Progress Report
1-14 September 2012

Amin Karbasi
E-mail: amin,karbasi@epfl.ch
Amir Hesam Salavati
E-mail: hesam.salavati@epfl.ch

Supervisor: Prof. Amin Shokrollahi
E-mail: amin.shokrollahi@epfl.ch

Algorithmics Laboratory (ALGO)
Ecole Polytechnique Federale de Lausanne (EPFL)

October 8, 2012

1 Summary

In the pas two weeks, Mr. Amin Karbasi and I have been busy working on proper models to apply our method to a database of natural images. We have been playing with various combinations of feature learning stages as well as non-linear transformations. In this report, we will explain two models, one that works and one that fails. However, the one that fails could be modified so that it also work. We will explain this modification and discuss problems in which these models can be useful.

2 Applying our model to datasets of natural images

To apply our neural associative algorithm to any dataset, we need inputs to fulfill the following properties:

1. They (almost) form a subspace so that the dual vectors can be learned using our neural learning algorithm.

2. The dual vectors are rather sparse. Not all datasets lead to sparse dual vectors.

For synthetic datasets, inputs directly satisfy the above properties (we make them in such a way!). However, for natural stimuli, it is not usually the case and one has to find a *transform* to map the input stimuli to another space where the new transformed vectors (almost) form a subspace. The mapping can be non-linear. Depending on the required task, the mapping can also be irreversible, i.e. one can not get back to the original stimuli from the mapped sequences. For dataset of spoken English words, one notices that a simple FFT is sufficient as a mapping.

For dataset of natural images, we have been playing around with different mappings. So far, we have found a mapping that is irreversible, i.e. one can not reconstruct the input image from the features. The model, shown in Figure 1, is inspired from the work of [1] and have multiple non-linear transforms.

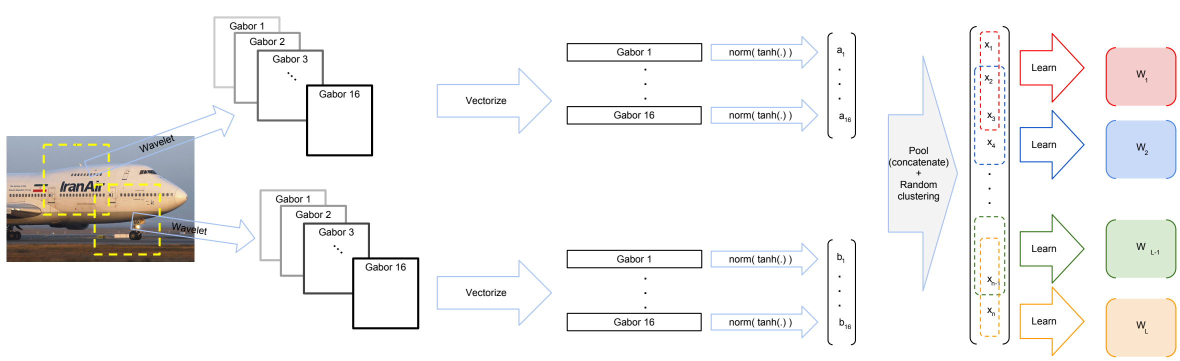

Figure 1: An example of irreversible feature mapping. The mapping is irreversible because of the tanh, |.| and *norm* functions.

In words, the proposed model is composed of the following stages:

1. We divide each image into *overlapping* patches.

2. We apply the Gabor wavelet to each patch by computing the 2D convolution of 16 Gabor filters and the image. Each of Gabor filters is responsible for oriented edge detection, with orientations varying from 0 to 180 degrees in 16 steps.

3. We then vectorize each of Gabor filter outputs for each patch and apply the function $|\tanh(.)|$ to all entries of each vector. Then, the norm of the resulting vector is calculated. This way, we get a 16-tuple for each patch of an input image.

4. We gather all these 16-tuples and concatenate them to get a very large vector v of size $16N$, where N is the total number of patches.

5. We define *overlapping random* clusters over this large vector and apply the learning algorithm to each cluster, to get the dual weight matrices $W_1, \dots, W_L$, where L is the number of clusters.

This approach has been tested and works quite nicely. Note that the whole system is irreversible in the sense that we can not reconstruct the original image from the entries of vector v.

