Scientific Experimentation and Evaluation

Assignment 9 - MNIST

Nitish Koripalli and Senga Ndimubanzi Boris

June 12, 2016

CONTENTS

1 Abstract 3

2 Description of Methods 3

2.1 Preprocessing 3

2.1.1 Image Resizing 3

2.1.2 Patch Extraction 4

2.1.3 Normalization 5

2.1.4 Whitening 6

2.2 Feature Extraction 7

2.2.1 Dictionary Learning 7

2.2.2 Sparse Encoding 8

2.2.3 Max Pooling 8

2.3 Classification 9

3 Procedure 9

4 Parameters 9

5 Results 10

6 Comparison with Other Alogorithms 10

7 Limitations 10

1 Abstract

The MNIST dataset contains 70000 images of 10 image classes of digits from 0 to 9. It is a popular starting point for working on image classification since the dataset is basic, gray scale, cropped, centered and it is very well documented in the scientific community. We apply the technique of Sparse Coding for extracting features and using SVM for classification.

2 Description of Methods

We will discuss the methods used for preprocessing, feature extraction and classification and the motivations behind each of the methods.

2.1 Preprocessing

There are four important preprocessing steps which are:

- Image resizing
- Patch extraction
- Normalization
- Whitening

2.1.1 Image Resizing

Image resizing has the following properties:

- Reducing the size of data.
- Spatial scaling of image characteristics. NOTE - Feature extraction using sparse coding is scale invariant when the entire dataset is resized.

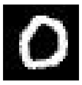

Figure 1: *Original image of size 28x28 pixels.*

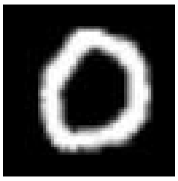

Figure 2: *Image resized to 64x64 pixels to scale features.*

2.1.2 Patch Extraction

Patch extraction has the following properties.

- Patches are mainly used to extract localized features.
- Patch sizes have a huge impact on the dimensionality of the features.

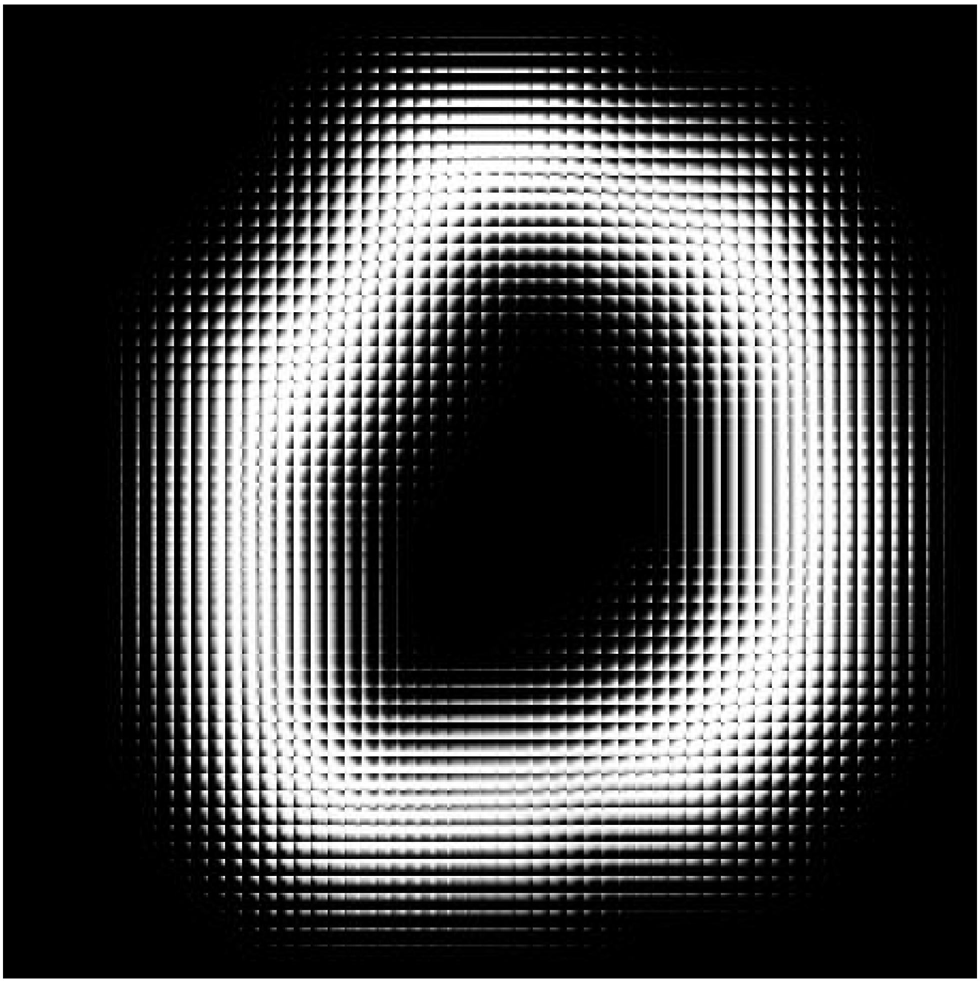

Figure 3: *A montage of patches of size 8x8 pixels are extracted. A total of 3249 patches are extracted for one 64x64 pixels image.*

2.1.3 Normalization

Normalization has the following properties.[1]

- Standardize the range of independent variables or features of data to fall in range of some objective function such as tanh, sigmoid etc.
- Gradient descent converges much faster with feature scaling (normalization) than without it.

