

ICA

Introduction

If you recall, in Sparse Coding (<http://ufldl.stanford.edu/tutorial/unsupervised/SparseCoding>), we wanted to learn an **over-complete** basis for the data. In particular, this implies that the basis vectors that we learn in sparse coding will not be linearly independent. While this may be desirable in certain situations, sometimes we want to learn a linearly independent basis for the data. In independent component analysis (ICA), this is exactly what we want to do. Further, in ICA, we want to learn not just any linearly independent basis, but an **orthonormal** basis for the data. (An orthonormal basis is a basis  $(\phi_1, \dots \phi_n)$  such that  $\phi_i \cdot \phi_j = 0$  if  $i \neq j$  and 1 if  $i = j$ ).

Like sparse coding, independent component analysis has a simple mathematical formulation. Given some data  $x$ , we would like to learn a set of basis vectors which we represent in the columns of a matrix  $W$ , such that, firstly, as in sparse coding, our features are **sparse**; and secondly, our basis is an **orthonormal** basis. (Note that while in sparse coding, our matrix  $A$  was for mapping **features**  $s$  to **raw data**, in independent component analysis, our matrix  $W$  works in the opposite direction, mapping **raw data**  $x$  to **features** instead). This gives us the following objective function:

$$J(W) = \|Wx\|_1$$

This objective function is equivalent to the sparsity penalty on the features  $s$  in sparse coding, since  $Wx$  is precisely the features that represent the data. Adding in the orthonormality constraint gives us the full optimization problem for independent component analysis:

$$\begin{array}{ll} \text{minimize} & \|Wx\|_1 \\ \text{s. t.} & WW^T = I \end{array}$$

As is usually the case in deep learning, this problem has no simple analytic solution, and to make matters worse, the orthonormality constraint makes it slightly more difficult to optimize for the objective using gradient descent - every iteration of gradient descent must be followed by a step that maps the new basis back to the space of orthonormal bases (hence enforcing the constraint).

In practice, optimizing for the objective function while enforcing the orthonormality constraint (as described in the section below) is feasible but slow. Hence, the use of orthonormal ICA is limited to situations where it is important to obtain an orthonormal basis.

Orthonormal ICA

The orthonormal ICA objective is:

$$\begin{array}{ll} \text{minimize} & \|Wx\|_1 \\ \text{s. t.} & WW^T = I \end{array}$$

Observe that the constraint  $WW^T = I$  implies two other constraints.

Firstly, since we are learning an orthonormal basis, the number of basis vectors we learn must be less than the dimension of the input. In particular, this means that we cannot learn over-complete bases as we usually do in [\[\[Sparse Coding: Autoencoder Interpretation | sparse coding\]\]](#).

Secondly, the data must be ZCA whitened (<http://ufldl.stanford.edu/tutorial/unsupervised/PCAWhitening>) with no regularization (that is, with  $\epsilon$  set to 0).

Hence, before we even begin to optimize for the orthonormal ICA objective, we must ensure that our data has been **whitened**, and that we are learning an **under-complete** basis.

Following that, to optimize for the objective, we can use gradient descent, interspersing gradient descent steps with projection steps to enforce the orthonormality constraint. Hence, the procedure will be as follows:

Repeat until done:

- 1.  $W \leftarrow W - \alpha \nabla_W \|Wx\|_1$
- 2.  $W \leftarrow \text{proj}_U W$  where  $U$  is the space of matrices satisfying  $WW^T = I$

In practice, the learning rate  $\alpha$  is varied using a line-search algorithm to speed up the descent, and the projection step is achieved by setting  $W \leftarrow (WW^T)^{-\frac{1}{2}} W$ , which can actually be seen as ZCA whitening ( [TODO](#) : explain how it is like ZCA whitening).

