Bidirectional Activation-based Neural Network Learning Algorithm

Igor Farkaš and Kristína Rebrová

Faculty of Mathematics, Physics and Informatics
Comenius University in Bratislava
{farkas,rebrova}@fmph.uniba.sk

Abstract. We present a model of a bidirectional three-layer neural network with sigmoidal units, which can be trained to learn arbitrary mappings. We introduce a bidirectional activation-based learning algorithm (BAL), inspired by O'Reilly's supervised Generalized Recirculation (GeneRec) algorithm that has been designed as a biologically plausible alternative to standard error backpropagation. BAL shares several features with GeneRec, but differs from it by being completely bidirectional regarding the activation propagation and the weight updates. In pilot experiments, we test the learning properties of BAL using three artificial data sets with binary patterns of increasing complexity.

Keywords: bidirectional network, error-driven learning, activation states, binary codes association, biological plausibility.

1 Introduction

The standard error backpropagation learning algorithm [8] is known to be biologically implausible because it requires the mechanism of error propagation and it does not use locally available, activation-based variables. With this in mind, O'Reilly [4] designed Generalized Recirculation (GeneRec) algorithm that avoids the computation of error derivatives, yet can lead to error minimization. GeneRec was designed as an extension of Hinton and McClelland's model based on recirculation [2] between two layers of units (visible and hidden) with symmetric weights, which was restricted to autoassociation. To make it work, they used a four-stage activation update process. Unlike the recirculation algorithm, GeneRec is applied to a three-layer network using bidirectional interaction (only) between two layers of units (hidden and output) in a two-phase activation update process. In his work, O'Reilly experimented with several modifications of GeneRec, determined by weight update rules. For instance, he showed that the symmetry-preserving version of GeneRec (i.e. with symmetric hidden-to-output weights and symmetric weight update), combined with the so-called midpoint method, is equivalent to Contrastive Hebbian learning (CHL) for training Boltzmann machines (both in stochastic and deterministic versions) [4]. Both GeneRec and CHL are based on differences between two activation phases. Forward (minus) phase involves activation propagation from inputs toward outputs producing the network estimate of the output values. Subsequent backward (plus) phase

V. Mladenov et al. (Eds.): ICANN 2013, LNCS 8131, pp. 154–161, 2013.

flows in the opposite direction propagating the desired output throughout the network (see Sec. 2). We propose a bidirectional activation-based learning (BAL), which is based on the GeneRec model, but is completely symmetrical regarding the activation propagation and the weight update rules (Sec. 3). Our motivation for designing the BAL model was to implement it in our robotic mirror neuron system model, that is assumed to require the bidirectional mapping between high-level sensory and motor representations [7]. The behavior of BAL is tested in three preliminary experiments (Sec. 4).

2 GeneRec Model

The GeneRec model is a three-layer network with full connectivity between layers whose activation rules are described in Table 1, following [4]. The model has reciprocal connectivity between hidden and output layer with symmetric weights. The activation flow starts in minus phase, when the stimulus s_i is presented. Note that the net input term at the hidden layer includes the input from both visible layers before applying the sigmoid activation function $\sigma(\eta) = 1/(1 + \exp(-\eta))$. Output units produce activations o_k^- in minus phase but can also be clamped to target activations o_k^+ at the onset of plus phase. Input units can only deliver stimuli s_i at the onset of minus phase. This model was developed in the Leabra framework [3], which uses dynamic units approximating the behavior of biological neurons. O'Reilly [4] has shown that, under certain conditions, GeneRec computes the same error derivatives as Almeida-Pineda recurrent backpropagation [1,6].

Table 1. Equilibrium network variables in GeneRec model

Layer	Phase	Net Input	Activation
Input (s)	−	-	s_i = stimulus input
Hidden (h)	−	$\eta_j^- = \sum_i w_{ij} s_i + \sum_k w_{kj} o_k^-$	$h_j^- = \sigma(\eta_j^-)$
	+	$\eta_j^+ = \sum_i w_{ij} s_i + \sum_k w_{kj} o_k^+$	$h_j^+ = \sigma(\eta_j^+)$
Output (o)	−	$\eta_k^- = \sum_j w_{jk} h_j$	$o_k^- = \sigma(\eta_k^-)$
	+	-	o_k^+ = target output

The basic weight update rule in GeneRec is:

$$\Delta w_{pq} = \lambda \; a_p^-(a_q^+ - a_q^-) \tag{1}$$

where a_p^- denotes the presynaptic and a_q^- denotes the postsynaptic unit activation in minus phase, a_p^+ is the presynaptic activation from plus phase (in output-to-hidden direction) and λ denotes the learning rate. The learning rule given in Eq. 1 is applied to both input-hidden and hidden-output weights. Due to the lack of space, the reader is left to consult the original paper [4] regarding the underlying math behind the derivation of the GeneRec learning rule.

