UBAL: a Universal Bidirectional Activation-based Learning Rule for Neural Networks

Kristína Malinovská
kristina.malinovska@cvut.cz
Czech Technical University in Prague,
Czech Institute of Informatics,
Robotics, and Cybernetics
Prague, Czech Republic

Ľudovít Malinovský
ludovit.malinovsky@gmail.com
Independent Researcher
Bratislava, Slovakia

Pavel Krsek
pavel.krsek@cvut.cz
Czech Technical University in Prague,
Czech Institute of Informatics,
Robotics, and Cybernetics
Prague, Czech Republic

Svatopluk Kraus
svatopluk.kraus@cvut.cz
Czech Technical University in Prague,
Czech Institute of Informatics,
Robotics, and Cybernetics
Prague, Czech Republic

Igor Farkaš
farkas@fmph.uniba.sk
Faculty of Mathematics, Physics and
Informatics, Comenius University
Bratislava, Slovakia

ABSTRACT

Artificial neural networks, in particular the deep end-to-end architectures trained by error backpropagation (BP), are currently the topmost used learning systems. However, learning in such systems is only loosely inspired by the actual neural mechanisms. Algorithms based on local activation differences were designed as a biologically plausible alternative to BP. We propose Universal Bidirectional Activation-based Learning, a novel neural model which enhances the contrastive Hebbian learning rule with special hyperparameters yielding a single learning rule that can perform multiple ways of learning, similarly to what is assumed about learning in the brain. Unlike others, our model consists of mutually dependent, yet separate weight matrices for different directions of activation propagation. We show that UBAL can learn different tasks (such as pattern retrieval, denoising, or classification) with different setups of the learning hyperparameters. We also demonstrate the performance of our algorithm on a machine learning benchmark (MNIST). The experimental results presented in this paper confirm that UBAL is comparable with a basic version BP-trained multilayer network and the related biologically-motivated models.

CCS CONCEPTS

• **Computing methodologies → Neural networks**.

KEYWORDS

error-driven learning, activation propagation, biological plausibility, local learning rule

ACM Reference Format:
Kristína Malinovská, Ľudovít Malinovský, Pavel Krsek, Svatopluk Kraus, and Igor Farkaš. 2019. UBAL: a Universal Bidirectional Activation-based Learning Rule for Neural Networks. In *2nd International Conference on Computational Intelligence and Intelligent Systems (CIIS 2019), November 23–25, 2019, Bangkok, Thailand.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

From its origin in 1980's until present boom of the deep learning architectures, the error backpropagation (BP) of Rumelhart et al. [16] has been the most prominent supervised learning algorithm for artificial neural networks (ANN). The cost paid for efficiency of the BP is it's biological implausibility [3]. Unlike the BP, learning in the brain works locally, based on the firing of the pre- and postsynaptic neurons and the upstream or downstream weights bare no knowledge of the other connections in the network. In the end-to-end BP-trained architectures, the information from the output is transferred backwards on the predefined "route" so each weight has some information about all preceding connections. This is called the weight transport problem [6]. Furthermore, activation in biological neural networks never propagates fully bidirectionally, i.e. through exactly the same connection weights. A more biologically plausible neural model should be based on both local activation differences and distinct connectivity for forward and backward activation propagation.

1.1 Error backpropagation alternatives

In 1996 O'Reilly proposed the Generalized Recirculation (GeneRec) as a biologically plausible alternative to error back-propagation. GeneRec has its roots in the alternative work of Hinton published only shortly after the BP, namely the recirculation algorithm for auto-encoders [8] and the Contrastive Hebbian learning used in restricted Boltzmann machines [7]. The recirculation has been shown to approximate the gradient descent in particular conditions [8]. It is also possible to realize the error backpropagation via contrastive Hebbian learning [18].