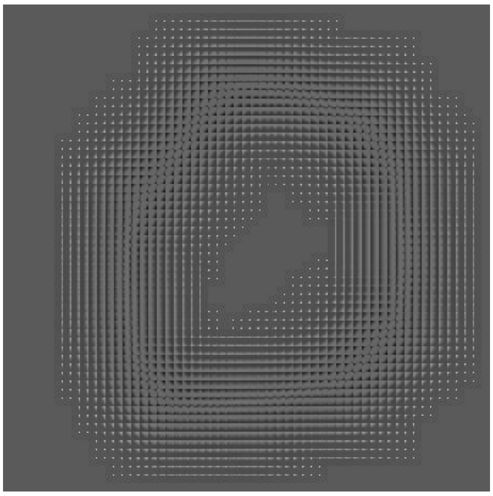

Figure 4: *A montage of normalized patches of patches from figure 3.*

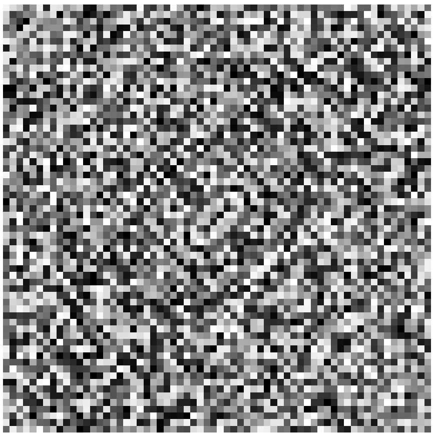

Figure 5: *The covariance matrix of normalized patches of figure 4. showing correlations or redundancies among the features.*

2.1.4 Whitening

- Raw input data is usually redundant therefore adjacent pixels are correlated which causes optimization algorithms to take longer for convergence.
- The aim is to have features be decorrelated with each other but have the same variance for all the images. [2]
- Two kinds of whitening are PCA whitening and ZCA whitening.

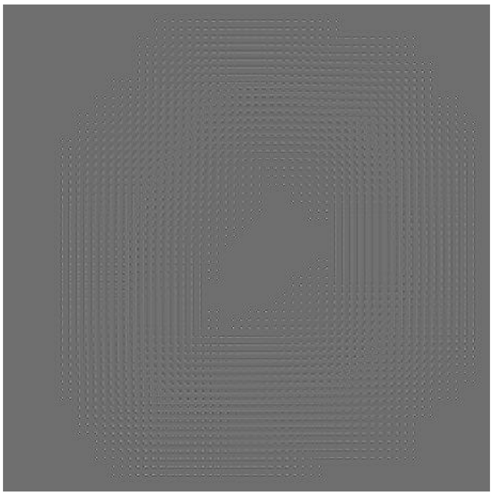

Figure 6: *A montage of whitened patches of normalized patches from figure 3.*

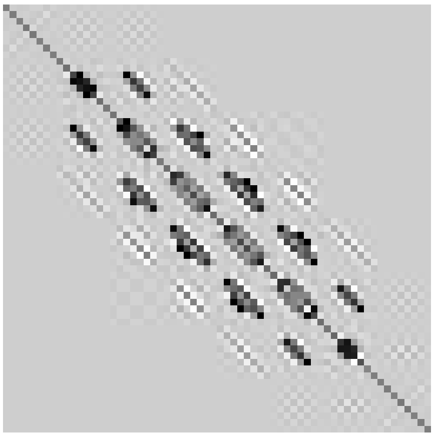

Figure 7: *The covariance matrix of whitened patches showing decorrelation i.e. only the diagonal is active. Some of the correlation may be due to the regularization factor which prevents division by zero. Regularization factor is usually 1e-5.*

2.2 Feature Extraction

Feature extraction involves the following:

- Dictionary learning
- Sparse encoding
- Max pooling

2.2.1 Dictionary Learning

Dictionary Learning has the following properties:

- To learn of a set of basis functions / basic features from the training data.

