

Deep Representation Learning with Target Coding Supplementary Material

Shuo Yang¹, Ping Luo^{2,1}, Chen Change Loy^{1,2}, Kenneth W. Shum¹, and Xiaoou Tang^{1,2}

¹Department of Information Engineering, The Chinese University of Hong Kong

²Shenzhen Key Lab of CVPR, Shenzhen Institutes of Advanced Technology,
Chinese Academy of Sciences, Shenzhen, China

{ys014,pluo,ccloy,xtang}@ie.cuhk.edu.hk, {wkshum}@inc.cuhk.edu.hk

Abstract

This document provides the details of the CNN structure and parameters used in our paper, to facilitate future reimplementing of our work. We also describe our data processing methods for each dataset. In addition, the Hadamard Code with different lengths is attached at end.

CNN Structure and Learning Parameters

STL-10, MNIST

We used the same network structure on STL-10, MNIST following existing works (Srivastava and Salakhutdinov 2013) and removed two fully-connected layers. The deep network has three convolutional layers each followed by a max pooling layer. The last hidden layer is a locally-connected layer. The dropout rate for the first two convolutional layers is 0.25. For the last convolutional layer and locally-connected layer the dropout rate is 0.4 or 0.5. In fact, there are no significant difference between them.

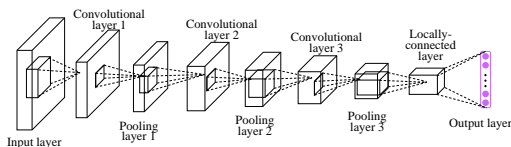


Figure 1: The network structure of STL-10, MNIST.

CIFAR-100

The CIFAR-100 deep network following existing works (Srivastava and Salakhutdinov 2013) has three convolutional layers, three max pooling layer, one local connected layer and two fully-connected layers. The dropout rate for the first two convolutional layers is 0.25. For the last convolutional layer, local connected layer and two fully-connected layers the dropout rate is 0.5.

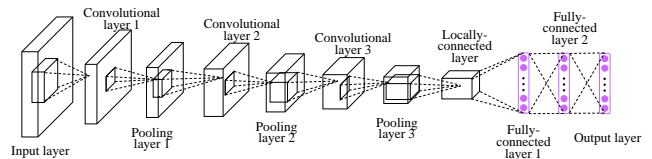


Figure 2: The network structure of CIFAR-100.

ImageNet-2012

The ImageNet-2012 deep network following existing works (Krizhevsky, Sutskever, and Hinton 2012; Zeiler and Fergus 2014) has five convolutional layers, three max pooling layers and two fully-connected layers. The dropout rate for the two fully-connected layers the dropout rate is 0.5.

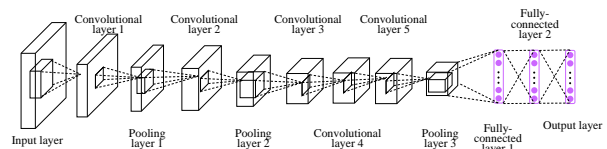


Figure 3: The network structure of ImageNet-2012.

Datasets

Overview

MNIST Following Larochelle et al (Larochelle et al. 2007), we use the MNIST extension dataset which is generated with additional factors of variation to the MNIST images. Specifically, the MNIST-rot-back-image generate with two procedures. The digits were first rotated by an angle generated uniformly between 0 and 2π radians, then a random patch from a black and white image was used as the background for the digit image. The MNIST-rot-back-image consists of 12000 training images and 50000 testing images.

STL-10 STL-10 dataset is an image recognition dataset with 96×96 natural RGB images in 10-classes. Compared with CIFAR-10 dataset STL-10 dataset has less labeled

data, it has 5000 training images with 500 images per class and 8000 testing images with 800 per class. It has 100,000 unlabeled data for the unsupervised learning.

CIFAR-100 The CIFAR-100 dataset consists of 60000 natural 32×32 RGB images belonging to 100 classes. This dataset contains 50000 training images and 10000 test images. There are 500 training images and 100 testing images per class. 100 classes are divided into 20 groups of 5 classes each. For example, the superclass flowers contains orchids, poppies, roses, sunflowers, tulips, and the superclass insects contains bee, beetle, butterfly, caterpillar, cockroach.

ImageNet-2012 The ImageNet-2012 dataset consists of 1.35 million natural RGB images belonging to 1000 classes. This dataset contains 1.2 million training images and 150,000 validation test images. There are about 1000 training images and 100 testing images per class. The validation and test data are not contained in the training data.

Data preprocessing

All the processed data were used on 1-of- k and Hadamard Code based CNN model.

MNIST No preprocessing.

STL-10 Deep model training following the pre-defined STL-10 partition is infeasible due to the small number of samples in each fold. Therefore, deep learning studies (Coates and Ng 2011) tend to adopt different protocols for STL10. Note that our main goal is to compare the effectiveness of Hadamard coding vs. 1-of-K coding. We trained both coding schemes using the same data partition. Following the augmentation strategy in Hinton’s (Krizhevsky, Sutskever, and Hinton 2012) work, we first resized the original images to 64×64 then randomly cropped 48×48 patches 20 times from resized images and test on the center 48×48 patch. We only used the labeled data and ignored the 100,000 items of unlabeled data.

CIFAR-100 Following the same process with STL-10, we first resized the original images to 40×40 then randomly cropped 32×32 patches 15 times from the resized images and test on the center 32×32 patch.

