units follow:

\[
\tau \dot{g}_i = -g_i + \sum_j V^f_{ij} \rho(s_j) \quad (2)
\]

which makes \( g_i \) converge to \( \sum_j V^f_{ij} \rho(s_j) \).

**Different architectures and learning procedures**

**Network with 1-1 correspondence between the pyramidal units and the ghost units (MA)**

**Model description** In this section, we consider that each pyramidal unit has a corresponding ghost unit\(^1\). In order to make the reading easier in this part, we will use the same indices for the ghosts units and the pyramidal ones. For example, ghost units \( g_i \) will be associated to the pyramidal unit \( s_i \).

This architecture can be seen in Fig. 1.

![Architecture of the proposed network](image)

(a) Connectivity between pyramidal units (in grey, \( s_j \)) and ghost units (in red, \( g_i \)).

(b) Architecture of the proposed network

**Figure 1:** Architecture for model A (MA) network with one-to-one correspondence between each ghost unit and each pyramidal unit.

The local learning rules for the different synaptic weights are defined as follow:

- \( s_i \) acts like a target for the ghost unit \( g_i \) to learn \( V^f_{ij} \):

\[
\Delta V^f_{ij} \propto (s_i - g_i) \rho(s_j) \quad (3)
\]

This minimizes \( ||s - g||^2 \), i.e., the inhibitory ghost unit learns to imitate its associated pyramidal unit

\(^1\) There are several times more pyramidal units than inhibitory units, but a single inhibitory neuron could be associated with multiple pyramidal units forming a pyramidal unit.
We have also developed a less constrained version of the equations (6), which minimizes $||r-c||^2$.

\[
\Delta V_{ij}^f \propto (r_i - c_i) p(g_j).
\]

This minimizes $||r-c||^2$.

- The main weights (feedforward from pyramidal unit $j$ to pyramidal unit $i$) are updated using a Hebbian learning rule:
  \[
  \Delta W_{ij}^f \propto e_i p'(s_i) p(s_j).
  \]
  We show that this approximates gradient descent on $L$.

- The feedback weights are set equal to the transpose of the feedforward ones: $W_{ij}^b = W_{ji}^f$.

A good approximation of backpropagation We define two loss functions $C_1$ and $C_2$, derived from eq. (3) and eq. (4), which are:

\[
C_1 = ||s-g|| \quad \text{and} \quad C_2 = ||t-c||
\]

Eq. [5] can be justified as follows.

**Theorem 1.** If $W_{ij}^b \approx W_{ij}^f$, $C_1 \to 0$, $C_2 \to 0$ in the prediction phase and $e_i$ is small, in the sense that $p'(s_i + e_i) \approx p'(s_i)$, and if for output units we set $t_k$ such that during the nudging phase $e_k \approx -\frac{\partial L}{\partial p(t_k)}$, then the nudged network converges to $e_i \approx -\frac{\partial L}{\partial p(t_i)}$ for hidden units $s_i$. This shows that the feedback weights, thanks to the ghost units, backpropagate error gradients.

Deep neural network with Ghost Units replicating online the feedback from the pyramidal units (MB)

We have also developed a less constrained version of the model MA, that we quickly introduce in this section. In this model (MB), we do not make any hypothesis on the number of pyramidal units and ghost units. We also consider that the feedforward connections $V_{ij}^f$ from the pyramidal unit $i$ to the ghost unit $j$ are fixed to a randomly initialized value, therefore $\Delta V_{ij}^f = 0$. The ghost units only aim to match, example by example, the feedback $c_i$, coming from the ghost units to the feedback $b_i$ coming from the pyramidal units. Thanks to this property, $c_i = b_i$ for a given example at the end of the prediction phase after learning of $V_{ij}^f$. This enables the network to correctly learn $W_{ij}^f$ in the nudging phase.

As just described, the top-down feedback $r_i$ onto unit $i$ acts as a target for the weights $V_{ij}^f$ forming the canceling feedback $c_i$. Therefore, the weights are updated as follow in the prediction phase:

\[
\Delta V_{ij}^f \propto (r_i - c_i) p(g_j).
\]

which minimizes $||r-c||^2$. The main weights $W_{ij}^f$ evolve in the nudging phase through the same Hebbian rule as in (MA):

\[
\Delta W_{ij}^f \propto e_i p'(s_i) p(s_j).
\]

It can be also proved that under some assumptions this strategy is equivalent to backpropagation.