CIIS 2019, November 23–25, 2019, Bangkok, Thailand
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-7259-6-9/19/11…$15.00
https://doi.org/10.1145/1122445.1122456

In the standard BP algorithm, the network is firstly given an input in the form of activation of neurons on the input layer which propagates through the network and produces an estimate on the output layer (forward pass). Subsequently, the error – the difference between the desired and estimated values on the output layer – is propagated through the weights of the network in the backward direction (backward pass). In recirculation, CHL and GeneRec, instead of propagating the error value, the desired value is "clamped" on the output layer and the activation is propagated in the backward direction. The weights are then updated locally, based on the difference between the forward (minus phase) and the backward activation pass (plus phase). The CHL learning rule can be expressed as:

$$\Delta \boldsymbol{W}_{pq} = \lambda \, (\boldsymbol{p}^{+}\boldsymbol{q}^{+} - \boldsymbol{p}^{-}\boldsymbol{q}^{-}) \tag{1}$$

and the GeneRec learning rule as:

$$\Delta \boldsymbol{W}_{pq} = \lambda \, \boldsymbol{p}^{-}(\boldsymbol{q}^{+} - \boldsymbol{q}^{-}), \tag{2}$$

where $\boldsymbol{p}$ denotes the presynaptic and $\boldsymbol{q}$ the postsynaptic layer activation, $\boldsymbol{W}_{pq}$ is the matrix of synaptic weights and λ is the learning rate.

The topic of biologically plausible learning is gaining popularity in the current works, especially in prospective connection to Deep Learning. Bengio and colleagues [1] proposed a variational EM algorithm which can be interpreted as the training of a denoising autoencoder without the need of propagating error derivatives. Scellier and Bengio [17] proposed the Equilibrium Propagation (EP) algorithm, which works based on the differences between activation phases, but additionally also implements the so-called "weak clamping". Lee and colleagues [10] proposed the difference target propagation (DTP) to improve the original autoencoders and reached the performance of BP in deep networks. The feedback alignment (FA) model of Lillicrap [11] demonstrates that there is no need for symmetric weight connections for propagation of errors. It is a δ-rule similar to BP but, instead of the $W^{\intercal}$, it uses random feedback weights B to deliver teaching signals. Inspired by the FA model and the predictive coding theory, Ororbia and Mali [14] introduced Local Representation Alignment (LRA-E) that outperforms its predecessors, and similar models, including the EP and the standard BP.

2 OUR MODEL

In continuation of our previous work [4, 5, 12] we present UBAL, a Universal Bidirectional Activation-based Learning algorithm, which can be used for different kinds of tasks and and together with its predecessors aims to contribute to search for more biologically plausible alternatives in neural network learning.

2.1 Activation propagation in UBAL

Our UBAL model was mainly inspired by GeneRec [13], but also by its predecessor, the recirculation [8]. UBAL and its predecessors [4, 5] were firstly intended as a biologically plausible heteroassociative model to connect highlevel sparse neural representations in a cognitive robotic system such as a robotic mirror neuron system for a humanoid robot [15]. Unlike the GeneRec model, UBAL forms completely bidirectional associations between the pairs of data patterns.

Since UBAL is primarily designed for heteroassociative mapping, i.e. the inputs on the visible layers are targets for each other, we will call the visible layers of the network x and y, rather than the input and the output layer. Due to this heteroassociative nature, we distinguish the forward (F) direction of association (x–to–y), and the backward (B) direction (y–to–x association). Unlike BP or GeneRec, UBAL addresses the weight transport problem by using separate weight matrices for forward and backward activation propagation. In the forward direction, we label the weights W and in the backward direction we label the weights M. In each direction we define the *prediction* phase (P), in which activation propagates from one visible layer to another via the hidden layer (or layers). Additionally, after each prediction phase, the activation on the postsynaptic layer is also propagated back to the presynaptic layer. We call this bounce-back activation the *echo* phase (E). This echo is somewhat similar to the regression mechanism in the recirculation algorithm [8].

The UBAL model as proposed here is generalized for variable amount of layers. However, in the work presented here, we always use a three-layer architecture as displayed in Fig. 1 which also illustrates how activation flows in the network in different phases. Table 1 displays the general activation propagation rule between two connected layers p and q, where f is the activation function and b and d are trainable biases with constant input 1.0. This propagation rule is the same for all connected layers, so the p and the q can be thought of as both x and h, as well as h and y.

Table 1: Activation propagation in UBAL.

