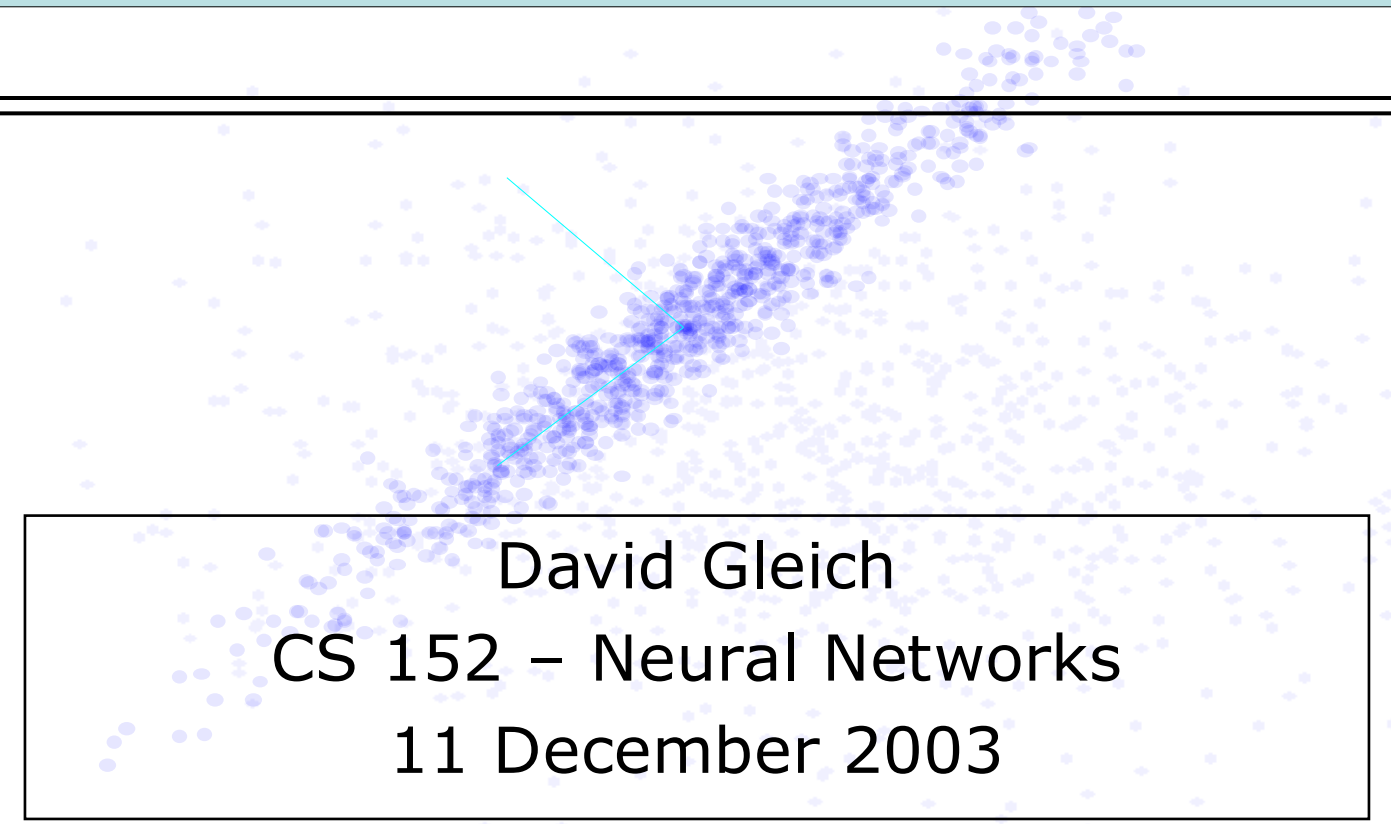
Principal Component Analysis and Independent Component Analysis in Neural Networks



David Gleich

CS 152 – Neural Networks

11 December 2003

Outline

- Review PCA and ICA
- Neural PCA Models
- Neural ICA Models
- Experiments and Results
- Implementations
- Summary/Conclusion
- Questions

Principal Component Analysis

- PCA identifies an m dimensional explanation of n dimensional data where m < n.
- Originated as a statistical analysis technique.
- PCA attempts to minimize the reconstruction error under the following restrictions
  - Linear Reconstruction
  - Orthogonal Factors
- Equivalently, PCA attempts to maximize variance.

