Intrinsic gradient networks:
Highly recurrent neural networks with biologically plausible training

Jason Tyler Rolfe[1], Matthew Cook[2], and Yann LeCun[1]
[1] The Courant Institute of Mathematical Sciences - New York University
719 Broadway, 12th Floor, New York, NY 10003
[2] Institute of Neuroinformatics - ETH Zurich
Winterthurerstrasse 190, CH-8057 Zurich, Switzerland
rolfe@cs.nyu.edu, cook@ini.phys.ethz.ch, yann@cs.nyu.edu

Artificial neural networks are computationally powerful and exhibit brain-like dynamics. However, it is generally believed that the backpropagation algorithm, commonly used to train neural networks, is not biologically plausible [1]; backpropagation messages must not directly affect the original feedforward messages, in contradiction to the pervasively recurrent architecture of the cortex [2].

More generally, the set of potential algorithms that the cortex might use for sensory processing, motor planning, and learning is constrained by the structural properties of cortical architecture and dynamics. The recurrence of the cortex, together with the requirement that training signals project directly to the trained structures in the brain, implies that the cortex uses a single interdependent set of messages for both computation and learning. Moreover, when faced with complicated or ambiguous input, the cortex can withhold a motor response until processing is complete. We further assume that learning requires the approximate calculation of the gradient of a loss function, and restrict our attention to networks in which this gradient can be calculated completely at a single network state.

From this modest set of constraints, we derive a novel class of recurrent neural networks, *intrinsic gradient networks*, for which the gradient of the loss function with respect to the parameters is a simple function of the network state when a self-identified output has been produced. Intrinsic gradient networks do not generally segregate "feedforward" computation signals from "feedback" training signals, and so are potentially consistent with the pervasive recurrence observed in the cortex. Within the class of intrinsic gradient networks, it is easy to identify highly recurrent instances for which training is simple and local, thus satisfying all of the biological and computational constraints identified above.

The set of intrinsic gradient networks is large and diverse; an example will demonstrate the power of this technique. We can construct a hierarchical sparse coding intrinsic gradient network with consecutive layers of non-negative real-valued units $\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_n$ connected by parameter matrices $\mathbf{W}_i$. The dynamics minimize[1] the potential function[2]

$$V(\vec{x}, \mathbf{W}) = \lambda \cdot (||\mathbf{W}_1 \cdot \vec{x}_1||_2 - 1) + \sum_i \frac{1}{2} \cdot ||\vec{x}_{i-1} - \mathbf{W}_i \cdot \vec{x}_i||_2^2 + \alpha \cdot ||\vec{x}_i||_1$$

with respect to $\vec{x}$ and maximize it with respect to the Lagrange multiplier λ, where $\vec{x}_0$ is the input to the sparse coder. The gradient of the loss function $L(\vec{x}, \mathbf{W}) = \frac{1}{2} \cdot ||\vec{x}_0 - \mathbf{W}_1 \cdot \vec{x}_1||_2^2 + \alpha \cdot \sum_i ||\vec{x}_i||_1 - \lambda$ at a stationary point of $V(\vec{x}, \mathbf{W})$ is simply

$$\frac{dL}{d\mathbf{W}_i} = -2 \cdot \left[\vec{x}_{i-1} - \left(1 + \delta_{i,1} \cdot \frac{\lambda}{||\mathbf{W}_i \cdot \vec{x}_i||_2}\right) \cdot \mathbf{W}_i \cdot \vec{x}_i\right] \cdot \vec{x}_i^\top \, , \tag{1}$$

where δ is the Kronecker delta. This network differs from traditional hierarchical sparse coding networks [e.g., 3, 4] primarily in that the magnitude of the output is fixed by a Lagrange multiplier. As a result of this small change, the total derivative of the loss function[3] (*not* the partial derivative, which is zero for all layers after the first) can be calculated by equation 1, a simple function of the adjacent layers. Unlike probabilistic deep networks [e.g., 5], the gradient is calculated exactly after a single run to convergence. To the extent that equation 1 can be computed using only local signals, this network satisfies all of our biological and computational desiderata. When trained on the MNIST database, a two-hidden-layer network with this structure learns small pen strokes as first-level features, and agglomerations of strokes that differentiate between the digit classes as second-level features, as shown in figure 1. Such a network can classify digits with no more than 2.2% error, and performance continues to improve as we refine our network and training methodology.

Topic: Learning algorithms
Preference: Oral
Presenter: Jason Tyler Rolfe

[1] Minimization is subject to the non-negativity constraint.

[2] The quantity we refer to as the potential function is sometimes called the energy function.

[3] The loss function only depends on the reconstruction of the input by the first hidden layer, the sparsity of each layer, and the Lagrange multiplier. It is not the same as the potential function, which depends on the reconstruction of every hidden layer.