	Direction and phase	Label	Activation
1.	Forward Prediction	FP	$\boldsymbol{q}^{\mathrm{FP}} = f(\boldsymbol{W}_{pq}\boldsymbol{p}^{\mathrm{FP}} + \boldsymbol{b}_p)$
2.	Forward Echo	FE	$\boldsymbol{p}^{\mathrm{FE}} = f(\boldsymbol{M}_{qp}\boldsymbol{q}^{\mathrm{FP}} + \boldsymbol{d}_q)$
3.	Backward Prediction	BP	$\boldsymbol{p}^{\mathrm{BP}} = f(\boldsymbol{M}_{qp}\boldsymbol{q}^{\mathrm{BP}} + \boldsymbol{d}_q)$
4.	Backward Echo	BE	$\boldsymbol{q}^{\mathrm{BE}} = f(\boldsymbol{W}_{pq}\boldsymbol{p}^{\mathrm{BP}} + \boldsymbol{b}_p)$

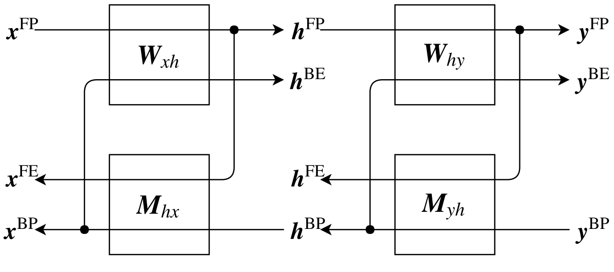

Figure 1: Activation propagation in 3-layer UBAL.

2.2 The learning rule

The UBAL rule is local and fundamentally it works in the same way as the CHL. The basis for learning is the difference between the activation values in the two direction of association (F and B) and the activation phases as defined in Table 1. The weight update is proportional to the input term multiplied by error term, which usually stands for the difference between the propagated clamped activation and the network's estimate.

Table 2: Learning rule terms.

Term	Symbol	Value
Forward Target	$\boldsymbol{t}_q^{\mathrm{F}}$	$\beta_q^{\mathrm{F}}\boldsymbol{q}^{\mathrm{FP}} + (1-\beta_q^{\mathrm{F}})\boldsymbol{q}^{\mathrm{BP}}$
Forward Estimate	$\boldsymbol{e}_q^{\mathrm{F}}$	$\gamma_q^{\mathrm{F}}\boldsymbol{q}^{\mathrm{FP}} + (1-\gamma_q^{\mathrm{F}})\boldsymbol{q}^{\mathrm{BE}}$
Backward Target	$\boldsymbol{t}_p^{\mathrm{B}}$	$\beta_p^{\mathrm{B}}\boldsymbol{p}^{\mathrm{BP}} + (1-\beta_p^{\mathrm{B}})\boldsymbol{p}^{\mathrm{FP}}$
Backward Estimate	$\boldsymbol{e}_p^{\mathrm{B}}$	$\gamma_p^{\mathrm{B}}\boldsymbol{p}^{\mathrm{BP}} + (1-\gamma_p^{\mathrm{B}})\boldsymbol{p}^{\mathrm{FE}}$

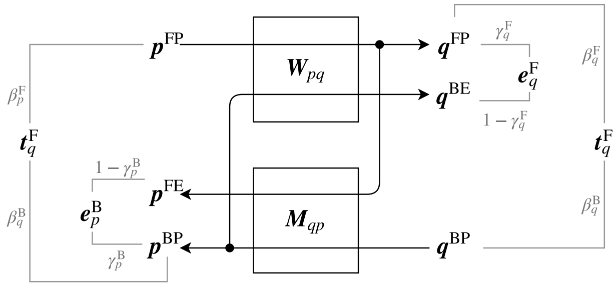
$\boldsymbol{p}^{\mathrm{FP}}$
$\boldsymbol{W}_{pq}$
$\boldsymbol{q}^{\mathrm{FP}}$
γ_q^{F}
β_q^{F}
β_p^{F}
$\boldsymbol{e}_q^{\mathrm{F}}$
$\boldsymbol{q}^{\mathrm{BE}}$
$1-\gamma_q^{\mathrm{F}}$
$\boldsymbol{t}_q^{\mathrm{F}}$
$\boldsymbol{t}_q^{\mathrm{F}}$
$1-\gamma_p^{\mathrm{B}}$
$\boldsymbol{p}^{\mathrm{FE}}$
β_q^{B}
$\boldsymbol{e}_p^{\mathrm{B}}$
$\boldsymbol{M}_{qp}$
β_q^{B}
$\boldsymbol{p}^{\mathrm{BP}}$
$\boldsymbol{q}^{\mathrm{BP}}$
γ_p^{B}

Figure 2: Activation propagation and learning rule terms.