Independent Component Analysis

- Also known as Blind Source Separation.
- Proposed for neuromimetic hardware in 1983 by Herault and Jutten.
- ICA seeks components that are independent in the statistical sense.

Two variables x, y are *statistically independent* iff $P(x \cap y) = P(x)P(y)$.

Equivalently,

$E\{g(x)h(y)\} - E\{g(x)\}E\{h(y)\} = 0$

where g and h are any functions.

Independent Component Analysis

Given m signals of length n, construct the data matrix

$$X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix}.$$

We assume that X consists of m sources such that

$$X = AS$$

where A is an unknown m by m mixing matrix and S is m independent sources.

Independent Component Analysis

ICA seeks to determine a matrix W such that

$$Y = BX$$

where W is an m by m matrix and Y is the set of independent source signals, i.e. the independent components.

$$B \approx A^{-1} \Rightarrow Y = A^{-1}AX = X$$

- Note that the components need not be orthogonal, but that the reconstruction is still linear.

PCA with Neural Networks

- Most PCA Neural Networks use some form of Hebbian learning.

  "Adjust the strength of the connection between units A and B in proportion to the product of their simultaneous activations."

$$w_{k+1} = w_k + \beta_k(y_k\, x_k)$$

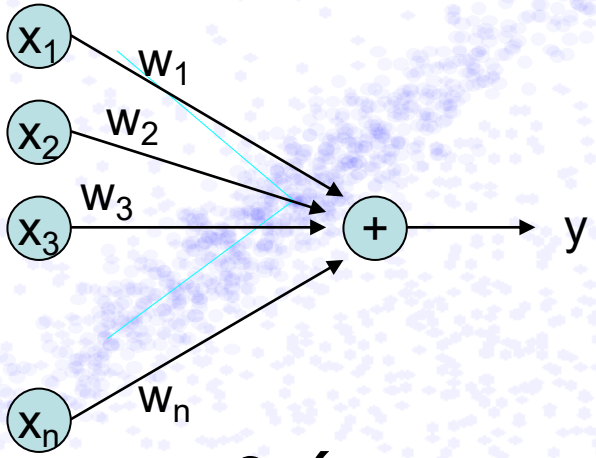
- Applied directly, this equation is unstable.

$$||w_k||^2 \to \infty \text{ as } k \to \infty$$

- Important Note: neural PCA algorithms are unsupervised.

PCA with Neural Networks

- Another fix: Oja's rule.
- Proposed in 1982 by Oja and Karhunen.

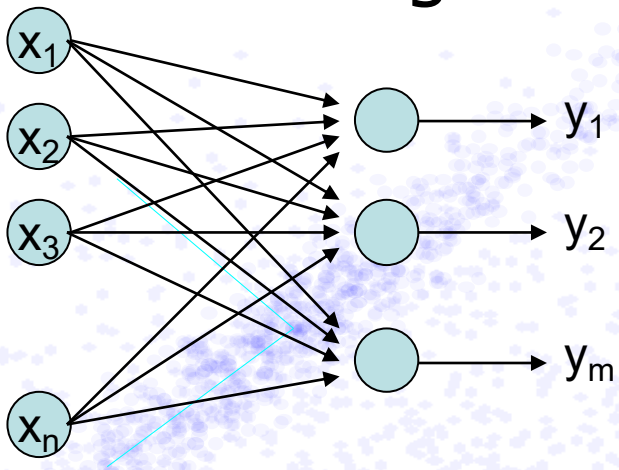


$$w_{k+1} = w_k + \beta_k(y_k x_k - y_k^2 w_k)$$

- This is a linearized version of the normalized Hebbian rule.
- Convergence, as $k \to \infty$, $w_k \to e_1$.

PCA with Neural Networks

- Generalized Hebbian Algorithm



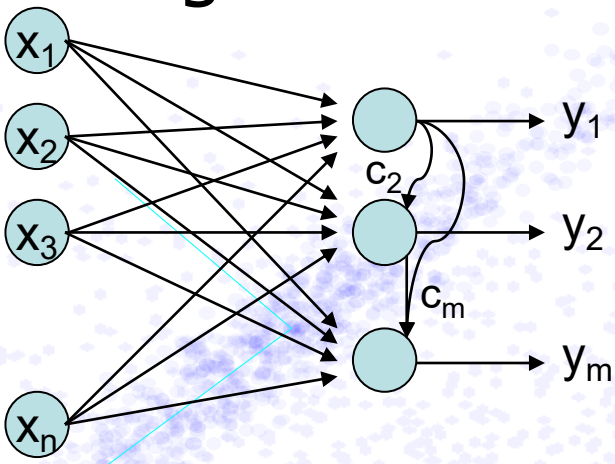
$$y = Wx$$

$$\Delta w_{ij,k} = \beta_k y_{ik} x_{jk} - \beta_k y_{ik} \sum_{l \le i} y_{lk} w_{lj,k}$$

