NONLINEAR ICA THROUGH LOW-COMPLEXITY AUTOENCODERS

Sepp Hochreiter

Fakultät für Informatik
Technische Universität München
80290 München, Germany
hochreit@in.tum.de

Jürgen Schmidhuber

IDSIA
Corso Elvezia 36
6900 Lugano, Switzerland
juergen@idsia.ch

ABSTRACT

We train autoencoders by Flat Minimum Search (FMS), a regularizer algorithm for finding low-complexity networks describable by few bits of information. As a by-product, this encourages nonlinear independent component analysis (ICA) and sparse codes of the input data.

Flat minima are regions in weight space where (a) there is tolerable small error and (b) you can perturb the weights without greatly affecting the network's output. Hence the weights may be given with low precision: few bits of information are required to describe the corresponding "simple" or low complexity-network. Low network complexity is generally associated with high generalization performance.

To simplify the algorithm for finding flat minima, we do not consider maximal connected regions but focus on so-called "boxes" within regions: for each weight vector w leading to tolerable small error, its box M_w in weight space is a W-dimensional hypercuboid with center w, where W is the number of weights. For simplicity, each edge of the box is taken to be parallel to one weight axis. Half the length of the box edge in direction of the axis corresponding to weight w_{ij} is denoted by δw_{ij}, which gives the precision of w_{ij}. M_w's box volume is defined by $V(\delta w) := 2^W \prod_{i,j} \delta w_{ij}$, where δw denotes the vector with components δw_{ij}. Our goal is to find large boxes within flat minima. Towards this end we try to find minimal $B := -\log\left(\frac{1}{2^W} V(\delta w)\right) = \sum_{i,j} -\log \delta w_{ij}$. Note the relationship to MDL: B is the number of bits (save a constant) required to describe all weights in the net.

FMS [1] minimizes $E = E_q + \lambda B$ by gradient descent, where E_q is the training set mean squared error, and $\lambda > 0$ scales the influence of $B = T1 + T2$, where

$$T1 := \sum_{i,j \in O \times H \cup H \times I} \log \sum_{k \in O} \left(\frac{\partial y^k}{\partial w_{ij}} \right)^2 \text{ and}$$

$$T2 := W \log \sum_{k \in O} \left(\sum_{i,j \in O \times H \cup H \times I} \frac{\left| \frac{\partial y^k}{\partial w_{ij}} \right|}{\sqrt{\sum_{k \in O} \left(\frac{\partial y^k}{\partial w_{ij}} \right)^2}} \right)^2 ,$$

where O, H, I denote index sets for output, hidden, input units, respectively. y^k denotes the activation of an output unit, which depends on the weights w_{ij}.

B is derived from two flatness conditions, FC1 and FC2. Perturbing the weights w by δw, we obtain $ED(w, \delta w) := \sum_{k \in O} \left(y^k(w + \delta w) - y^k(w) \right)^2$. To enforce flatness, FC1 wants to keep ED low:

$$ED(w, \delta w) \approx \sum_{k \in O} \left(\sum_{i,j} \frac{\partial y^k}{\partial w_{ij}} \delta w_{ij} \right)^2 \leq$$

$$\sum_{k \in O} \left(\sum_{i,j} \left| \frac{\partial y^k}{\partial w_{ij}} \right| |\delta w_{ij}| \right)^2 \leq \epsilon ,$$

where $\epsilon > 0$ is small enough to allow for linear approximation.

Many boxes M_w define a flat region and satisfy FC 1. To select a particular, very flat M_w, the following FC2 uses up degrees of freedom left by FC1 — it enforces minimal net output variance within a box given a constant box volume:

$$\forall i, j, u, v : \quad (\delta w_{ij})^2 \sum_{k \in O} \left(\frac{\partial y^k}{\partial w_{ij}} \right)^2 = (\delta w_{uv})^2 \sum_{k \in O} \left(\frac{\partial y^k}{\partial w_{uv}} \right)^2 .$$

Inserting FC2 into FC1 (using "=" instead of "$\leq$", since we search for maximal δw_{ij}), we obtain:

This work was supported by *DFG grant SCHM 942/3-1* and *DFG grant BR 609/10-2* from "Deutsche Forschungsgemeinschaft". J.S. would also like to acknowledge support from *SNF grant 21-43'417.95 "predictability minimization"*.

$$|\delta w_{uv}| = \frac{\sqrt{\epsilon}}{\sqrt{\sum_k \left(\frac{\partial o^k}{\partial w_{uv}}\right)^2} \sqrt{\sum_k \left(\sum_{i,j} \frac{\left|\frac{\partial o^k}{\partial w_{ij}}\right|}{\sqrt{\sum_k \left(\frac{\partial o^k}{\partial w_{ij}}\right)^2}} \right)^2}}$$

Inserting the previous equation into the definition of B we obtain above formula for B, where the constant factor $\frac{1}{2}$ and the term $\log \epsilon$ are skipped, since during gradient descent constant terms vanish and constant factors are absorbed by the learning factor.