We have tried to use this system for object classification. But this approach has failed for the following reason: for classification purposes, we must first learn L matrices for each class i, $W_1^{(i)}, \dots, W_L^{(i)}$, from the training set $\mathcal{X}_i$. Thus, for any pattern in $x \in \mathcal{X}_i$ we have $W_j^{(i)} \cdot x = 0$. Now for any pattern y in the *test* set of class i, $\mathcal{Y}_i$, we must also get $W_j^{(i)} \cdot y = 0$. However, this equality does not hold for our algorithm.

Therefore, as it stands, the current system can be used for denoising features of an image. This property might still be helpful for image denoising or object recognition purposes.

2.1 A failed variants

We have also came across various models that failed either because the subpatterns did not form a subspace, or because the resulting weight matrices were not sparse. However, a promising example which satisfied the above two requirement failed for another reason, explained in the following. The model is shown in Figure 2.

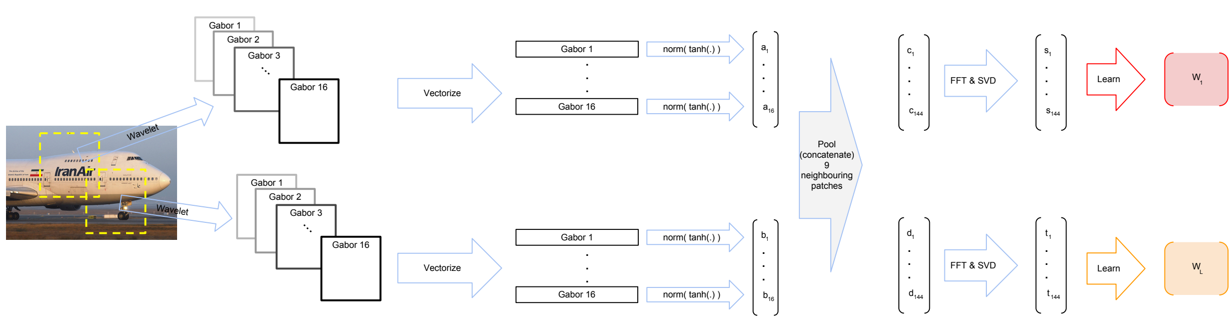

Figure 2: An exmaple of a failed irreversible feature mapping.

In words, the proposed model is composed of the following stages:

1. We divide each image into *overlapping* patches.

2. We apply the Gabor wavelet to each patch by computing the 2D convolution of 16 Gabor filters and the image. Each of Gabor filters is responsible for oriented edge detection, with orientations varying from 0 to 180 degrees in 16 steps.

3. We then vectorize each of Gabor filter outputs for each patch and apply the function $|\tanh(.)|$ to all entries of each vector. Then, the norm of the resulting vector is calculated. This way, we get a 16-tuple for each patch of an input image.

4. For each patch, we concatenate its 16-tuple with that of its 8 neighboring patches to get a new vector of size $9 * 16 = 144$.

5. These 144-tuples are then stored in a matrix A in which each row corresponds to one input image.

6. We apply SVD to this matrix and keep the scores, i.e. we get $A = U\Sigma V$ with SVD and keep $B = \Sigma V = U^\top A$ as the input matrix to the learning phase.

7. The traditional learning phase is applied to each of the matrices B to learn clusters $W_1, \dots, W_L$.

The reason that this model did not work was the fact that the weight matrices were not overlapping! Therefore, we could not use our recall algorithm over this system. This was due to the a mistake in the overall layout and can be easily fixed by applying a random clustering stage, similar to the one shown in Figure 1, and then apply the learning algorithm to these clusters instead of the output of the SVD for each patch. We will consider this model in our future works.

3 Conclusions and future works

So far, we have found an irreversible model that seems to work for denoising features of images. We will continue working to find a model that can be reversed, i.e. used to reconstruct images from the features as well. We might also consider the modified version of the model shown in Figure 2 for denoising image features, or even modifying it further to get a system that is reversible.

References

[1] K. Jarrett, K., Kavukcuoglu, M. A., Ranzato, Y., LeCun, *What is the best multi-stage architecture for object recognition?*, Proc. IEEE Int. Conf. Computer Vision, 2009, pp. 2146-2153.