$$\Delta W = \beta_k y x^T - \beta_k W \, LT(y y^T)$$

PCA with Neural Networks

- APEX Model: Kung and Diamantaras



$$y = Wx - Cy \Leftrightarrow y = (I+C)^{-1}Wx \approx (I-C)Wx$$

$$C = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ c_{21} & 0 & 0 & \dots & 0 \\ c_{31} & c_{32} & 0 & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ c_{m-1,1} & \dots & c_{m-1,m-2} & 0 & 0 \\ c_{m,1} & \dots & \dots & c_{m,m-1} & 0 \end{bmatrix}$$

PCA with Neural Networks

- APEX Learning

$$\Delta w_{i,j,k} = \beta_k (y_{ik} x_{jk} - y_{ik}^2 w_{i,j,k})$$
$$\Delta c_{i,j,k} = \beta_k (y_{ik} y_{jk} - y_{ik}^2 c_{i,j,k})$$

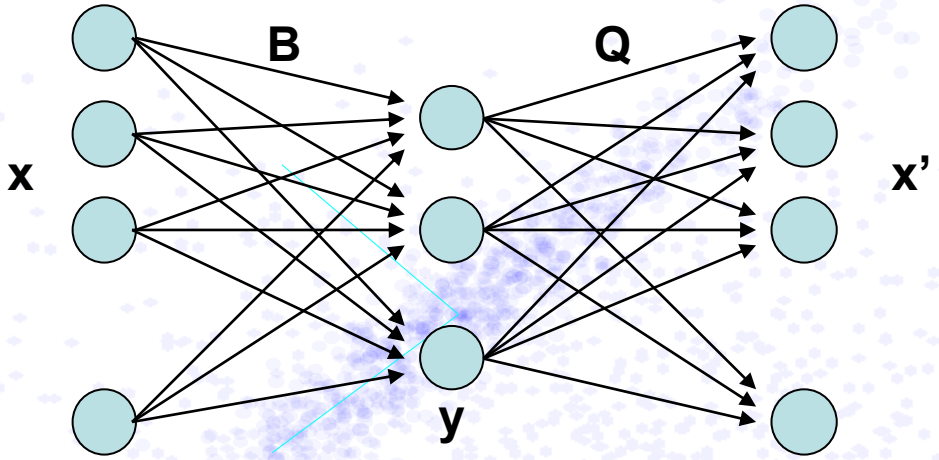
- Properties of APEX model:
  - Exact principal components
  - Local updates, Δw_{ab} only depends on x_a, x_b, w_{ab}
  - "-Cy" acts as an orthogonalization term

Neural ICA

- ICA is typically posed as an optimization problem.
- Many iterative solutions to optimization problems can be cast into a neural network.

Feed-Forward Neural ICA

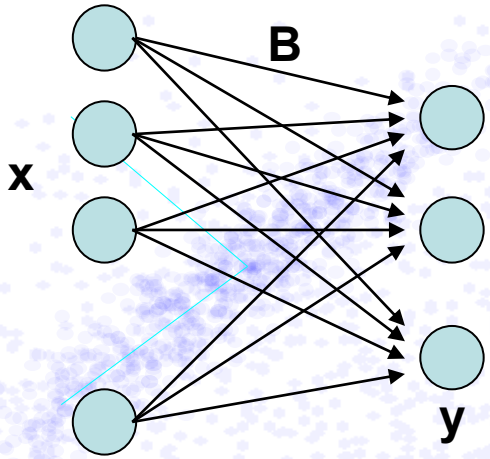
General Network Structure



1. Learn B such that y = Bx has independent components.
2. Learn Q which minimizes the mean squared error reconstruction.

Feed-Forward Neural ICA

General Network Structure



1. Learn B such that y = Bx has independent components.

Neural ICA

- Herault-Jutten: local updates

$$B = (I+S)^{-1}$$

$$S_{k+1} = S_k + \beta_k g(y_k) h(y_k^T)$$

$g = t$, $h = t^3$; g = hardlim, h = tansig

- Bell and Sejnowski: information theory

$$B_{k+1} = B_k + \beta_k [B_k^{-T} + z_k x_k^T]$$

$$z(i) = \partial/\partial u(i)\ \partial u(i)/\partial y(i)$$

$u = f(Bx)$; f = tansig, etc.