A component function (CF) is the function determining the activation of a code component (hidden unit) in response to a given input. Consider the rewritten first term of B:

$$T1 = \sum_{i,j \in O \times H \cup H \times I} \left(2 \log f_i'(s_i) + 2 \log y^j + \log \sum_{k \in O} \left(\frac{\partial y^k}{\partial y^i}\right)^2\right) =$$
$$2 \sum_{i \in O \cup H} \text{fan-in}(i) \log f_i'(s_i) +$$
$$2 \sum_{j \in H \cup I} \text{fan-out}(j) \log y^j +$$
$$\sum_{i \in O \cup H} \text{fan-in}(i) \log \sum_{k \in O} \left(\frac{\partial y^k}{\partial y^i}\right)^2,$$

where $f_i'(s_i)$ is the derivative of the activation function of unit i with activation y^i and fan-in(i) (fan-out(i)) denotes the number of incoming (outgoing) weights of unit i.

$T1$ makes (1) unit activations decrease to zero, (2) first-order derivatives of activation functions decrease to zero, and (3) the influence of units on the output decreases to zero. $T1$ is the reason why low-complexity (or simple) CFs are preferred. Point (1) above favors sparse hidden unit activations (here: few active code components); point (2) favors non-informative hidden unit activations hardly affected by small input changes. Point (3) favors sparse hidden unit activations in the sense that "few hidden units contribute to producing the output".

$T2$ punishes units with similar influence on the output. We reformulate it:

$$T2 = W \log \left(|O| \ |O \times H|^2 + |I|^2 \sum_{k \in O} \sum_{i \in H} \sum_{u \in H} \frac{\left|\frac{\partial y^k}{\partial y^i}\right| \left|\frac{\partial y^k}{\partial y^u}\right|}{\sqrt{\sum_{k \in O} \left(\frac{\partial y^k}{\partial y^i}\right)^2} \sqrt{\sum_{k \in O} \left(\frac{\partial y^k}{\partial y^u}\right)^2}} \right),$$

where $|.|$ denotes the number of elements in a set.

We observe: (1) an output unit that is very sensitive with respect to two given hidden units will heavily contribute to $T2$. (2) This large contribution can be reduced by making both hidden units have large impact on other output units.

So FMS essentially tries to figure out a way of using (1) as few CFs as possible for each output unit (this leads to separation of CFs), while simultaneously (2) using the same CFs for as many output units as possible (common CFs).

The results above give rise to a new method for source separation: simply train autoencoders (e.g., [2, 3, 4, 5]) via FMS. The method's name is LOCOCODE [6, 7, 8, 9, 10], which stands for "*Lo*w-*co*mplexity *co*ding and *de*coding." LOCOCODE generates *lococodes* that (1) convey information about the input data, (2) can be computed by a low-complexity mapping (LCM), (3) can be decoded by a LCM (for alternative approaches using low-complexity nets to achieve ICA see [11, 12].).

The analysis above shows that LOCOCODE essentially attempts at describing single inputs with as few and as simple features as possible. This reflects a basic assumption, namely, that the true input "causes" are indeed few and simple. Training sets whose elements are all describable by few features will result in *sparse* codes. Sparseness [13, 14, 15, 16, 17, 18, 19] is not viewed as an *a priori* good thing, and is not enforced explicitly, but only if the input data indeed is naturally describable by a sparse code.

LOCOCODE (a) is not (like PCA and ICA [20, 21, 22, 23, 24, 25, 26, 27]) inherently limited to the linear case [10], (b) does not need (like ICA) *a priori* information about the number of independent data sources (even when ICA knows the number of sources, LOCOCODE outperforms ICA) [8], and (c) has a higher coding efficiency (bits per input pixel) than PCA and ICA [9]. Unlike codes obtained with standard autoencoders, lococodes are based on feature detectors, never unstructured, usually sparse, sometimes factorial or local (depending on statistical properties of the data). Although LOCOCODE is not explicitly designed to enforce sparse or factorial codes, it extracts optimal codes for non-linear, difficult versions of the "bars" benchmark problem, whereas ICA and PCA do not [10, 8]. It produces familiar, biologically plausible feature detectors when applied to real world images, and codes with fewer bits per pixel than ICA and PCA. Unlike ICA it does not need to know the number of independent sources. As a preprocessor for a vowel recognition benchmark problem it sets the stage for excellent classification performance [10].

Although LOCOCODE works well for visual inputs, it may be less useful for discovering input causes that can only be represented by high-complexity input transformations, or for discovering many features (causes) collectively determining single input components (as, e.g., in acoustic signal separation, where ICA does not suffer from the fact that

each source influences each input component and none is computable by a low-complexity function). For even more general, algorithmic methods reducing net complexity see [28]. For the authors' alternative neural approaches to nonlinear ICA see [29, 30].

Our results reveil an interesting, previously ignored connection between two important fields: regularization, and ICA. They may represent a first step towards unification of regularization and unsupervised learning.

1. REFERENCES

[1] S. Hochreiter and J. Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

[2] P. Baldi and K. Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2:53–58, 1989.

[3] M. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37:233–243, 1991.

