CS682 Final Write Up

Non-Convolution But Locally Connected Neural Network

September 28, 2023

1 Abstract

Locally Connected Neural Network (LCN) is contrast to the standard Convolutional Neural Network (CNN). LCN forms local connection not by convolving the filter but by penalizing the score by distance. In this way, the influence of the edge weights will be minimized while the influence of the center weights are more emphasized. LCN achieves a test accuracy of 49.2% and validation accuracy of 50.5% in CIFAR-10, and a validation accuracy of 24.3% in CIFAR-100 and 37.3% in CIFAR-100 superclass.

2 Introduction

In human vision, each neuron in primary visual cortex has a local receptive field and can detect certain specific orientations i.e humans apply different filters to different region of the image, while in computer vision, convolutional neural networks (CNN) handles image by convolving filters over all spatial locations to form activation maps. The difference between human vision and computer vision has inspired us to construct a non-convolution but locally connected neural network (LCN) to resemble how humans process images. Instead of sliding filters over the input image to form local connections, LCN form local connections by penalizing each score by the weight and the pixel's distance. The distance of each pair of weight element and pixel is decided by the center distance, a hyperparmeter controlling the spatial distance between the weight matrix and the image matrix, and by input distance, the Euclidean distance between the weight element and the pixel in 2D plane.

The paper is structured as follows. Section 3 presents the background and the related work. Section 4 presents our approach to the problem, describing the framework of our project. Section 5 describes the experiment we have conducted, datasets we have used and the qualitative and quantitative evaluations of our result, Section 6 concludes the paper, and Section 7 lists supplementary material.

3 Background/ Related Work

- ImageNet Classification with Deep Convolutional Neural Networks [1]

  The convolutional neural network in the paper achieved top-1 and top-5 error rates of 37.5% and 17.0% in the ImageNet LSVRC-2010 contest. The network contains eight layers with weight, with five convolutional and the remaining three fully-connected. The output of the last fully-connected layer is fed to a 1000-way softmax which produces a distribution over the 1000 class labels. Some novel features of the network's architecture includes ReLU Nonlinearity, Local Response Normalization and Overlapping Pooling.

- Best practices for convolutional neural networks applied to visual document analysis [2]

  The paper describes a set of best practices on document analysis, including expanding training set

by adding a new form of distorted data, and adopting convolutional neural netowrk instead of fully connected networks for visusal document tasks.

- Locally connected neural networks for image recognition [3]

  Locally connected neural networks (LCN) is defined as a non-weight-sharing version of convolutional neural network in this paper. It argues that weight sharing property of CNN reduces number of parameters, but the property also limits the learning capacity of a layer due to fewer parameters. LCN in this paper keeps the number of output maps of each layer same and only remove weight-sharing constraint in CNNs. Thus, number of parameters in a layer increases without increasing number of output maps, capacity of the network also increased without increase in number of computations during inference as compared to weight-shared CNN. The implemenatation of a Partially-local connected network (P-LCN) VGG-16 which is hybrid of convolutional layers and Locally connected layers can achieve on average 2.0% accuracy gain over VGG-16 full convolutional network on CIFAR100 and 0.32% on CIFAR10.

4 Approach

The following steps are our approach for implementing a LCN layer

- Step1: Assign coordinates
  We first start with assigning coordinates to both the inputs and the weights, the rule is to have the center set to (0,0) and then assign the rest accordingly as shown in **Figure 1**. Note that every position of the weights is a hidden unit.