UBAL takes into account not only the prediction and the training signal, but also the echo states that are internal to the network. In Table 2 we introduce the intermediate learning rule terms $\boldsymbol{t}$ and $\boldsymbol{e}$ and graphically depict them together with the activation states in the network in Fig. 2. The so-called *targets* (t) are proportional mixtures of activations in the prediction phases from the two directions (F and B). Because our model is essentially heteroassociative, the inputs in one direction are the targets for the other direction and vice versa. What would in other models be called the prediction, we will refer to as the *estimate*. The estimate terms (e) combine UBALs predictions with the echoes in a proportional manner.

The weights between two layers in the forward direction in our learning algorithm are updated according to

$$\Delta\boldsymbol{W}_{pq} = \lambda\, \boldsymbol{t}_p^{\mathrm{B}}(\boldsymbol{t}_q^{\mathrm{F}} - \boldsymbol{e}_q^{\mathrm{F}}) \tag{3}$$

and in backward direction according to:

$$\Delta\boldsymbol{M}_{qp} = \lambda\, \boldsymbol{t}_q^{\mathrm{F}}(\boldsymbol{t}_p^{\mathrm{B}} - \boldsymbol{e}_p^{\mathrm{B}}) \tag{4}$$

2.3 Learning rule hyperparameters

The local behavior on the respective layers of the network can vary based on the values of hyperparameters β and γ in the learning rule terms (Table 2). The β hyperparameter regulates the proportion of activation from the forward and the backward pass entering the weight update formula as target activation.

In UBAL, the hidden layer by desing converges to the same target activation pattern for both directions, creating a common representation of the input-output pair, which basically applies also to visible layers. Since for each layer l, $\boldsymbol{t}_l^{\mathrm{F}} = \boldsymbol{t}_l^{\mathrm{B}}$, we can derive that $\beta_l^{\mathrm{B}} = (1-\beta_l^{\mathrm{F}})$ for all l. This allows us to reduce the parameter space and use β^{F}s to express also β^{B}s.

The second learning rule hyperparameter is γ. In the UBAL rule, γ^F and γ^B determine how the estimate term is created by mixing prediction and echo activations. More precisely, these parameters regulate the strength of prediction in the given direction compared to the echo from the other direction.

Unlike β, the γ parameters cannot be reduced to just one direction. Rather than being related to a layer of neurons like β, γ influences the evolution of the weight matrix connecting the two layers. In other words, γ_q^{F} of weight matrix $\boldsymbol{W}_{pq}$ determines whether $\boldsymbol{W}_{pq}$ learns to predict the target values $t_q^{\mathrm{F}} = t_q^{\mathrm{B}}$ from the prediction $\boldsymbol{p}^{\mathrm{FP}}$ (yielding $\boldsymbol{q}^{\mathrm{FP}}$), or rather echo the value from the prediction in the backward direction $\boldsymbol{q}^{\mathrm{BP}}$ (yielding $\boldsymbol{q}^{\mathrm{BE}}$ from $\boldsymbol{p}^{\mathrm{BP}}$).

When using extreme values $(0, 1)$ of β and γ, the components of our learning rule reduce to different forms of learning that resemble canonical algorithms. Thus, our model can be shown to perform different kinds of learning (supervised, unsupervised, semi-supervised). The different setups of β and γ parameters for qualitatively different tasks found via extensive experimentation are shown in the next section.

3 EXPERIMENTS AND RESULTS

We evaluated UBAL on several canonical tasks and on a traditional classification benchmark of handwritten digit recognition. We compare our approach with related models and with standard error backpropagation. In our experiments reported in this paper we use a fully connected three layer network with standard sigmoidal units.

3.1 Tested problems

The 4–2–4 encoder is a very simple task of encoding 4 simple binary patterns through a narrow bottleneck of 2 hidden neurons, originally proposed for testing auto-associative models [8]. The inputs-outputs are: [0001, 0010, 0100, 1000].

