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 ρ 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_{ij}^f correspond to the influence of presynaptic unit j on the downstream post-synaptic unit i. The feedback weights W_{ij}^b 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_l for each unit l. A ghost neuron in a layer is connected to the pyramidal units of the previous layer only, through two matrices V_{lj}^f (for connections from the pyramidal unit j to ghost unit l of the previous) and V_{jl}^b (for the feedback connections of the ghost unit l 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 $-V^b\rho(g)$).

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 $-\frac{\partial L(\hat{y},y)}{\partial s_k}$ 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.

Notations

x	network input	y	target output
PU	pyramidal unit	GU	ghost unit
s_i	activity of PU i	g_l	activity of GU l
$\hat{y}$	predicted output	$L(\hat{y},y)$	loss function
ρ	non-linearity function	τ	time constant
W_{ij}^f	feedforward connection from PU j to PU i		
W_{ij}^b	feedback connection from PU j to PU i		
V_{lj}^f	lateral (recurrent) connection from PU j to GU l		
V_{il}^b	lateral (recurrent) connection from GU l back to PU i		

[†]In this article, we do not enforce that outgoing synaptic weights from inhibitory interneurons are actually negative.

Dynamics of the units

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

- $b_i = \sum_j W_{ij}^f \rho(s_j)$ is the bottom-up input on the feedforward path from the pyramidal units
- $t_i = \sum_j W_{ij}^b \rho(s_j)$ is the top-down feedback onto unit s_i from its successors.
- $c_i = \sum_l V_{il}^b \rho(g_l)$ 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 :

$$\tau\dot{s}_i = -s_i + b_i + t_i - c_i = -s_i + b_i + e_i \tag{1}$$

The ghost units follow :

$$\tau\dot{g}_l = -g_l + \sum_j V_{lj}^f \rho(s_j) \tag{2}$$

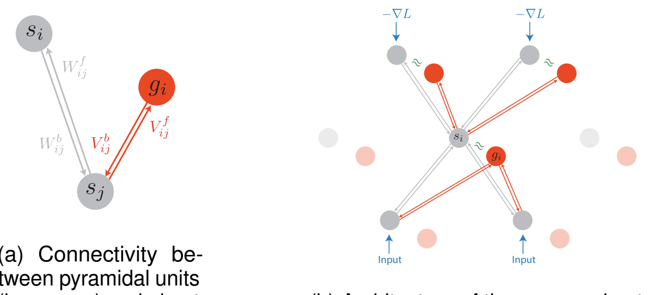
which makes g_l converge to $\sum_j V_{lj}^f \rho(s_j)$.

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[‡]. 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.



(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_{ij}^f:

$$\Delta V_{ij}^f \propto (s_i - g_i)\rho(s_j) \tag{3}$$

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

[‡]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

- The top-down feedback t_i onto unit i acts as a target for the weights forming the canceling feedback c_i:
$$\Delta V^b_{ij} \propto (t_i - c_i)\rho(g_j). \qquad (4)$$
This minimizes $||t-c||^2$.
- The main weights (feedforward, from pyramidal unit j to pyramidal unit i) are updated using a Hebbian learning rule :
$$\Delta W^f_{ij} \propto e_i \rho'(s_i)\rho(s_j). \qquad (5)$$
We show that this approximates gradient descent on L.
- The feedback weights are set equal to the transpose of the feedforward ones: $W^b_{ij} = W^f_{ji}$.

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

$$C_1 = ||s-g|| \text{ and } C_2 = ||t-c||$$

Eq. 5 can be justified as follows.

Theorem 1. *If $W^b_{ji} \approx W^f_{ij}$, $C_1 \to 0$, $C_2 \to 0$ in the prediction phase and e_i is small, in the sense that $\rho'(s_i+e_i) \approx \rho'(s_i)$, and if for output units we set t_k such that during the nudging phase $e_k \overset{\propto}{\sim} -\frac{\partial L}{\partial \rho(s_k)}$, then the nudged network converges to $e_i \overset{\propto}{\sim} -\frac{\partial L}{\partial \rho(s_i)}$ 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^f_{li} from the pyramidal unit i to the ghost unit l are fixed to a randomly initialized value, therefore $\Delta V^f_{lj} = 0$. 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^{il}_b. This enables the network to correctly learn W^f_{ij} in the nudging phase.

