Vision in Human and Machine - Tutorial 2
Sparse Coding Methods

Heiko Wersing

1 The bars test

The bars test is a test problem that has been proposed by Foldiak (1990) to evaluate the performance of feature learning approaches. The input image ensemble is defined as a number of square images, where each image is composed of a number of randomly placed horizontal and vertical bars.

1.1 Creating the bar images

Write a Matlab function `create_bar_images (data_path,dim,num_bars,nonlinear)` that creates `num_bars` matrices with size `dim x dim`, and sets 2 random columns and 2 random rows by adding '1.0' to the zero initialization. Write the matrices to a set of images '[data_path]/obj1__0.pgm', '[data_path]/obj1__1.pgm' , ... and check if they are appropriate. Use 'nonlinear' to toggle the summation of bars, if nonlinear=0 then bars add linearly, while for nonlinear=1, the bar crossings are limited to a maximum value of 1. Use a statement like

```
imwrite(img,[data_path '/obj1__' num2str(i-1) '.pgm'],'PGM','Encoding','rawbits');
```

for saving the pgm images. You can use `imread` to check, if the data is correct.

1.2 Data formats for preparing experiments

In this tutorial we perform sparse coding learning based on direct image input. Consequently, the input patterns for the learning are small n x n patches, sampled from the input image images ensemble. To build a visual hierarchy like in the ventral pathway, it is also necessary to perform the learning based on the output of a previous feature detection stage. Explicitly, if we perfom learning based on the output of a Gabor detection stage with 4 orientations, our input patches are now n x n x 4, thus can be modelled in Matlab as a 3-dimensional matrix.

To allow a consistent computation for these cases, the input image data is converted into a standard representation for representating feature map activity, the `dataset`. You can use the function `convert_bars_dataset` to convert the bars image ensembles into this standard format. You have to adapt the size parameters in this function according to your generated image set. Analyze this function, load the saved Matlab variable 'dataset' and investigate its content. The 'dataset' is a 4-dimensional matrix, with `dataset(x,y,feature_plane,image_index)`. For plain images `feature_plane=1` holds, if we save the output of 4 Gabor filters on an image set, then `feature_plane=1,...,4` (will be used in next Tutorial).

2 The sparse coding algorithm

After you have generated a set of bar images and saved the corresponding dataset, you can perform the sparse coding learning of a basis representation. Have a look at `sparse_coding.m` and the core learning algorithm for nonnegative sparse coding `nnsc.m` and try to understand the main parts.

2.1 Investigating representation properties

With a call of `sparse_coding(...)` you can trigger the learning of a basis representation (see interface description and parameter example in .m file). The basis is visualized during learning and you can later inspect the results also by replacing the 'learn' option with the 'show' option in the call to `sparse coding`. If you encounter slow convergence or divergence you must adapt the gradient stepsize.

Investigate the following questions:

1. How does the representation depend on the number of available basis vectors and the basis patch size `wsize` ?
2. What is the effect of changing the parameter lambda, and how can this be understood ?
3. Is there a difference in the learning results for the linear and the nonlinear bar combination image sets ?