XOR is a canonical classification task which is simple, yet linearly non-separable. For consistency, the inputs to the network are [00, 01, 10, 11] mapped to just one binary output [0, 1]. In order to compare UBAL with related models in 4–2–4 encoder and XOR tasks, we will evaluate our model based on the network convergence, i.e. the percentage of tested networks that found a stable solution with 100% accuracy for a given number of consecutive epochs.

Heteroassociation. To test whether our algorithm performs at least the same or better than its predecessor BAL we experimented with random binary pattern associations. In this task we use 200 pairs of randomly generated sparse binary patterns – binary vectors with maximum of 10 positive bits per pattern. An illustration of such data is depicted in Fig. 3. In this and the following task we evaluate the model based on the average pattern retrieval success. Since we are working with binary vectors where positive bits are rather sparse we use the F_1 score instead of standard accuracy.

Denoising. To explore denoising abilities of our algorithm we adopted one subtask from [2] with binary (black and white) raster images of letters with dimension 7×7 pixels. There are 8 capital

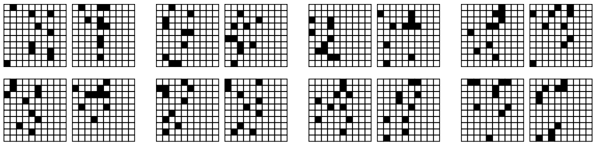

Figure 3: Examples of random sparse binary patterns.

letters from A to F corrupted by a random noise vector with Gaussian distribution with mean 0.0 and variance 0.5. We trained all networks with randomly generated distorted patterns (50 for each letter). For assessing the network generalization ability we use a separate testing set (10 instances per letter). Examples of noisy letter pairs are depicted in Fig. 4.

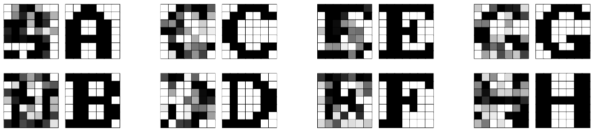

Figure 4: Examples of noisy letters.

Handwritten digits. The famous MNIST dataset [9] has been used to test the above-mentioned related models. The full MNIST dataset consists of labeled black and white images of 10 handwritten digits with resolution 28×28 pixels. The set is already split to 60,000 training and 10,000 testing samples. We use standard performance measure – the classification error on the test set. Unlike most of the other works, we do not preprocess the images or normalize the input values.

3.2 Model hyperparameters

To gather the optimal values of learning rule hyperparameters β and γ for the particular tasks we performed extensive grid search experiments. We experimented with different means and variance of initial weights as well as the learning rate λ and with the size of the hidden layer. The resulting hyperparameter values are listed in Table 3.

All synaptic weights in our experiments were initialized from Gaussian distribution with varying mean ν and variance σ^2. We explored these hyperparameters experimentally and found that ideal values vary from task to task. To illustrate this, we display the average training epochs needed until 100% convergence as a function of ν and σ^2 for Enc. 4–2–4 (Fig. 5) and XOR with 6 hidden neurons (Fig. 6). Although the space of best initial weights for these two problems configurations overlaps, ideal values for these two task differ (XOR: $\mathcal{N}(0.0, 2.5)$, Enc. 4–2–4: $\mathcal{N}(0.5, 0.5)$).

We found that the size of the hidden layer has significant influence on the performance of UBAL. We illustrate this phenomenon in Fig. 7 where the performance of UBAL in the handwritten digits task (MNIST) after 100 training epochs (average over 10 nets) is projected as the function of the hidden layer size. We used the hyperparameter values from Table 3 and with each setup trained 10 randomly initialized networks for 100 epochs.

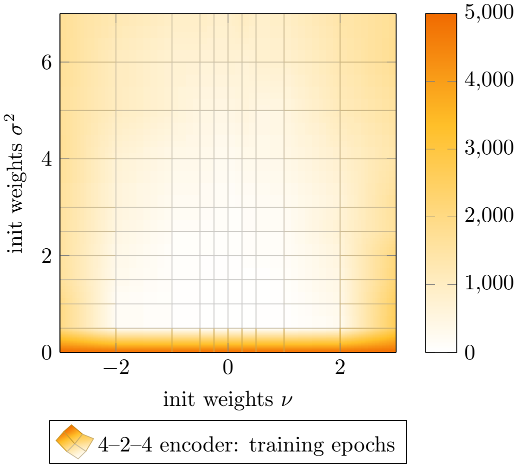

Figure 5: Initial weights experiment with 4–2–4 encoder.