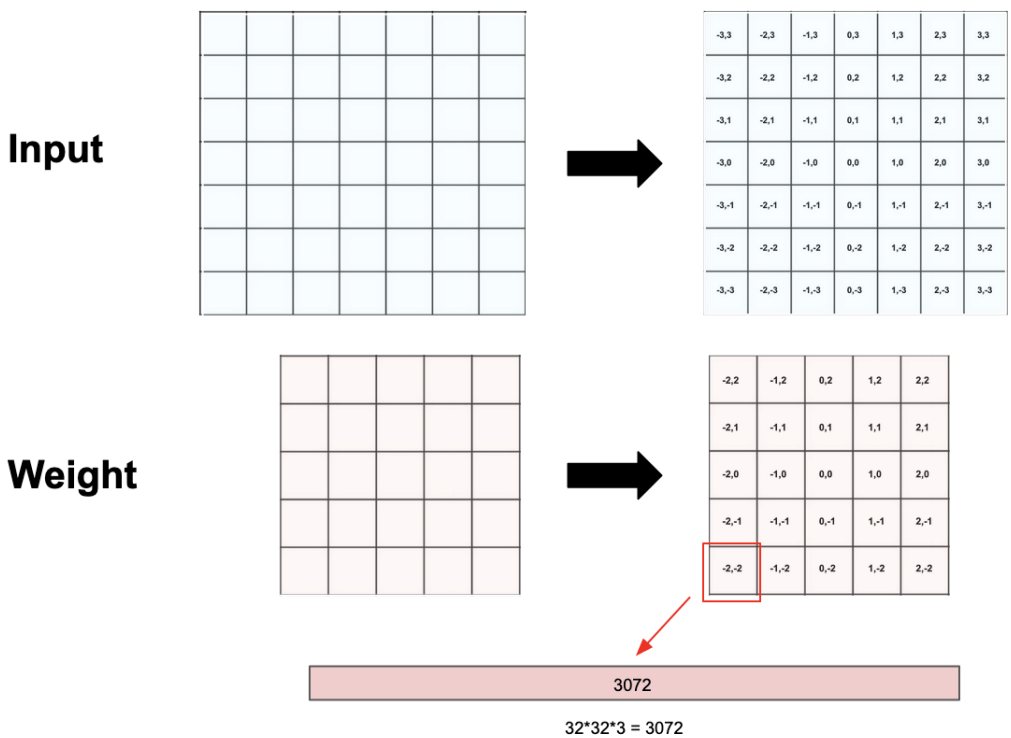

Figure 1: Assign coordinates

- Step2: Define Center Distance
  The center distance is a user-defined hyperparameter, it represents the distance between centers of the inputs and weights when aligned as shown in **Figure 2**.

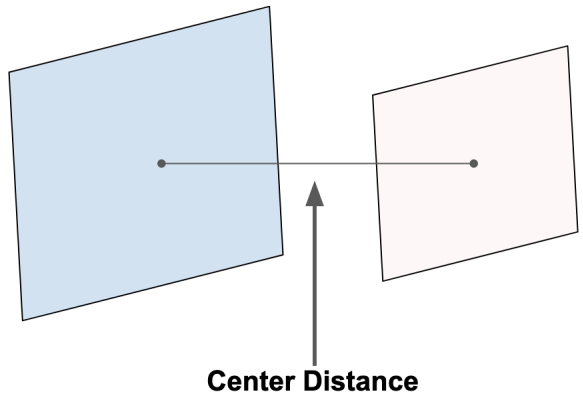

Figure 2: Define center distance

- Step3: Add Distance Penalty
  For every position in our activation map, we would calculate the square of the distance as the example shown in **Figure 3**. We then penalize the score accordingly by dividing the square of the distance.

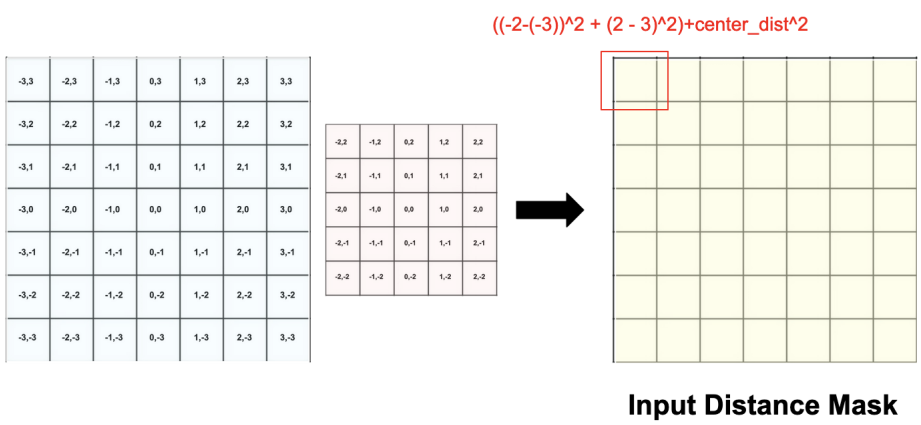

Figure 3: Calculate input distance mask

5 Experiment

- Check backward pass with numerical validation
  We use numerical gradient to validate our back-propagation, results shown in **Table 1**.

	error
dx	$2.9678e-08$
dw	$2.7108e-07$
db	$3.2756e-12$

Table 1: back-propagation numerical validation

- Speed test
  We compared our speed for forward and backward pass to the naive CNN we implemented in assignment 2, results shown in **Table 2**.