Supervised Learning and Optimization
Linear Regression ( <a href="http://ufldl.stanford.edu/tutorial/supervised/LinearRegression">http://ufldl.stanford.edu/tutorial/supervised/LinearRegression</a> )
Logistic Regression ( <a href="http://ufldl.stanford.edu/tutorial/supervised/LogisticRegression">http://ufldl.stanford.edu/tutorial/supervised/LogisticRegression</a> )
Vectorization ( <a href="http://ufldl.stanford.edu/tutorial/supervised/Vectorization">http://ufldl.stanford.edu/tutorial/supervised/Vectorization</a> )
Debugging: Gradient Checking ( <a href="http://ufldl.stanford.edu/tutorial/supervised/DebuggingGradientChecking">http://ufldl.stanford.edu/tutorial/supervised/DebuggingGradientChecking</a> )
Softmax Regression ( <a href="http://ufldl.stanford.edu/tutorial/supervised/SoftmaxRegression">http://ufldl.stanford.edu/tutorial/supervised/SoftmaxRegression</a> )
Debugging: Bias and Variance ( <a href="http://ufldl.stanford.edu/tutorial/supervised/DebuggingBiasAndVariance">http://ufldl.stanford.edu/tutorial/supervised/DebuggingBiasAndVariance</a> )
Debugging: Optimizers and Objectives ( <a href="http://ufldl.stanford.edu/tutorial/supervised/DebuggingOptimizersAndObjectives">http://ufldl.stanford.edu/tutorial/supervised/DebuggingOptimizersAndObjectives</a> )
Supervised Neural Networks
Multi-Layer Neural Networks ( <a href="http://ufldl.stanford.edu/tutorial/supervised/MultiLayerNeuralNetworks">http://ufldl.stanford.edu/tutorial/supervised/MultiLayerNeuralNetworks</a> )
Exercise: Supervised Neural Network ( <a href="http://ufldl.stanford.edu/tutorial/supervised/ExerciseSupervisedNeuralNetwork">http://ufldl.stanford.edu/tutorial/supervised/ExerciseSupervisedNeuralNetwork</a> )
Supervised Convolutional Neural Network
Feature Extraction Using Convolution ( <a href="http://ufldl.stanford.edu/tutorial/supervised/FeatureExtractionUsingConvolution">http://ufldl.stanford.edu/tutorial/supervised/FeatureExtractionUsingConvolution</a> )
Pooling ( <a href="http://ufldl.stanford.edu/tutorial/supervised/Pooling">http://ufldl.stanford.edu/tutorial/supervised/Pooling</a> )
Exercise: Convolution and Pooling ( <a href="http://ufldl.stanford.edu/tutorial/supervised/ExerciseConvolutionAndPooling">http://ufldl.stanford.edu/tutorial/supervised/ExerciseConvolutionAndPooling</a> )
Optimization: Stochastic Gradient Descent ( <a href="http://ufldl.stanford.edu/tutorial/supervised/OptimizationStochasticGradientDescent">http://ufldl.stanford.edu/tutorial/supervised/OptimizationStochasticGradientDescent</a> )
Convolutional Neural Network ( <a href="http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork">http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork</a> )
Excercise: Convolutional Neural Network ( <a href="http://ufldl.stanford.edu/tutorial/supervised/ExerciseConvolutionalNeuralNetwork">http://ufldl.stanford.edu/tutorial/supervised/ExerciseConvolutionalNeuralNetwork</a> )
Unsupervised Learning
Autoencoders ( <a href="http://ufldl.stanford.edu/tutorial/unsupervised/Autoencoders">http://ufldl.stanford.edu/tutorial/unsupervised/Autoencoders</a> )
PCA Whitening ( <a href="http://ufldl.stanford.edu/tutorial/unsupervised/PCAWhitening">http://ufldl.stanford.edu/tutorial/unsupervised/PCAWhitening</a> )
Exercise: PCA Whitening ( <a href="http://ufldl.stanford.edu/tutorial/unsupervised/ExercisePCAWhitening">http://ufldl.stanford.edu/tutorial/unsupervised/ExercisePCAWhitening</a> )
Sparse Coding ( <a href="http://ufldl.stanford.edu/tutorial/unsupervised/SparseCoding">http://ufldl.stanford.edu/tutorial/unsupervised/SparseCoding</a> )
ICA ( <a href="http://ufldl.stanford.edu/tutorial/unsupervised/ICA">http://ufldl.stanford.edu/tutorial/unsupervised/ICA</a> )
RICA ( <a href="http://ufldl.stanford.edu/tutorial/unsupervised/RICA">http://ufldl.stanford.edu/tutorial/unsupervised/RICA</a> )
Exercise: RICA ( <a href="http://ufldl.stanford.edu/tutorial/unsupervised/ExerciseRICA">http://ufldl.stanford.edu/tutorial/unsupervised/ExerciseRICA</a> )
Self-Supervised Learning

Self-Taught Learning
Self-Taught Learning ( <a href="http://ufldl.stanford.edu/tutorial/selftaughtlearning/SelfTaughtLearning">http://ufldl.stanford.edu/tutorial/selftaughtlearning/SelfTaughtLearning</a> )
Exercise: Self-Taught Learning ( <a href="http://ufldl.stanford.edu/tutorial/selftaughtlearning/ExerciseSelfTaughtLearning">http://ufldl.stanford.edu/tutorial/selftaughtlearning/ExerciseSelfTaughtLearning</a> )