Neural ICA

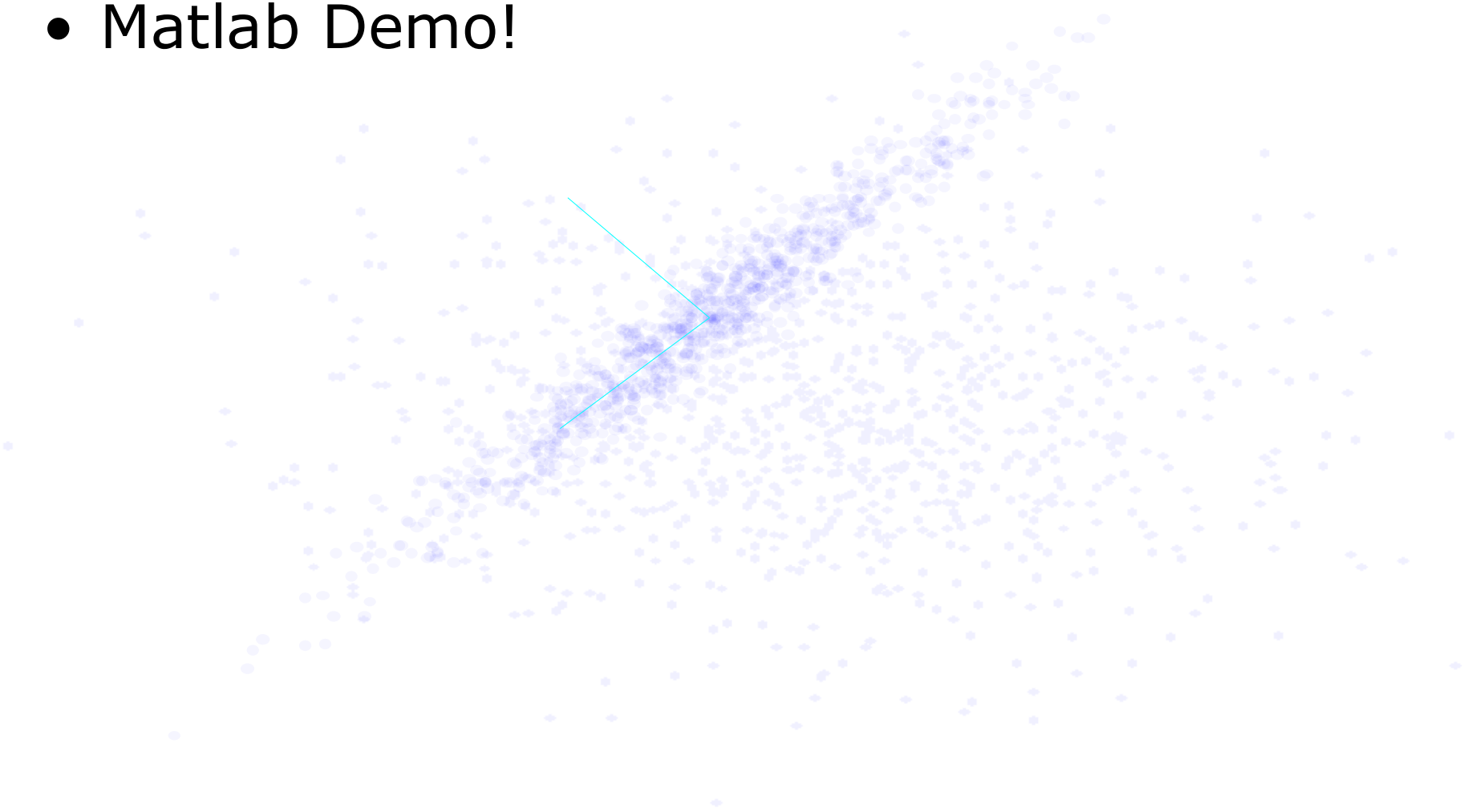
- EASI (Equivariant Adaptive Separation via Independence): Cardoso et al