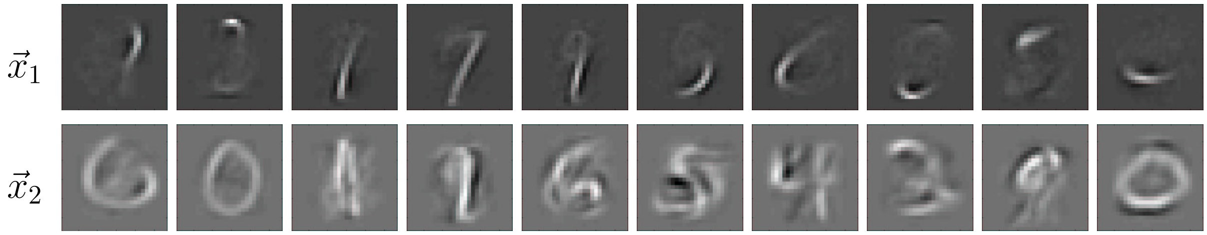

Figure 1: Projective fields from a two-hidden layer hierarchical sparse coding intrinsic gradient network, with 400 units per layer, trained on MNIST. The first row of the figure depicts columns of $\mathbf{W}_1$; the second row depicts columns of $\mathbf{W}_1 \cdot \mathbf{W}_2$.

This hierarchical sparse coder is an instance of a parameterized family of intrinsic gradient networks that we have constructed. This family includes generalizations of the example network to different connection topologies and potential functions, but also includes networks with radically different dynamics. Intrinsic gradient networks consist of a time-varying vector[4] of real-valued units $\vec{x}(t) \in \mathbb{R}^n$. We write $\vec{x}$ to denote $\vec{x}(t)$ at some arbitrary point in time t. The point at which the network has finished computing and produced an output is defined by the output functions $\vec{F}(\vec{x},\vec{w})$ according to the fixed-point equation $\vec{F}(\vec{x},\vec{w}) = \vec{x}$, where $\vec{w}$ is a vector of parameters.[5] At the output states $\vec{x}^*$ where $\vec{F}(\vec{x}^*,\vec{w}) = \vec{x}^*$, the desirability of the outputs is defined by the loss function $L(\vec{x},\vec{w})$. We find that intrinsic gradient networks are characterized by the equation

$$\mathbf{S}(\vec{x},\vec{w}) \cdot \left(\vec{x} - \vec{F}(\vec{x})\right) = \vec{T}(\vec{x}) - \nabla L(\vec{x}) - \left(\nabla \vec{F}^\top(\vec{x})\right) \cdot \vec{T}(\vec{x})\,, \tag{2}$$

where $\mathbf{S}(\vec{x},\vec{w})$ is some matrix function and ∇ is the gradient with respect to $\vec{x}$. When equation 2 is satisfied, the training function $\vec{T}(\vec{x},\vec{w})$ can be used to calculate the gradient of the loss function according to

$$\frac{dL(\vec{x}^*,\vec{w})}{dw'} = \frac{\partial L(\vec{x}^*,\vec{w})}{\partial w'} + \vec{T}^\top(\vec{x}^*,\vec{w}) \cdot \frac{\partial \vec{F}(\vec{x}^*,\vec{w})}{\partial w'}$$

at the output states $\vec{x}^*$; this gradient takes into account the fact that $\vec{x}^*$ is a function of $\vec{w}$ defined by the fixed-point equation. Any network dynamics that converge to a fixed point of $\vec{F}(\vec{x},\vec{w})$ may be used, such as
$\frac{d\vec{x}(t)}{dt} = \frac{1}{\tau} \cdot \left(\vec{F}(\vec{x}(t),\vec{w}) - x(t)\right)$.

We construct an analytic solution for the output functions $\vec{F}(\vec{x})$ in equation 2 in terms of the loss function $L(\vec{x},\vec{w})$ and the training function $\vec{T}(\vec{x})$, in which $\mathbf{S}(\vec{x})$ is a free parameter; the details of this solution are not included here due to space constraints. A wide variety of different network topologies and dynamics can be induced by various choices of the training function $\vec{T}(\vec{x})$ and the free parameters in this solution. For instance, we can construct the hierarchical sparse coding network described above by dividing $\vec{x}$ into consecutive layers $\vec{x}_1$, $\vec{x}_2$, $\cdots$, $\vec{x}_n$ connected by parameter matrices $\mathbf{W}_i$, and choosing $\vec{T}(\vec{x}) = \mathbf{S}(\vec{x}) = \vec{x}$.

References

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

[2] Douglas, R. J., & Martin, K. A. C. (2004). Neuronal circuits of the neocortex. *Annual Review of Neuroscience*, *27*, 419–451.

[3] Olshausen, B. A., & Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, *381*, 607–609.

[4] Rao, R. P. N., & Ballard, D. H. (1999). Predictive coding in the visual cortex: A functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, *2*(1), 79 – 87.

[5] Salakhutdinov, R., & Hinton, G. (2009). Deep Boltzmann machines. In D. van Dyk & M. Welling (Eds.), *AISTATS 2009* (pp. 448-455).

[4] All vectors are column vectors unless otherwise noted.

[5] Most of our functions are dependent on both the network state $\vec{x}$ and the parameters $\vec{w}$, but we often omit explicit mention of the dependence on $\vec{w}$ to avoid cluttering our notation.