**Initialization**

while not done do

Sample batch from the training set

for $k$ in range prediction_steps do

for output units $e_i = 0$

for hidden units $e_i = \sum_j W_{ij}^b p(s_j) - V_{ij}^f p(g_j)$

\[\forall \text{ units } i, s_i \leftarrow s_i + \frac{\partial}{\partial e_i} [s_i + \sum_j W_{ij}^f p(s_j) + e_i]
\]

$g_i \leftarrow g_i + \frac{\partial}{\partial g_i} [g_i + \sum_j V_{ij}^f p(s_j)]$

$V_{ij}^f \leftarrow V_{ij}^f + \eta_d \partial [L_i v p(s_j)]$

$V_{ij}^b \leftarrow V_{ij}^b + \eta_d \partial [L_i p(g_j)]$

end

for $k$ in range nudging_steps do

for output units $e_i = -\beta_{\text{ac}}$

for hidden units $e_i = \sum_j W_{ij}^b p(s_j) - V_{ij}^b p(g_j)$

\[\forall \text{ units } i, s_i \leftarrow s_i + \frac{\partial}{\partial e_i} [s_i + \sum_j W_{ij}^f p(s_j) + e_i]
\]

$g_i \leftarrow g_i + \frac{\partial}{\partial g_i} [g_i + \sum_j V_{ij}^f p(s_j)]$

$W_{ij}^f \leftarrow W_{ij}^f + \eta_d \partial [L_i v p(s_j)]$

$W_{ij}^b \leftarrow W_{ij}^b + \eta_d \partial [L_i p(g_j)]$

end

end

**Algorithm 1:** Learning procedure for model A (MA) network.

**Transverse feedback versus Feedback alignment**

The feedback weights $W_{ij}^b$ are initially supposed to be equal to the transpose of the feedforward ones: $W_{ij}^b = W_{ji}^f$, as in classical backpropagation. We refer to this hypothesis as transpose-feedback (TF). In practice, this hypothesis could be implemented using an addition reconstruction cost (between consecutive pyramidal layers), which has been shown to encourage symmetry of the weights [Vincent, Larochelle, Lajoie, Bengio, & Manzagol, 2010]. This assumption can also be relaxed thanks to [Lillicrap, Cownden, Tweed, & Akerman, 2016] and the feedback alignment effect (FA). In this case, feedback weights $W_{ij}^b$ are supposed to be fixed and randomly initialized. During learning, the feedforward matrix tends to align with the transpose of the feedback matrix. Both hypothesis (TF and FA) were studied here.

**Results**

**Credit assignment with replicating units**

We consider here a 500-unit network with 1 hidden layer with MSE loss. No preconditioning of the inputs is used. Batch size is equal to 100 and the activation is sigmoid. Figure 2 shows the learning dynamics for a network during training, following MA assumptions (mean for 5 experiments).
In MA, learning is composed of two phases. During the prediction phase only inputs are provided to the network. The weights $V^f$ push the ghost unit to mimic its corresponding pyramidal units while having $V^b$ learning to minimize the mismatch between the feedback coming from the ghost unit and its corresponding pyramidal unit. This pushes the matrix $V^f$ to reproduce $W^f$ and $V^b$ to copy $W^b$. This can be seen at the bottom of Figure 2 where the Frobenius norms between these matrices during training are reproduced.

This leads to the correct computation of the feedforward path because the feedback terms cancel each other, despite happening in a dynamical and imperfect way.

During the nudging phase, the output units are nudged towards the correct values. This shift is backpropagated through the dynamics of the network and gives rise to an error term in each hidden layer thanks to the mismatch between the feedforward coming from a pyramidal unit and its corresponding ghost unit. $W^f$ evolves in order to minimize this mismatch. As can be seen at the top (Accuracy) Figure 2 the network learns to classify correctly MNIST digits. It generalizes well without any regularization or tricks.

### Classification on MNIST

![Figure 2: Learning in a 500-unit MA neural network](image)

We tested several types of networks on the MNIST dataset:

- model MA (see Figure 3) or MB
- Transpose-feedback (TF) or Feedback alignment (FA)
- different number of units per hidden layer
- different numbers of hidden layers (data not shown)

We ran 5 experiments for each model, and represent these results in Table 1. These results approach state-of-the-art accuracies for multilayer Perceptron, with MA performing slightly better than MB.

<table>
<thead>
<tr>
<th>Architecture</th>
<th>TF</th>
<th>FA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>#units</td>
<td>Train</td>
</tr>
<tr>
<td>MA 100</td>
<td>99.97</td>
<td>97.69</td>
</tr>
<tr>
<td>MA 300</td>
<td>100</td>
<td>98.22</td>
</tr>
<tr>
<td>MA 500</td>
<td>100</td>
<td>98.29</td>
</tr>
<tr>
<td>MB 100</td>
<td>99.31</td>
<td>97.22</td>
</tr>
<tr>
<td>MB 300</td>
<td>99.7</td>
<td>98.05</td>
</tr>
<tr>
<td>MB 500</td>
<td>99.76</td>
<td>98.13</td>
</tr>
</tbody>
</table>

Table 1: Accuracy results on MNIST with MA and MB in a network with one hidden layer.

## Conclusion

The model presented here develops a hypothesis for explaining how the brain is able to do credit assignment in deep neural architectures. By introducing inhibitory neurons called Ghost Units, the network is able to locally compute errors in each hidden layer and use this error learn useful representations.

### References