$$B_{k+1}=B_k-\beta_k[y_k y_k^T-I+g(y_k)h(y_k)^T-(y_k)g(y_k)^T]B_k$$

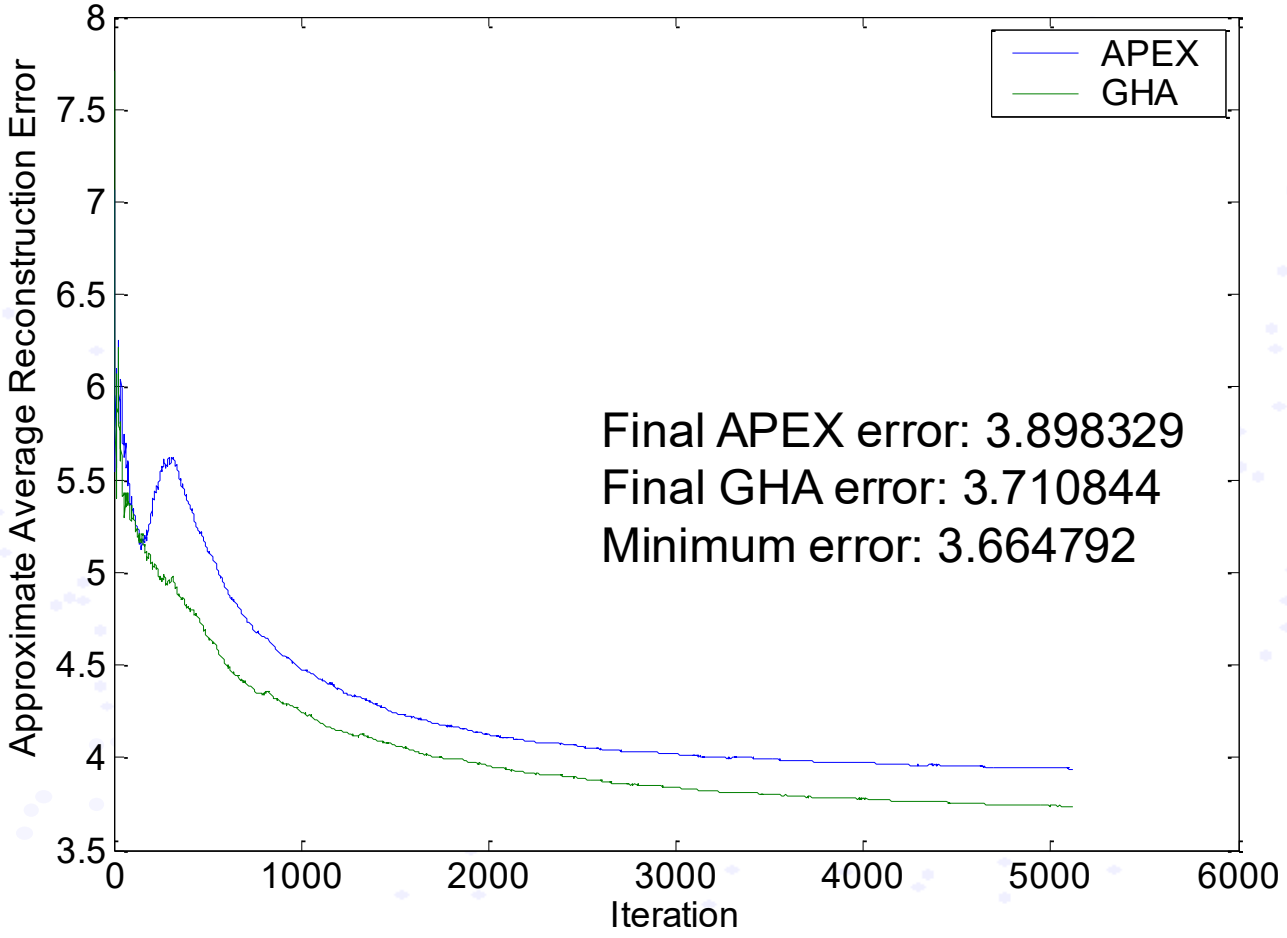
$$g=t,\ h=\text{tansig}(t)$$

Oja's Rule in Action

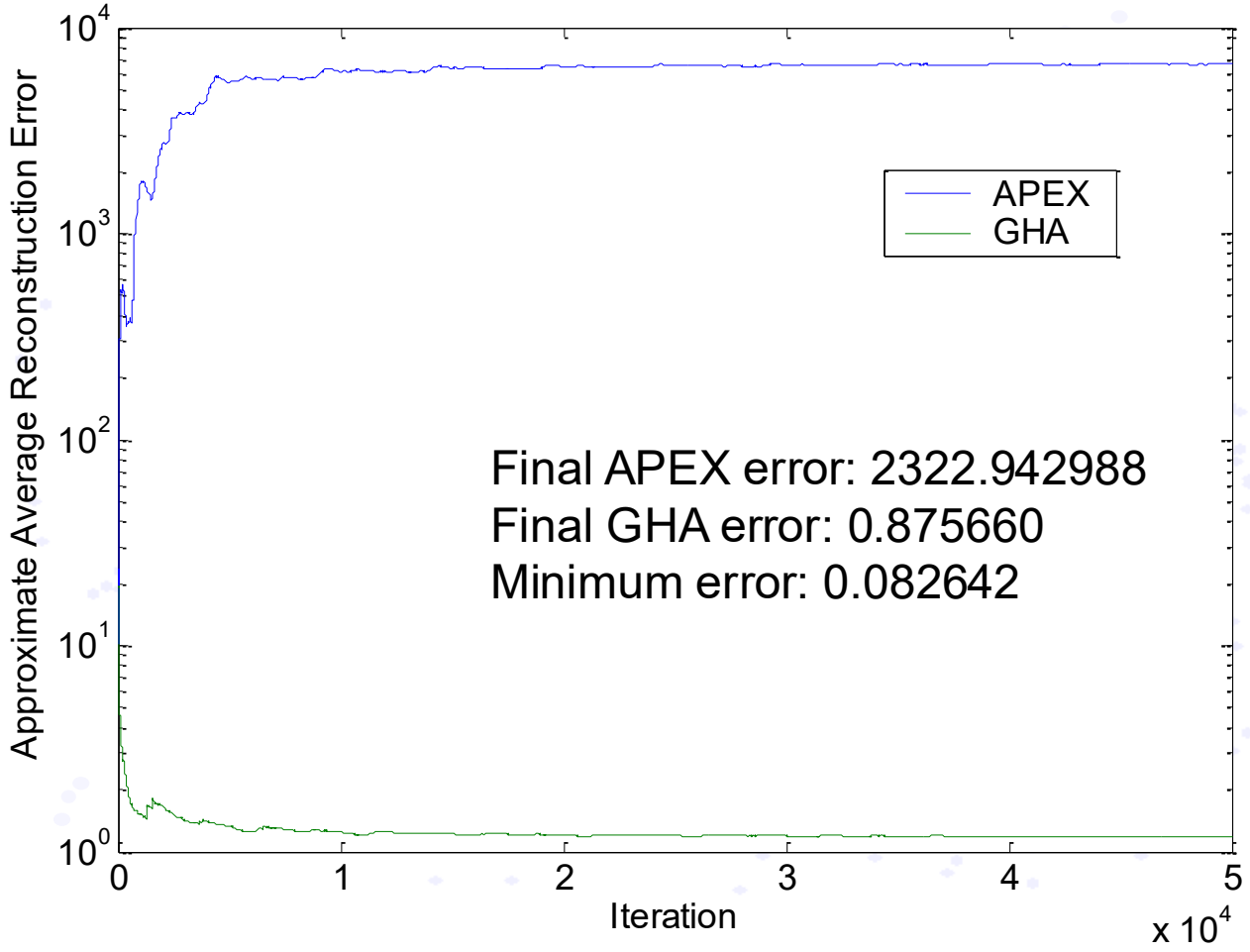
- Matlab Demo!



PCA Error Plots – Close Eigenvalues



PCA Error Plots – Separate Eigenvalues



Observations

- Why just the PCA subspace?
  - Bad Oja's example – Matlab demo!
- Why can APEX fail?
  - "-Cy" term continually orthogonalizes