As just described, the top-down feedback t_i onto unit i acts as a target for the weights V^b_{il} forming the canceling feedback c_i. Therefore, the weights are updated as follow in the prediction phase:
$$\Delta V^b_{il} \propto (t_i - c_i)\rho(g_l) \qquad (6)$$
which minimizes $||t-c||^2$. The main weights W^f_{ij} evolve in the nudging phase through the same Hebbian rule as in (MA):
$$\Delta W^f_{ij} \propto e_i \rho'(s_i)\rho(s_j). \qquad (7)$$

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 $e_i = \sum_j W^b_{ij}\rho(s_j) - V^b_{ij}\rho(g_j)$
        $\forall$ units $i$,
        $s_i \leftarrow s_i + \frac{dt}{\tau}[-s_i + \sum_j W^f_{ij}\rho(s_j) + e_i]$
        $g_i \leftarrow g_i + \frac{dt}{\tau}[-g_i + \sum_j V^f_{ij}\rho(s_j)]$
        $V^f_{ij} \leftarrow V^f_{ij} + \eta_V \, dt[(s_i - g_i)\rho(s_j)]$
        $V^b_{ij} \leftarrow V^b_{ij} + \eta_V \, dt[e_i\rho(g_j)]$
    end
    for k in range nudging_steps do
        for output units $e_i = -\beta\frac{\partial C}{\partial \rho(s)}$
        for hidden units $e_i = \sum_j W^b_{ij}\rho(s_j) - V^b_{ij}\rho(g_j)$
        $\forall$ units $i$,
        $s_i \leftarrow s_i + \frac{dt}{\tau}[-s_i + \sum_j W^f_{ij}\rho(s_j) + e_i]$
        $g_i \leftarrow g_i + \frac{dt}{\tau}[-g_i + \sum_j V^f_{ij}\rho(s_j)]$
        $W^f_{ij} \leftarrow W^f_{ij} + \eta_W \, dt[e_i\rho'(s_i)\rho(s_j)]$
        $W^b_{ij} \leftarrow W^f_{ji}$
    end
end
```
Algorithm 1: Learning procedure for model A (MA) network.

Transpose feedback versus Feedback alignment

The feedback weights W^b_{ij} are initially supposed to be equal to the transpose of the feedforward ones: $W^b_{ij} = W^f_{ji}$, 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^b_{ij} 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).

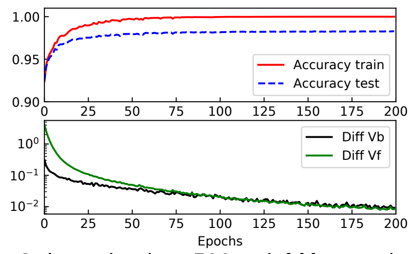


Figure 2: Learning in a 500-unit MA neural network

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 feedback 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

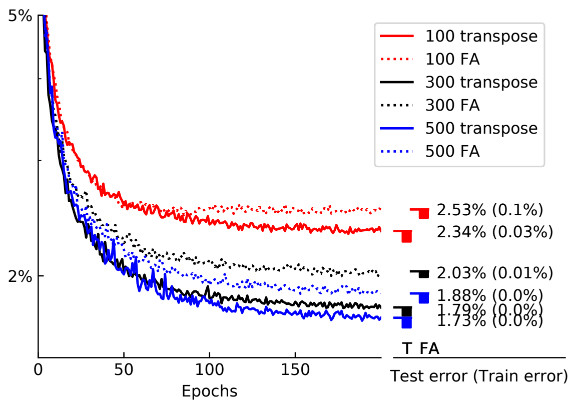


Figure 3: Test and train error on MNIST for MA

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.

Architecture		TF		FA	
Model	#units	Train	Test	Train	Test
MA	100	99.97	97.69	99.88	97.50
MA	300	100	98.22	99.99	97.98
MA	500	100	98.29	100	98.10
MB	100	99.31	97.22	98.93	97.39
MB	300	99.7	98.05	99.48	97.98
MB	500	99.76	98.13	99.56	98.01

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

Almeida, L. B. (1987). A learning rule for asynchronous perceptrons with feedback in a combinatorial environment. In *Proceedings, 1st first international conference on neural networks* (Vol. 2, pp. 609–618).

Bengio, Y., Lee, D.-H., Bornschein, J., Mesnard, T., & Lin, Z. (2015). Towards biologically plausible deep learning. *arXiv preprint arXiv:1502.04156*.

Jaderberg, M., Czarnecki, W. M., Osindero, S., Vinyals, O., Graves, A., & Kavukcuoglu, K. (2016). Decoupled neural interfaces using synthetic gradients. *arXiv preprint arXiv:1608.05343*.

Lillicrap, T. P., Cownden, D., Tweed, D. B., & Akerman, C. J. (2016). Random synaptic feedback weights support error backpropagation for deep learning. *Nature communications*, *7*, 13276.

Neftci, E. O., Augustine, C., Paul, S., & Detorakis, G. (2017). Event-driven random back-propagation: Enabling neuromorphic deep learning machines. *Frontiers in neuroscience*, *11*, 324.

Pineda, F. J. (1987). Generalization of backprop to recurrent neural networks. *Physical review letters*, *59*(19), 2229.

Sacramento, J., Costa, R. P., Bengio, Y., & Senn, W. (2018). Dendritic error backpropagation in deep cortical microcircuits. *arXiv preprint arXiv:1801.00062*.

Scellier, B., & Bengio, Y. (2016). Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. *arXiv:1602.05179*.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P.-A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Machine Learning Res.*, *11*.