3 Bidirectional Activation-based Learning

Bidirectional Activation-based Learning algorithm (BAL) shares with GeneRec the phase-based activations and unit types, but differs from it by the connectivity that allows completely bidirectional associations to be established (GeneRec focuses on input-to-output mapping). Unlike GeneRec, BAL uses two pairs of weight matrices for each activation phase. In addition, in BAL we do not use dynamical settling process but compute the activations in one step as described in Table 2.

Table 2. Activation phases and states in BAL model

Layer	Phase	Net Input	Activation
$\mathbf{x}$	F	-	$x_i^{\mathrm{F}} = \text{stimulus}$
$\mathbf{h}$	F	$\eta_j^{\mathrm{F}} = \sum_i w_{ij}^{IH} x_i^F$	$h_j^{\mathrm{F}} = \sigma(\eta_j^{\mathrm{F}})$
$\mathbf{y}$	F	$\eta_k^{\mathrm{F}} = \sum_j w_{jk}^{HO} h_j^F$	$y_k^{\mathrm{F}} = \sigma(\eta_k^{\mathrm{F}})$
$\mathbf{y}$	B	-	$y_k^{\mathrm{B}} = \text{stimulus}$
$\mathbf{h}$	B	$\eta_j^{\mathrm{B}} = \sum_k w_{kj}^{OH} y_k^{\mathrm{B}}$	$h_j^{\mathrm{B}} = \sigma(\eta_j^{\mathrm{B}})$
$\mathbf{x}$	B	$\eta_i^{\mathrm{B}} = \sum_j w_{ji}^{HI} h_j^{\mathrm{B}}$	$x_i^{\mathrm{B}} = \sigma(\eta_i^{\mathrm{B}})$

We avoid input-output notation of layers as used in GeneRec, because in our case not only output can be evoked by input presentation, but also vice versa. Hence, we label the two outer (visible) layers $\mathbf{x}$ and $\mathbf{y}$ and the hidden layer $\mathbf{h}$. Let forward activation be denoted by subscript F, backward activation denoted by subscript B. Then during the forward pass, the $\mathbf{x}$ units are clamped to $\mathbf{x}^{\mathrm{F}}$ and we get the activations $\mathbf{x}^{\mathrm{F}} \rightarrow \mathbf{h}^{\mathrm{F}} \rightarrow \mathbf{y}^{\mathrm{F}}$. During the backward pass, the $\mathbf{y}$ units are clamped to $\mathbf{y}^{\mathrm{B}}$ and we get the activations $\mathbf{y}^{\mathrm{B}} \rightarrow \mathbf{h}^{\mathrm{B}} \rightarrow \mathbf{x}^{\mathrm{B}}$.

The mechanism of weights update partially matches that of GeneRec. Each weight in BAL network (i.e. belonging to one of the four weight matrices) is updated using the same learning mechanism, according to which the weight difference is proportional to the product of the presynaptic (sending) unit activation a_p and the difference of postsynaptic (receiving) unit activations a_q, corresponding to two activation phases (F and B, in particular order). Namely, weights in $\mathbf{x}$-to-$\mathbf{y}$ direction (belonging to $\mathbf{h}$ and $\mathbf{y}$ units) are updated as

$$\Delta w_{pq}^{\mathrm{F}} = \lambda \; a_p^{\mathrm{F}} (a_q^{\mathrm{B}} - a_q^{\mathrm{F}}), \tag{2}$$

where, as in the GeneRec algorithm, a_p^{F} denotes the presynaptic activity, a_q^{F} is the postsynaptic activity, and a_q^{B} denotes the postsynaptic activity from the opposite phase ($\mathbf{y}$-to-$\mathbf{h}$). Analogically, the weights in $\mathbf{y}$-to-$\mathbf{x}$ direction (belonging to $\mathbf{h}$ and $\mathbf{x}$ units) are updated as

$$\Delta w_{pq}^{\mathrm{B}} = \lambda \; a_p^{\mathrm{B}} (a_q^{\mathrm{F}} - a_q^{\mathrm{B}}) \tag{3}$$

All units have trainable thresholds (biases) that are updated in the same way as regular weights (being fed with a constant input 1).