ICA Image Example

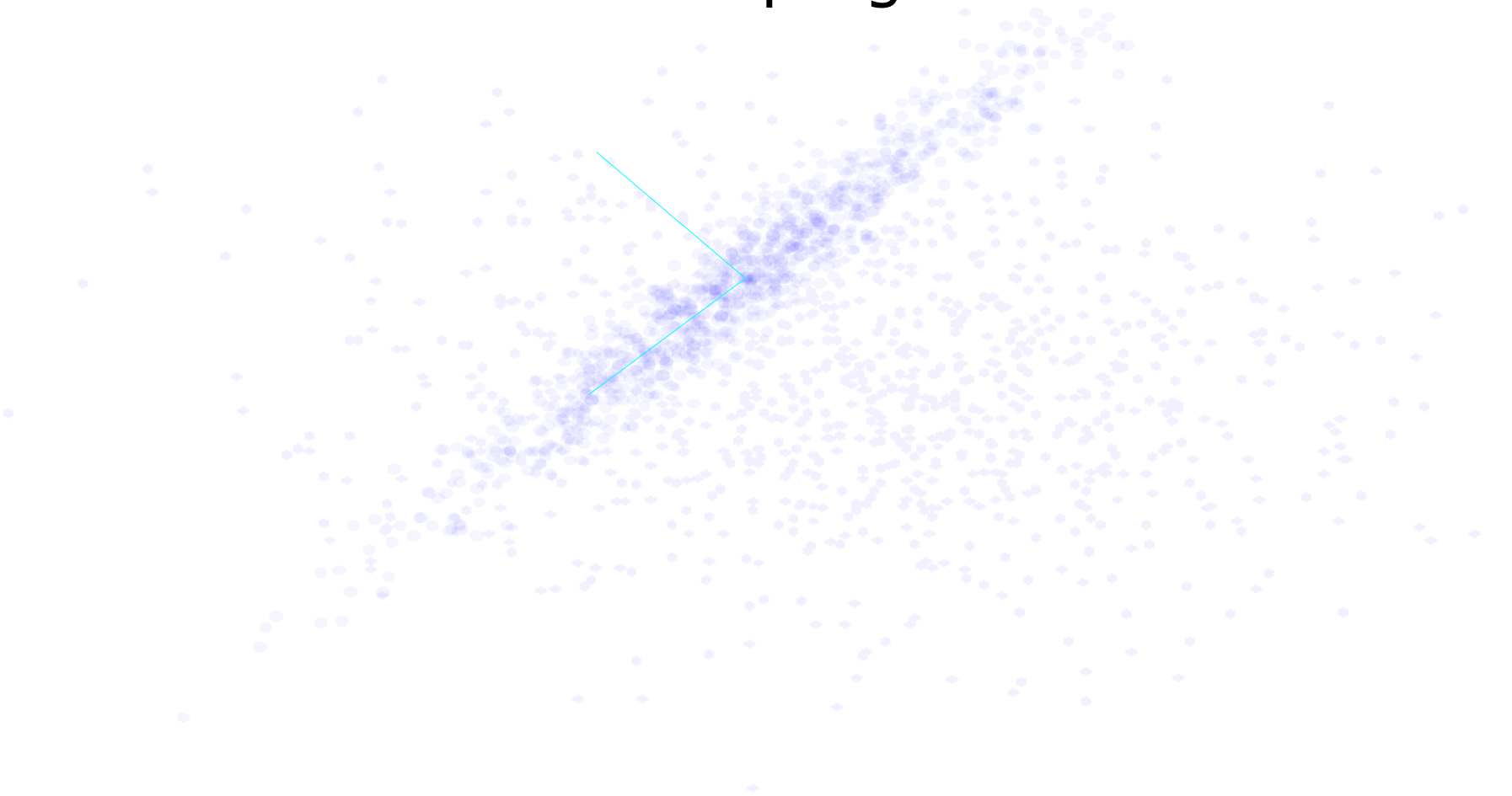
ICA Image Example

Bell and Sejnowski

EASI

Real World Data

- 8 channel ECG from pregnant mother



Implementation

- Matlab sources for each neural algorithm, and supporting functions
  - PCA: oja.m; gha.m; apex.m
  - ICA: hj.m; easi.m; bsica.m
  - Evaluation: pcaerr.m
  - Demos: fetal_plots.m; pca_error_plots.m; ojademo.m; ica_image_demo.m
  - Data generators: evalpca.m; evalica.m