[4] E. Oja. Data compression, feature extraction, and autoassociation in feedforward neural networks. In T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas, editors, *Artificial Neural Networks*, volume 1, pages 737–745. Elsevier Science publishers B.V., North-Holland, 1991.

[5] D. DeMers and G. Cottrell. Non-linear dimensionality reduction. In J. D. Cowan S. J. Hanson and C. L. Giles, editors, *Advances in Neural Information Processing Systems 5*, pages 580–587. Morgan Kaufmann, San Mateo, CA, 1993.

[6] S. Hochreiter and J. Schmidhuber. Unsupervised coding with Lococode. In W. Gerstner, A. Germond, M. Hasler, and J.-D. Nicoud, editors, *Proceedings of the International Conference on Artificial Neural Networks, Lausanne, Switzerland*, pages 655–660. Springer, 1997.

[7] S. Hochreiter and J. Schmidhuber. Low-complexity coding and decoding. In K. M. Wong, I. King, and D. Yeung, editors, *Theoretical Aspects of Neural Computation (TANC 97), Hong Kong*, pages 297–306. Springer, 1997.

[8] S. Hochreiter and J. Schmidhuber. Lococode versus PCA and ICA. In L. Niklasson, M. Boden, and T. Ziemke, editors, *Proceedings of the International Conference on Artificial Neural Networks, Skövde, Sweden*, pages 669–674. Springer, 1998.

[9] S. Hochreiter and J. Schmidhuber. Source separation as a by-product of regularization. In M. Kearns, S. A. Solla, and D. Cohn, editors, *Advances in Neural Information Processing Systems 12*. MIT Press, Cambridge MA, 1999.

[10] S. Hochreiter and J. Schmidhuber. Feature extraction through LOCOCODE. *Neural Computation*, 11(3), 1999. In press.

[11] H. Lappalainen. Ensemble learning for independent component analysis. In J.-F. Cardoso, C. Jutten, and P. Loubaton, editors, *Proceedings of the First International Workshop on Independent Component Analysis and Signal Separation, Aussois, France*, pages 7–12, 1999.

[12] P. Pajunen. Blind source separation of natural signals based on approximate complexity minimization. In J.-F. Cardoso, C. Jutten, and P. Loubaton, editors, *Proceedings of the First International Workshop on Independent Component Analysis and Signal Separation, Aussois, France*, pages 267–270, 1999.

[13] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

[14] Z. Li. A theory of the visual motion coding in the primary visual cortex. *Neural Computation*, 8(4):705–730, 1995.

[15] H. B. Barlow. *Understanding natural vision*. Springer-Verlag, Berlin, 1983.

[16] P. Földiák and M. P. Young. Sparse coding in the primate cortex. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 895–898. The MIT Press, Cambridge, Massachusetts, 1995.

[17] G. Palm. On the information storage capacity of local learning rules. *Neural Computation*, 4(2):703–711, 1992.

[18] M. S. Lewicki and B. A. Olshausen. Inferring sparse, overcomplete image codes using an efficient coding framework. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems 10*, pages 815–822. MIT Press, Cambridge MA, 1998.

[19] A. Hyvarinen, P. Hoyer, and E. Oja. Sparse code shrinkage: Denoising by maximum likelihood extimation. In M. Kearns, S. A. Solla, and D. Cohn, editors, *Advances in Neural Information Processing Systems 12*. MIT Press, Cambridge MA, 1999.

[20] J.-P. Nadal and N. Parga. Redundancy reduction and independent component analysis: Conditions on cumulants and adaptive approaches. *Neural Computation*, 9(7):1421–1456, 1997.

[21] P. Comon. Independent component analysis – a new concept? *Signal Processing*, 36(3):287–314, 1994.

[22] C. Jutten and J. Herault. Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24(1):1–10, 1991.

[23] J.-F. Cardoso and A. Souloumiac. Blind beamforming for non Gaussian signals. *IEE Proceedings-F*, 140(6):362–370, 1993.

[24] S. Amari, A. Cichocki, and H.H. Yang. A new learning algorithm for blind signal separation. In David S. Touretzky, Michael C. Mozer, and Michael E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8*, pages 757–763. The MIT Press, Cambridge, MA, 1996.

[25] A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.

[26] L. Molgedey and H. G. Schuster. Separation of independent signals using time-delayed correlations. *Phys. Reviews Letters*, 72(23):3634–3637, 1994.

[27] G. Yang and S. Amari. Adaptive online learning algorithms for blind source separation: Maximum entropy and minimum mutual information. *Neural Computation*, 9(7):1457–1482, 1997.

[28] J. Schmidhuber. Discovering neural nets with low Kolmogorov complexity and high generalization capability. *Neural Networks*, 10(5):857–873, 1997.

[29] J. Schmidhuber. Learning factorial codes by predictability minimization. *Neural Computation*, 4(6):863–879, 1992.

[30] J. Schmidhuber, M. Eldracher, and B. Foltin. Semilinear predictability minimization produces well-known feature detectors. *Neural Computation*, 8(4):773–786, 1996.