	CNN	LCN	speed up
forward	23.070871s	0.249119s	92.609825x
backward	36.127207s	0.353023s	102.336750x

Table 2: forward and backward speed comparison between LCN and CNN

- Overfit
  We use small size data to check if the our model could overfit, and as our results shown in **Figure 4**, it did.

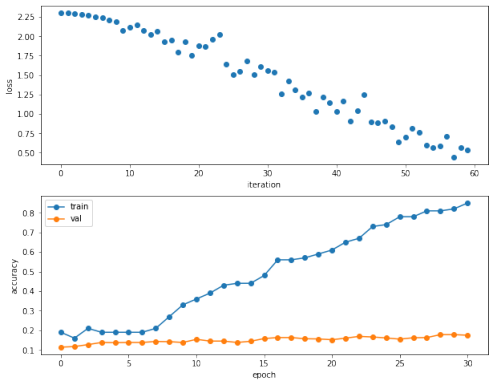

Figure 4: Overfitting to a small amount of data

- Tune hyperparameter

weight size= 7	validation accuracy	time
center_dist= 1	0.481	612.6s
center_dist= 5	0.489	607.6s
center_dist= 10	0.493	607.1s
center_dist= 15	0.505	607.0s
center_dist= 20	0.503	605.9s
center_dist= 25	0.480	604.9s
center_dist= 30	0.484	625.2s
center_dist= 35	0.494	624.6s
center_dist= 40	0.458	624.6s
center_dist= 45	0.446	624.3s
center_dist= 50	0.457	627.6s
center_dist= 55	0.447	626.8s

Table 3: Tune hyperparameter

  There are two hyperparameters defined in LCN, weight size and center distance. Weight size controls the size of the filter while center distance controls the spatial distance between the weight matrix and the input matrix. Through the use of the validation set, we have achieved a highest validation accuracy of 50.5% with weight size as 7 and center distance as 15 as shown in **Table 3**.

- Print out weights

Figure 5: weight size:7, center distance:1

Figure 6: weight size:7, center distance:15

Through the weight we printed in **Figure 5, 6, 7**, we can observe that weight with center distance 1 has many darkened pixels, which we suspect is the reason of the low accuracy. Also, we can observe that weight with center distance 100 has only around three variations in weight, which should be the reason of low accuracy.

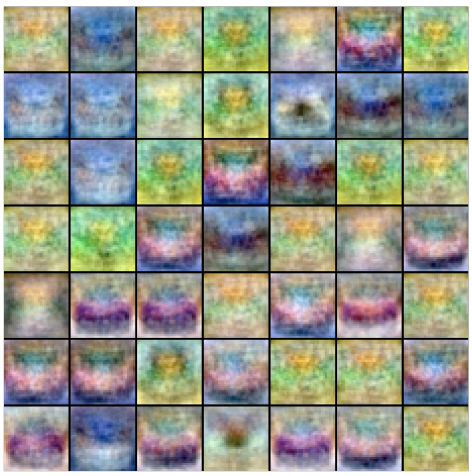

Figure 7: weight size:7, center distance:100

- Compare to CNN and FC

CIFAR-10	validation accuracy	test accuracy
LCN	0.505	0.492
CNN	0.489	0.463
FC	0.482	0.456

Table 4: validation and test accuracy compared to CNN and FC

LCN achieves a test accuracy of 49.2% and validation accuracy of 50.5% in CIFAR-10, which outperforms the test and validation accuracy of FC and CNN, which is shown in **Table 4**.

- CIFAR-100

	train accuracy	validation accuracy
CIFAR-100	0.322	0.243
CIFAR-100-superclass	0.462	0.373

Table 5: validation and test accuracy compared to CNN and FC

LCN achieves a validation accuracy of 24.3% in CIFAR-100 and 37.3% in CIFAR-100 superclass as shown in **Table 5**.