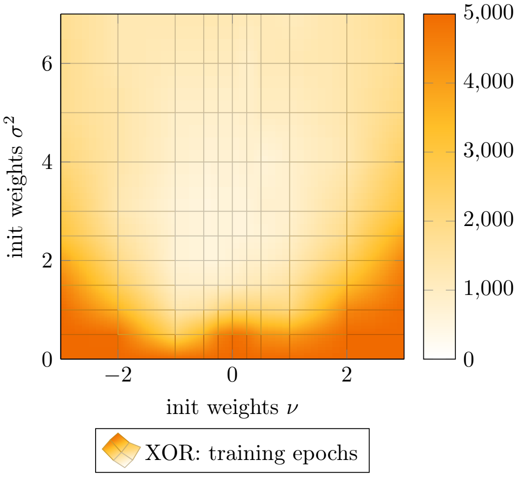

Figure 6: Initial weights experiment with 6-neuron XOR.

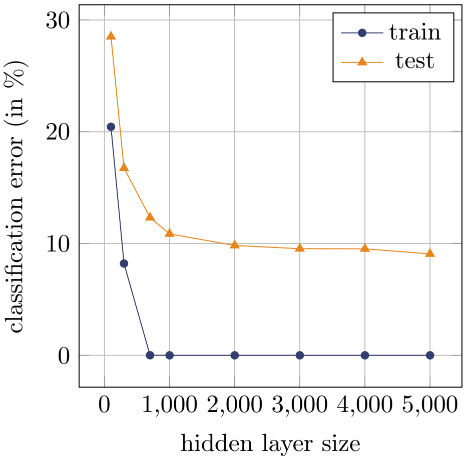

Figure 7: Hidden layer size experiment with MNIST.

Table 3: Best hyperparameter values from experiments

Parameter/Task	Enc. 4–2–4	XOR	XOR minimal	Heteroassociation	Denosing	MNIST
Architecture	4–2–4	2–6–1	2–2–1	100–100-100	49–30–49	784–1000–10
Learning rate λ	1.0	2.0	2.0	1.0	0.001	0.2
Weight initialization	$\mathcal{N}(0.5, 0.5)$	$\mathcal{N}(0.0, 2.5)$	$\mathcal{N}(0.0, 2.5)$	$\mathcal{N}(0.0, 1.0)$	$\mathcal{N}(0.0, 0.3)$	$\mathcal{N}(0.0, 0.5)$
β^F	1.0 - 0.5 - 0.0	0.0 - 1.0 - 0.0	0.25 - 0.9 - 0.0	0.2 - 0.8 - 0.0	1.0 - 0.5 - 0.0	1.0 - 1.0 - 0.9
γ^F	0.5 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.5	0.5 - 0.0	0.0 - 0.0
γ^B	0.0 - 0.5	0.0 - 0.0	0.0 - 0.0	0.0 - 0.5	0.0 - 0.5	0.1 - 0.0
β^B	0.0 - 0.5 - 1.0	1.0 - 0.0 - 1.0	0.75 - 0.1 - 1.0	0.8 - 0.2 - 1.0	0.0 - 0.5 - 1.0	0.0 - 0.0 - 0.1

3.3 Training progress and generalization

To illustrate how the network learns to generalize in the course of training, we display training and testing errors as a function of the training epoch (averaged over 10 randomly initialized networks with best parameters from Table 3). We display the classification error in the handwritten digits task in Fig. 8 for training, validation and testing datasets in 100 epochs. For validation, we took 10% of the original training set. We display also the performance on the dedicated MNIST testing set. Note, that UBAL may not suffer from overtraining, with growing success on the training set, the validation error does not show a tendency to drop. More experimentation is needed to better understand this interesting property.