ImageNet-2012 Following the same process with Hinton’s (Krizhevsky, Sutskever, and Hinton 2012) work, we first resized the original images to 256×256 then cropped 224×224 patches from four corners and the center of the resized images then flipped the cropped patches.

Hadamard Code(HC) Codewords

HC-63, HC-127, HC-255 for 10 classes

For HC-63 every two rows in the table represent one codeword. Similar with HC-63, a codeword in HC-127 and HC-255 has four rows and eight rows respectively.

Table 7: HC-63.

111010101100110111011010010011100010
111100101000110000100000111
000001111110101011001101110110100100
111000101111001010001100001
111100101000110000100000111111010101
100110111011010010011100010
000100000111111010101100110111011010
010011100010111100101000110
000010000011111101010110011011101101
001001110001011110010100011
01000001111101010110011011101101001
001110001011110010100011000
111111010101100110111011010010011100
010111100101000110000100000
111001010001100001000001111110101011
001101110110100100111000101
101011001101110110100100111000101111
001010001100001000001111110
000111111010101100110111011010010011
100010111100101000110000100

Table 8: HC-127.

```

11001010101111110000001000001100001
010001111001000101100111010100111110
100001110001001001101101011011110110
    0011010010111011100
111100000010000011000010100011110010
001011001110101001111101000011100010
010011011010110111101100011010010111
    0111001100101010111
101000111100100010110011101010011111
010000111000100100110110101101111011
000110100101110111001100101010111111
    1000000100000110000
101011111110000001000001100001010001
111001000101100111010100111110100001
11000100100110110101101110110001101
    00101110111001100110010
000110100101110111001100101010111111
100000010000011000010100011110010001
011001110101001111101000011100010010
    0110110101101111011
011101010011111010000111000100100110
110101101111011000110100101110111001
100101010111111100000010000011000010
    1000111100100010110
10100101110111001100101010111111000
000100000110000101000111100100010110
011101010011111010000111000100100110
    110101101111011001
001000001100001010001111001000101100
111010100111110100001110001001001101
101011011110110001101001011101110011
    0010101011111110000
011111110000001000001100001010001111
001000101100111010100111110100001110
001001001101101011011110110001101001
    0111011100110010101
011100110010101011111110000001000001
100001010001111001000101100111010100
111110100001110001001001101101011011
    1101100011010010111

```

Table 9: HC-255-1.

```

110001110000110000011101100001010110
010011100111010101111111101101100111
10001101011100100001111011101110100
000100000010110111110011011100010111
010011001010101001001000101000010011
010001111101011010010100111111000000
011011010100010010111100101100010001
    100
110111011110100000100000010110111110
011011100010111010011001010101001001
000101000010011010001111101011010010
100111111000000011011010100010010111
100101100010001100110001110000110000
011101100001010110010011100111010101
111111101101100111100011010111001000
    011
010011001010101001001000101000010011
010001111101011010010100111111000000
011011010100010010111100101100010001
100110001110000110000011101100001010
110010011100111010101111111101101100
111100011010111001000011110111011110
100000100000010110111110011011100010
    111
100010010111100101100010001100110001
110000110000011101100001010110010011
100111010101111111101101100111100011
010111001000011110111011110100000100
000010110111110011011100010111010011
001010101001001000101000010011010001
111101011010010100111111000000011011
    010
0011111010110100101001111111000000011
0110101000100101111100101100010001100
110001110000110000011101100001010110
010011100111010101111111101101100111
10001101011100100001111011101110100
000100000010110111110011011100010111
010011001010101001001000101000010011
    010

```

Table 10: HC-255-2.

010111010011001010101001001000101000
0100110100011111010111010010100111111
000000011011010100010010111100101100
010001100110001110000110000011101100
00101011001001110011101010111111101
101100111100011010111001000011110111
011110100000100000010110111110011011
100
111110110110011110001101011100100001
111011101111010000010000001011011111
001101110001011101001100101010100100
100010100001001101000111110101101001
010011111100000001101101010001001011
110010110001000110011000111000011000
001110110000101011001001110011101010
111
111010101111111101101100111100011010
111001000011110111011110100000100000
010110111110011011100010111010011001
010101001001000101000010011010001111
101011010010100111111000000011011010
100010010111100101100010001100110001
110000110000011101100001010110010011
100
011100001100000111011000010101100100
111001110101011111111011011001111000
110101110010000111101110111101000001
000000101101111100110111000101110100
110010101010010010001010000100110100
011111010110100101001111110000000110
110101000100101111001011000100011001
100
011010100010010111100101100010001100
110001110000110000011101100001010110
010011100111010101111111101101100111
100011010111001000011110111011110100
000100000010110111110011011100010111
010011001010101001001000101000010011
010001111101011010010100111111000000
011

References

- Coates, A., and Ng, A. Y. 2011. Selecting receptive fields in deep networks. In *NIPS*.
- Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet classification with deep convolutional neural networks. In *NIPS*.
- Larochelle, H.; Erhan, D.; Courville, A.; Bergstra, J.; and Bengio, Y. 2007. An empirical evaluation of deep architectures on problems with many factors of variation. In *ICML*, 473–480.
- Srivastava, N., and Salakhutdinov, R. 2013. Discriminative transfer learning with tree-based priors. In *NIPS*.
- Zeiler, M., and Fergus, R. 2014. Visualizing and understanding convolutional neural networks. In *ECCV*.