Figure 8: CIFAR-100, weights

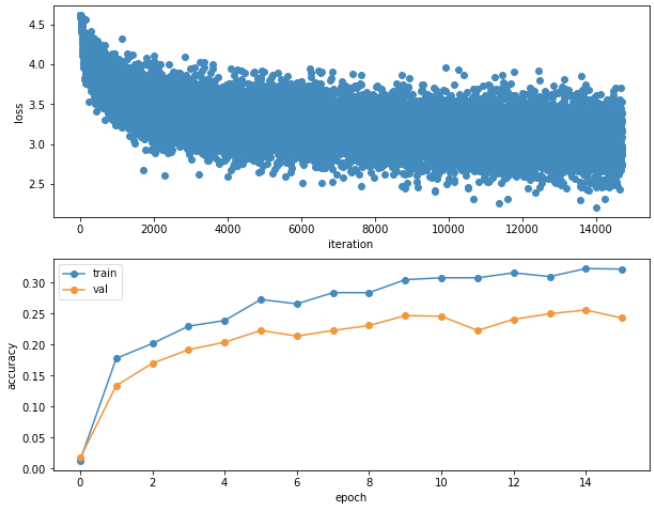

Figure 9: CIFAR-100, accuracy plot

6 Conclusion

weight size= 7	validation accuracy	test accuracy
center_dist= 15	0.505	0.492

Table 6: test accuracy for the best model

LCN forms local connections by penalizing distance and removes the need of convolving filters over the input data. Thus, LCN reduces the heavy computation needed for iterations of the standard CNN, and achieved a **92x** and **102x** speedup in forward pass and backward pass comparing to CNN. Also, LCN achieves a test accuracy of 49.2% and validation accuracy of 50.5% in CIFAR-10, which outperforms the test and validation accuracy of FC and CNN.

At this point, we finished the basic architecture of LCN. What we would like to do as a future improvement is to add a hyperparameter defining the spacing of the coordinator of each neighboring pixel/weights. Currently, the spacing of the coordinator between each element increases/decreases only by 1. Without the spacing hyperparameter, the center distance hyperparameter now seems to be too dominant to our results,

Figure 10: CIFAR-100-superclass, weights

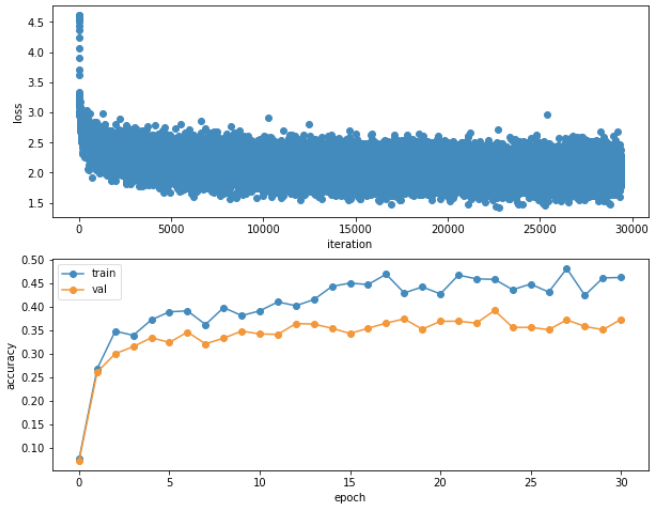

Figure 11: CIFAR-100-superclass, accuracy plot

and thus we think that addding this spacing hyperparameter to help spread out coordinators would be a great idea.

Some other future improvements including adding Dropout as a mechanism for regularization to prevent over-relying on specific features, inserting Batch Normalization layer after LCN layer in order to form deeper networks, and training Fully Connected Neural Network and Convolutional Neural Network with CIFAR-100 to compare results with LCN.

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems*, 25, 01 2012.

[2] Patrice Y Simard, David Steinkraus, John C Platt, et al. Best practices for convolutional neural networks applied to visual document analysis. In *Icdar*, volume 3, 2003.

[3] Shakti Nagnath Wadekar. Locally connected neural networks for image recognition, Dec 2019.