4 Experiments

We test the learning properties of BAL on three experiments with artificial binary data that differ in dimensionality, size of the training set and mapping complexity. We chose binary data because they simplify the assessment of network accuracy and they have properties of discrete sparse patterns (used in our intended application). For assessing the network performance, we used three quantitative measures (separately for F and B directions): (1) pattern success (patSucc), which indicates the proportion of output patterns that completely match targets, (2) bit success (bitSucc), the proportion of units matching their target, and (3) mean squared error (MSE) per neuron. Based on earlier experiments, we initialize the weights in all tests to small values from the normal distribution $\mathcal{N}(0; 1/\sqrt{n_I + 1})$, where n_I denotes the input data dimension.

4.1 4-2-4 Encoder

To compare the performance of BAL with GeneRec, we ran tests using the well-known 4-2-4 encoder task, following O'Reilly [4]. We investigated the convergence of BAL and the number of required training epochs as a function of the learning rate. Fig. 1 shows the convergence success for 100 networks and the average numbers of epochs needed. The simulations showed that convergence of BAL depends on the learning rate, with the highest number of 65% successful runs achieved for $\lambda = 0.9$. For comparison, O'Reilly [4] reports 90% success for basic GeneRec algorithm and 56% for a symmetric modification of GeneRec and its modification equivalent to CHL. In sum, probability of BAL convergence is lower than that of basic GeneRec rule, but comparable to its symmetric versions. We expect that the smaller number of successful runs is in both cases influenced by the bidirectional nature of the weight update.

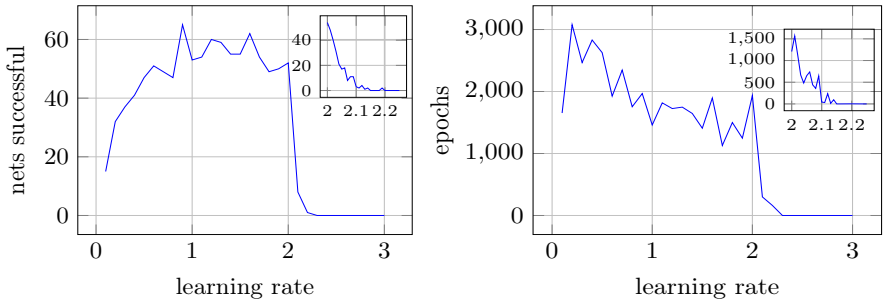

Fig. 1. 4-2-4 encoder: results for 100 nets, number of successful runs (left), average number of training epochs needed for convergence (right), both as a function of λ. Details for critical values are shown in inset plots.

BAL was observed to require a higher number of training epochs than GeneRec, with very high variability (and skewed distribution), ranging from 100

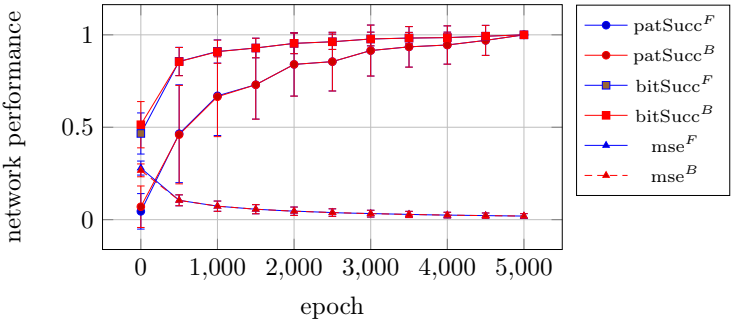

Fig. 2. Encoder 4-2-4: Development of network convergence (50 successful nets).

to thousands of epochs. On the contrary, O'Reilly reports only 418 epochs for GeneRec to converge, and less than 100 epochs for symmetric versions of GeneRec. An interesting property of BAL is that convergence probability sharply drops to zero beyond certain range of values of the learning rate, for 4-2-4 task at $\lambda = 2$. BAL convergence in 4-2-4 task and sensitivity to learning rate deserves further investigation. Fig. 2 illustrates the learning process of 50 successful networks during 5000 epochs using $\lambda = 0.9$. We conclude that MSE drops to minimum values satisfying error-free performance of the network as indicated by all success-based measures (converging to one) in both directions. If the network converges, it masters the encoder task perfectly.

