

Assignment 3

IFT 6268 Winter 2015

Date posted: March 18, 2015

Due: March 31, 2015, at the *beginning* of class

1. (1/10) In what way does whitening help learn an ICA model?
2. (3/10) Recall that the energy function of an autoencoder, $r(\mathbf{x})$, with a single layer of non-linear hidden units and real valued outputs is defined as the scalar function $\mathcal{F}(\mathbf{x})$ whose derivative is the vector field $r(\mathbf{x}) - \mathbf{x}$.

Write down the energy function of an autoencoder with sigmoid hidden units and real valued outputs. Show that the derivative of the energy function with respect to \mathbf{x} is indeed $(r(\mathbf{x}) - \mathbf{x})$. Then do the same for an autoencoder with ReLU hidden units.

3. (6/10) In this question you will extend the local feature based CIFAR classification experiment in Assignment 2, Question 6, by adding (i) feature learning, (ii) a simple form of pooling, and then (iii) back-prop to train all parameters together.

Recall that in that question you turned each image into a 100-dimensional feature vector by filtering the image with whitened random patches, followed by the ReLU non-linearity, $\max(0, \mathbf{f}_i^T \mathbf{x})$, followed by averaging the result over the image.

- (a) Use the same pipeline, with patchwise ZCA-whitening as preprocessing, but instead of defining \mathbf{f}_i to be random patches, *learn* the prototypes using the online (Hebbian) K-means algorithm we discussed in class. Train on a subset of image patches from the training data.
- (b) Then, using your K-means features, apply the quadrant-pooling approach proposed in “An analysis of single-layer networks in unsupervised feature learning” by Coates, Lee, Ng. AISTATS 2011: First compute for each image the *mean feature vector for all those features that came from the same quadrant in the image*. This will give you a 100-dimensional vector for *each* quadrant (and thus *four* 100-dimensional feature vectors per image). For each image, concatenate the four feature vectors. You will now have a 400-dimensional descriptor per image.
- (c) Finally, instead of using K-means features, use back-prop to learn all parameters of the pipeline jointly by optimizing the log-probability of the softmax output-layer. You may want to implement the pipeline using the theano to this end, and make use of its conv2d-operation.

For this assignment perform your experiments on the *whole* CIFAR dataset not a subset. If the size of the dataset causes problems talk to instructor.

What to hand in:

- An image showing the 100 features learned using online K-means.
- An image showing the 100 features learned using back-prop.
- For each of the three modifications described above, what is the percentage of correctly classified training cases and of correctly classified test cases after training? It is up to you to experiment with hyperparameters, initialization and/or regularization strategies to get good performance. Mention in detail any modifications (including regularization approaches) to the basic pipeline that you used. Also mention which learning rate(s) you used and for how many iterations you trained your models, as well as the final log-probability of the training data and of the test-data.