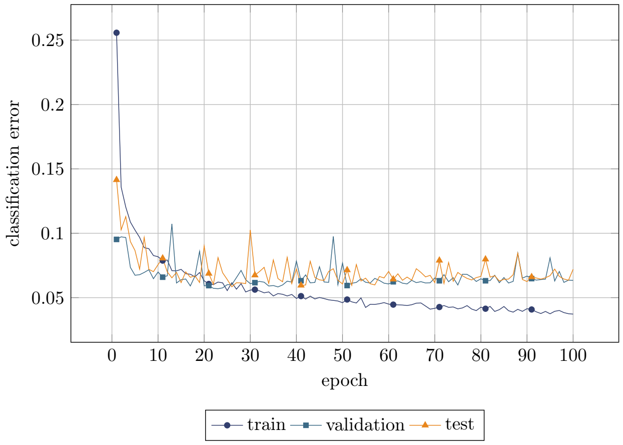

Figure 8: Training progress with MNIST.

Table 4: Comparison with other models in Enc. 4–2–4

Algorithm	λ	Success rate
Backprop [13]	2.4	100%
GeneRec [13]	0.6	90%
CHL [13]	1.2	56%
BAL [5]	0.9	65%
BAL2 [4]	0.0002/500	93.1%
UBAL	1.0	100%

Table 5: Comparison with other models in XOR 2-2-1

Algorithm	λ	Success rate
Backprop [13]	1.95	74%
GeneRec [13]	0.2	18%
CHL [13]	1.8	52%
BAL [5]/BAL2 [4]	any	0%
UBAL	2.0	64%

3.4 Comparison with related models

Comparison for 4–2–4 encoder task is in Table 4 and for XOR task in Table 5, where success rate refers to the percentage of networks that successfully converged to a solution (out of 50 runs).

For a proper comparison with the results from the GeneRec model we used the minimal architecture for XOR with just 2 hidden neurons, whose convergence is not ensured even for standard error backpropagation method. In our experiments, we observe that this also applies to our UBAL model, and that UBAL's convergence rate is slightly lower than BP, but higher then CHL and GeneRec. Compared to the minimal model, the performance rapidly increases with 3 hidden neurons and 100% of networks converge with 6 or more hidden neurons.

In the random sparse binary pattern heteroassociation task we can compare UBAL only with [5]. From our experimentation we conclude that the new UBAL performs much better than the old BAL model. In the denoising task, adopted from [2], we do not have sufficient data (and space in this paper) to make a proper comparison with the BHM+SOFM model. Our best models reach the F1 score of approximately 99.6% on the test set.

For the handwritten digit recognition task, we took some baselines from the original article by Lecunn [9], from previous work on BAL2 model [4], and from some of the related models with a single hidden layer. For comparison of UBAL with the related fully connected models in Table 6 we used a standard architecture with

Table 6: Comparison with other models for MNIST.

Algorithm	Architecture	Classification error
Linear classifier [9]	784-10	12.0%
Backprop [9]	784-1000-10	4.5%
GeneRec [4]	784-300-50-10	56.8%
BAL [4]	784-300-10	90.2%
BAL2 [4]	784-300-10	11.5%
UBAL	784-1000-10	6.0%
EP [17]	784-500-10	3.0%
FA [11]	784-1000-10	2.1%

1000 hidden neurons and the softmax activation function on the visible layer y.

3.5 Discussion and future work

Our results indicate that the performance of our model is slightly worse than that of a simple BP-trained multilayer network, but better than preceding biologically motivated models. In the case of the handwritten digits problem, it is important to note that we present results from quite a plain model and apply no image preprocessing or normalization. We believe that better performance can still be achieved after more extensive experimentation with model parameters. In the future, we plan to experiment with deep and convolutional architectures which could significantly improve the performance in visual classification tasks and allow to explore how the model would be able to perform image classification, segmentation, mapping, and denoising tasks more effectively.

Since the UBAL learning rule is local, each pair of connected layers can have a particular parameter setup tuned up for different type of learning – supervised, unsupervised, or a combination of these paradigms. We find the possibility of having one network employ multiple learning paradigms as one of the most challenging and interesting research prospects of UBAL. Such a network would be, for instance, able to denoise a pattern, restore it, retrieve missing information via associated patterns and subsequently classify the input correctly. The form of our rule also allows us to build more than just two-ended input-output architectures and generally build various complex architectures.