4.2 Simple Binary Vector Associations

We created a sparse binary data set with high dimensionality, which resembles sensory-motor patterns from our related work. The motivation for using sparse representations comes from considering the mapping between two domains that could lend itself to generalization and robustness. In biological networks, such representations are typically achieved by lateral inhibition. As a computational shortcut, we use the k-WTA (winner-takes-all) mechanism [5]. This mechanism sets k maximally responding units to one and resets the remaining units to zero. In our related work, we apply this mechanism to output maps from two lower level modules consisting of self-organizing maps, which are then associated using the BAL algorithm. Data used in this experiment are 144-dimensional binary vectors with $k = 12$ active units. Two sets of 100 vectors are arbitrarily associated to form one-to-one mapping. Similarly to the previous experiment, we tested the network performance with various values of learning rate using 144–120–144 architecture. Fig. 3 displays the results. The network learns the mapping well up to a certain value of the learning rate ($\lambda = 0.3$), beyond which it is again observed to quickly deteriorate (Fig. 3 left). Subsequently, using the estimated optimal learning rate ($\lambda = 0.2$), we also tested selected sizes of the hidden layer n_H (Fig. 3 right). We can conclude that n_H has significant influence only on the amount of training epochs needed to reach 100% success (inset figure).

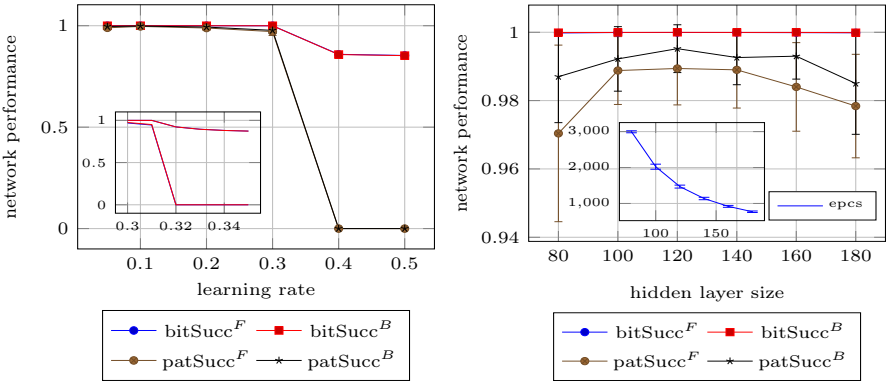

Fig. 3. Bidirectional associator: network performance as a function of λ (on the left, detail for critical values in the inset plot) and n_H (on the right, with the number of epochs needed in the inset plot).

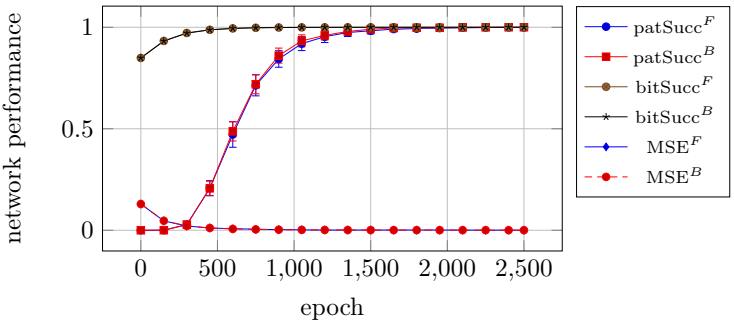

Fig. 4. Bidirectional associator: development of network performance over time (50 nets). All nets reliably converge after 1500 epochs.

To demonstrate the network training process, we computed performance measures for 50 nets trained for 2500 epochs using optimized parameters $\lambda = 0.2$ and $n_H = 120$. Results in Fig. 4 show that the networks reliably converge to successful mappings between sparse patterns. To understand the network behavior we also examined the hidden layer. We observed that $\mathbf{h}^{\mathrm{F}}$ and $\mathbf{h}^{\mathrm{B}}$ activations have a tendency to move closer to each other, as could be expected from BAL (and also from GeneRec) learning rule. Interestingly, activations of $\mathbf{h}$ units in both directions converged roughly to 0.5, so no tendency towards binary internal representations was observed. This property of internal coding is also worth further investigation.

4.3 Complex Binary Vector Associations

We evaluated the network performance on n–to–1 data associations, motivated by the sensory-motor mappings between distributed patterns. For this purpose

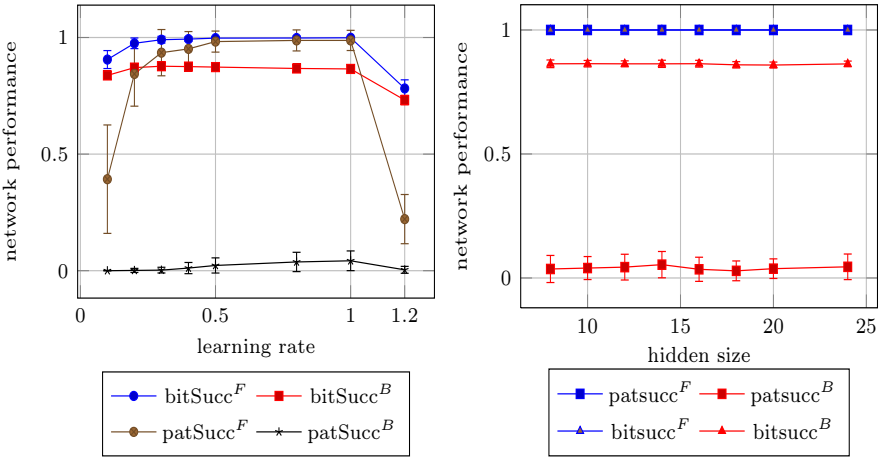

Fig. 5. Bidirectional associator with complex data: network performance as a function of λ (left) and n_H (right).