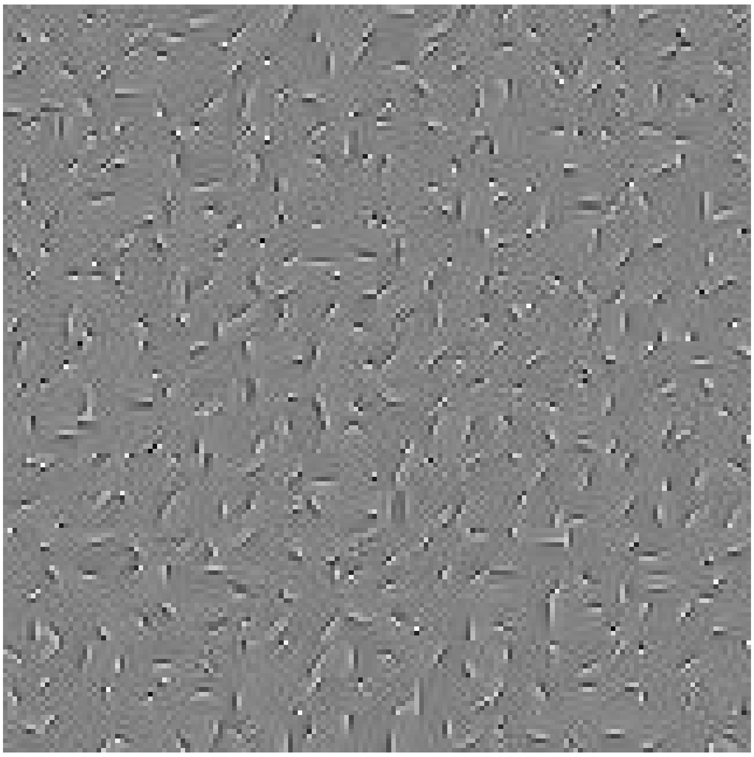

Figure 8: *Dictionary montage of 400 dictionary-elements/basis-functions with each element being 8x8 pixels.*

2.2.2 Sparse Encoding

Sparse encoding has the following properties:

- Encodes each preprocessed patch with a weighted sum of dictionary elements.
- The encoding is sparse i.e. most of the weights are are zeros.
- A common encoding algorithm is ’lasso’.

2.2.3 Max Pooling

Max pooling has the following properties:

- It selects the maximum weights per dimension of encodings in a window.
- It reduces dimensionality of the features by choosing only one pooled encoding over all other encodings in a window.
- Since it is a non-overlapping window, prominent weights are not chosen multiple times.
- Pooling adds a property of localization of features due to the windows.

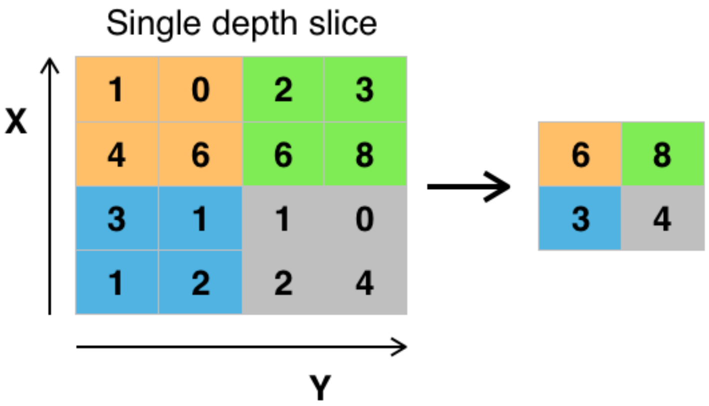

Figure 9: *An illustration of max pooling for one slice/dimension of the encoding vector. Source : https://en.wikipedia.org/wiki/Convolutional_neural_network*

2.3 Classification

We use a linear SVM classifier to classify the images with a slack factor C = 0.1.

3 Procedure

1. Obtain dataset
2. Split dataset into training and test set
3. Resize images
4. Extract patches
5. Normalize patches
6. Whiten patches
7. Learn dictionary on training set
8. Encode the training patches on the dictionary
9. Train SVM classifier using the training encoding
10. Encode the test patches on the dictionary
11. Test the classification rate using the test encoding
12. Repeat three times from step 2, as per 3-fold cross validation

4 Parameters

- image_resize : (28,28) pixels
- patch_size : (8,8) pixels
- pool_size : (7,7) pixels
- svm_kernel : linear
- svm_slack_coefficient (C) : 0.1
- encoding_sparseness : 4 / 400 (non zero elements only)
- encoding_algorithm : lasso

5 Results

Validation of model was done using 3-fold cross validation:

1. Run 1 : 90.60 % accuracy
2. Run 2 : 88.69 % accuracy
3. Run 3 : 90.47 % accuracy

Average accuracy : 89.92 %

6 Comparison with Other Alogorithms

7 Limitations

- Patch extraction consumes a lot memory.
- Meaningful dictionary learning takes a lot of time.
- Sparse encoding is not fully supported for GPUs unlike neural networks, which means we need a lot of CPU cores.

References

[1] https://en.wikipedia.org/wiki/Feature_scaling

[2] http://ufldl.stanford.edu/tutorial/unsupervised/PCAWhitening/

[3] https://en.wikipedia.org/wiki/Convolutional_neural_network