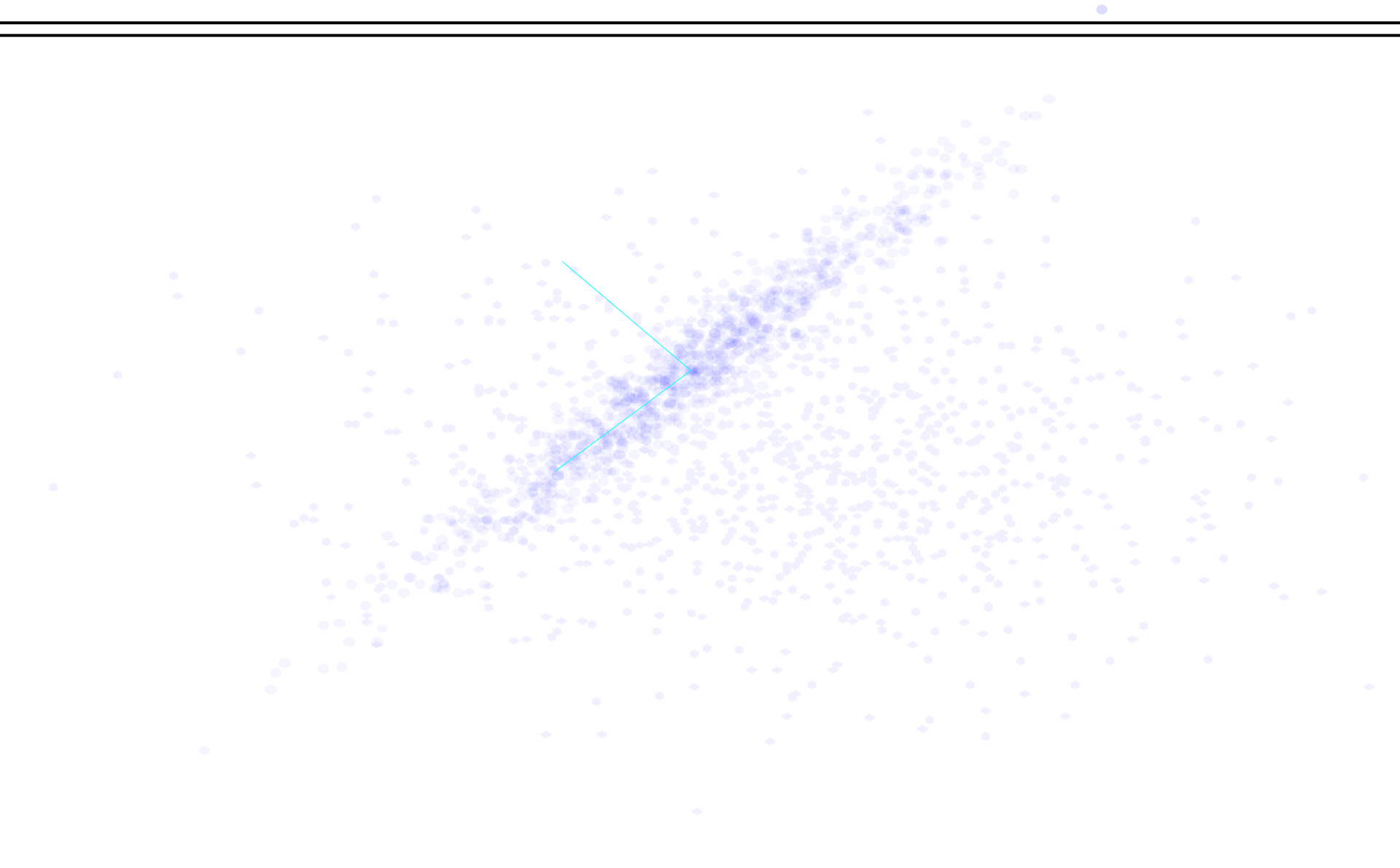
Future Work

- Investigate adaptive learning and changing separations
  - Briefly examined this; the published paper on ICA adjusted the learning rate correctly, but didn't converge to a separation.
- Convergence of PCA algorithms based on eigenvalue distribution?
- Use APEX to compute "more" components than desired?

Summary

- Neural PCA:
  - Algorithms work "sometimes." They tend to find the PCA subspace rather than the exact PCA components
  - GHA is better than APEX
- Neural ICA:
  - Algorithms converge to something reasonable.
  - Bell and Sejnowski's ICA possibly superior to EASI.

Questions?



References

- Diamantras, K.I. and S. Y. Kung. *Principal Component Neural Networks*.
- Comon, P. "Independent Component Analysis, a new concept?" In *Signal Processing*, vol. 36, pp. 287-314, 1994.
- FastICA, http://www.cis.hut.fi/projects/ica/fastica/
- Oursland, A., J. D. Paula, and N. Mahmood. "Case Studies in Independent Component Analysis."
- Weingessel, A. "An Analysis of Learning Algorithms in PCA and SVD Neural Networks."
- Karhunen, J. "Neural Approaches to Independent Component Analysis."
- Amari, S., A. Cichocki, and H. H. Yang. "Recurrent Neural Networks for Blind Separation of Sources."