we created low-dimensional sparse binary codes, 16-dimensional vectors (4×4 map) with $k = 3$ active units with $n = 4$. For each target ($\mathbf{y}$), these four patterns ($\mathbf{x}$) were assumed to have nonzero overlap. Again, we searched for optimal λ and n_H (Fig. 5). The best performance was achieved using $\lambda \approx 1$. We can observe that the ambiguity in the data association causes the network to produce errors in B direction. For the best λ the networks yielded patSuccB $\approx$ 4% and bitSuccB $\approx$ 86%, which means that the networks made small errors in most patterns. This could be expected since the network cannot know which of the four ($\mathbf{x}$) patterns is to be reconstructed. It is known, that a network trained to associate more binary target patterns with one pattern tends to produce a mesh of outputs, weighed by their frequency of occurrence in the training set. Examples of network outputs are illustrated in Fig. 6.

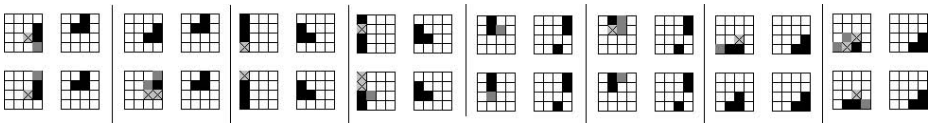

Fig. 6. Bidirectional associator with complex data: active units are filled with color, black = target–estimate match, gray = target only, gray with a cross = false-positive estimate.

5 Conclusion

We presented a new training algorithm BAL for bidirectional mappings derived from biologically plausible GeneRec model. Unlike the original GeneRec model,

our model is used with standard multi-layer perceptron network without activation settling dynamics, and it learns a bidirectional mapping rather than input-output mapping. BAL also differs from the original model in the design of weight matrices, the learning rule, and partially also in the activation flow in the two activation phases. Our preliminary experiments have shown that using an appropriate learning rate, the BAL model can converge, albeit requiring more training epochs than GeneRec. In particular, for 4-2-4 encoder task the convergence is not guaranteed, which was observed also in the GeneRec model. The next step in our research should be to investigate the reasons for these performance discrepancies. Our experiments also revealed that hidden unit activations tend to converge to similar values for F and B phases. They do not tend to binarize, which is probably not necessary for learning the task. Further experiments and a more detailed analysis of BAL are required to better understand this biologically motivated bidirectional learning algorithm that will be exploited in our robotic mirror neuron system model.

Acknowledgments. This work was supported by grants 1/0439/11 from Slovak Grant Agency for Science (VEGA), and Comenius University UK/631/2013 (K.R.).

References

1. Almeida, L.: A learning rule for asynchronous perceptrons with feedback in a combinatorial environment. In: Proceedings of the IEEE First International Conference on Neural Networks, San Diego, CA (1987)
2. Hinton, G.E., McClelland, J.L.: Learning representations by recirculation. In: Neural Information Processing Systems, pp. 358–366. American Institute of Physics, New York (1988)
3. O'Reilly, R.: The Leabra model of neural interactions and learning in the neocortex. Ph.D. thesis, Carnegie Mellon University (1996)
4. O'Reilly, R.: Biologically plausible error-driven learning using local activation differences: The generalized recirculation algorithm. Neural Computation 8(5), 895–938 (1996)
5. O'Reilly, R., Munakata, Y., Frank, M., Hazy, T.: Computational Cognitive Neuroscience, 2nd edn. Wiki Book (2012)
6. Pineda, F.: Generalization of back-propagation to recurrent neural networks. Physics Review Letters 59, 2229–2232 (1987)
7. Rebrová, K., Pecháč, M., Farkaš, I.: Towards a robotic model of the mirror neuron system. In: The 3rd Joint IEEE International Conference on Development and Learning and on Epigenetic Robotics (2013)
8. Rumelhart, D., Hinton, G., Williams, R.: Learning internal representations by error propagation, vol. 1, pp. 318–362. MIT Press, Cambridge (1986)