4 CONCLUSION

We presented UBAL, a novel learning algorithm for artificial neural networks as a more biologically plausible alternative to error backpropagation. Our model is essentially a bidirectional heteroassociative network where patterns presented to both visible layers simultaneously serve as targets to each other. Along with learning from the teaching signal the network also learns from bounced-back predictions (echoes). In each direction of activation propagation there is a separate weight matrix, therefore UBAL does not suffer from the weight transport problem. The UBAL learning rule is local, so only presynaptic and postsynaptic activations in different activation phases contribute to updating the weights. Based on learning rule hyperparameters our model can learn various qualitatively different tasks, including encoding, denoising, and classification. Our experimental results confirm that UBAL successfully solves the given problems and that its performance is comparable with the related models.

ACKNOWLEDGMENTS

This research was supported by the Ministry of the Interior of the Czech Republic under Project VI20172019082 - Smart Camera. The work of Igor Farkaš was supported by Slovak Grant Agency for Science (VEGA), project 1/0798/18.

REFERENCES

[1] Yoshua Bengio, Dong-Hyun Lee, Jorg Bornschein, Thomas Mesnard, and Zhouhan Lin. 2015. Towards biologically plausible deep learning. *arXiv preprint arXiv:1502.04156* (2015).

[2] Sylvain Chartier, Gyslain Giguère, and Dominic Langlois. 2009. A new bidirectional heteroassociative memory encompassing correlational, competitive and topological properties. *Neural Networks* 22, 5-6 (2009), 568–578.

[3] Francis Crick. 1989. The recent excitement about neural networks. *Nature* 337, 6203 (1989), 129–132.

[4] Peter Csiba and Igor Farkaš. 2015. Computational analysis of the Bidirectional Activation-based Learning in autoencoder task. In *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–6.

[5] Igor Farkaš and Kristína Rebrová. 2013. Bidirectional activation-based neural network learning algorithm. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN), Sofia, Bulgaria*.

[6] Stephen Grossberg. 1987. Competitive learning: From interactive activation to adaptive resonance. *Cognitive Science* 11, 1 (1987), 23–63.

[7] Geoffrey E Hinton. 1989. Deterministic Boltzmann learning performs steepest descent in weight-space. *Neural Computation* 1, 1 (1989), 143–150.

[8] Geoffrey E Hinton and James L McClelland. 1988. Learning representations by recirculation. In *Neural Information Processing Systems*. 358–366.

[9] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[10] Dong-Hyun Lee, Saizheng Zhang, Asja Fischer, and Yoshua Bengio. 2015. Difference target propagation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 498–515.

[11] Timothy P Lillicrap, Daniel Cownden, Douglas B Tweed, and Colin J Akerman. 2016. Random synaptic feedback weights support error backpropagation for deep learning. *Nature Communications* 7 (2016), 13276.

[12] Kristína Malinovská, Ľudovít Malinovský, and Igor Farkaš. 2018. Towards More Biologically Plausible Error-Driven Learning for Artificial Neural Networks. In *International Conference on Artificial Neural Networks*. Springer, 228–231.

[13] Randall C O'Reilly. 1996. Biologically plausible error-driven learning using local activation differences: The generalized recirculation algorithm. *Neural Computation* 8, 5 (1996), 895–938.

[14] Alexander G Ororbia and Ankur Mali. 2018. Biologically Motivated Algorithms for Propagating Local Target Representations. *arXiv preprint arXiv:1805.11703* (2018).

[15] Kristína Rebrová, Matej Pecháč, and Igor Farkaš. 2013. Towards a robotic model of the mirror neuron system. In *Proceedings of the 3rd joint IEEE International Conference on Development and Learning and on Epigenetic Robotics*. Osaka, Japan.

[16] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Number 1. The MIT Press, Cambridge, MA, 318–362.

[17] Benjamin Scellier and Yoshua Bengio. 2017. Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. *Frontiers in Computational Neuroscience* 11 (2017), 24.

[18] Xiaohui Xie and Sebastian H Seung. 2003. Equivalence of backpropagation and contrastive Hebbian learning in a layered network. *Neural Computation* 15, 2 (2003), 441–454.