7 Supplementary Material

Figure 12: weight size:7, center distance:1

Figure 13: weight size:7, center distance:5

Figure 14: weight size:7, center distance:10

Figure 15: weight size:7, center distance:15

Figure 16: weight size:7, center distance:20

Figure 17: weight size:7, center distance:25

Figure 18: weight size:7, center distance:30

Figure 19: weight size:7, center distance:35

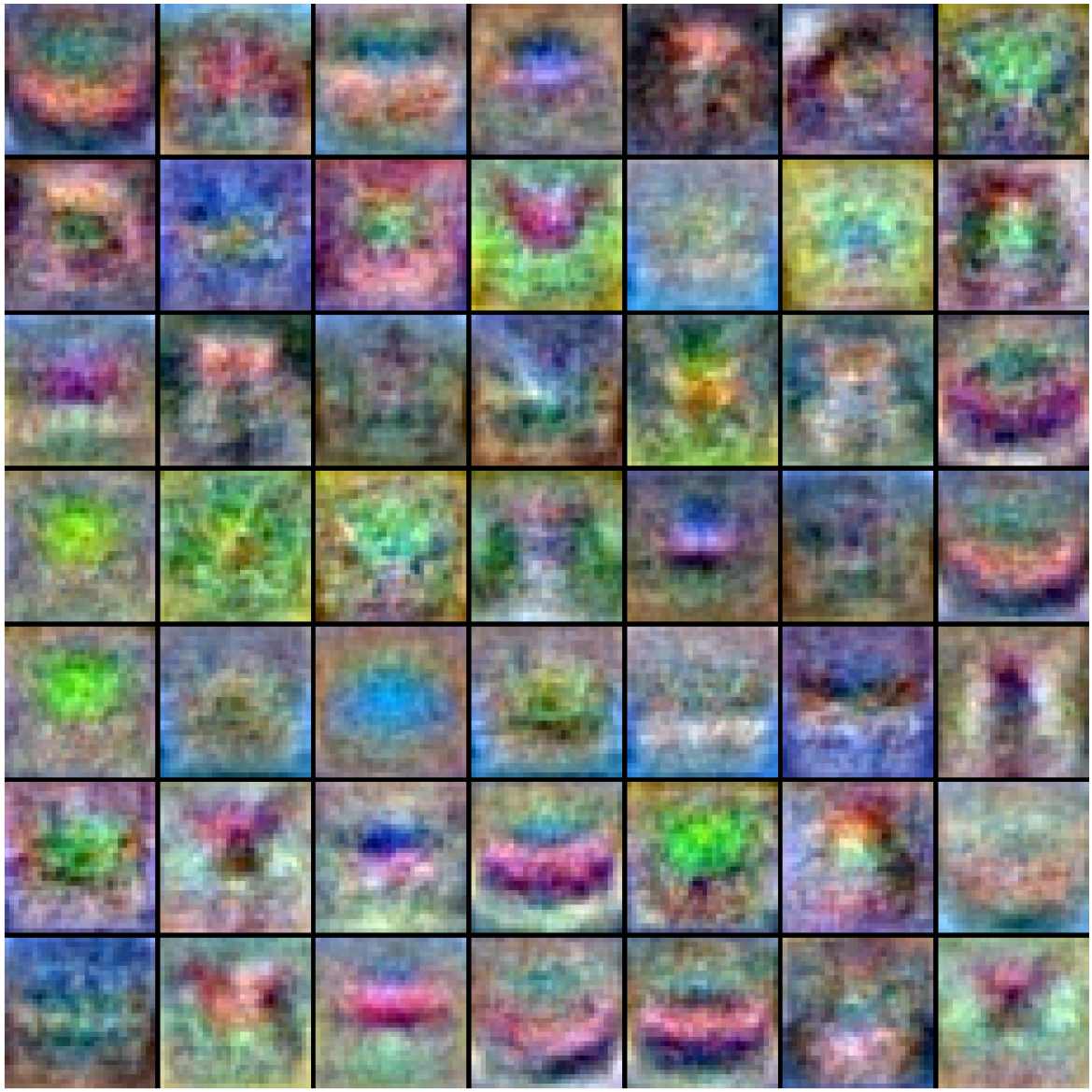

Figure 20: weight size:7, center distance:40

Figure 21: weight size:7, center distance:45

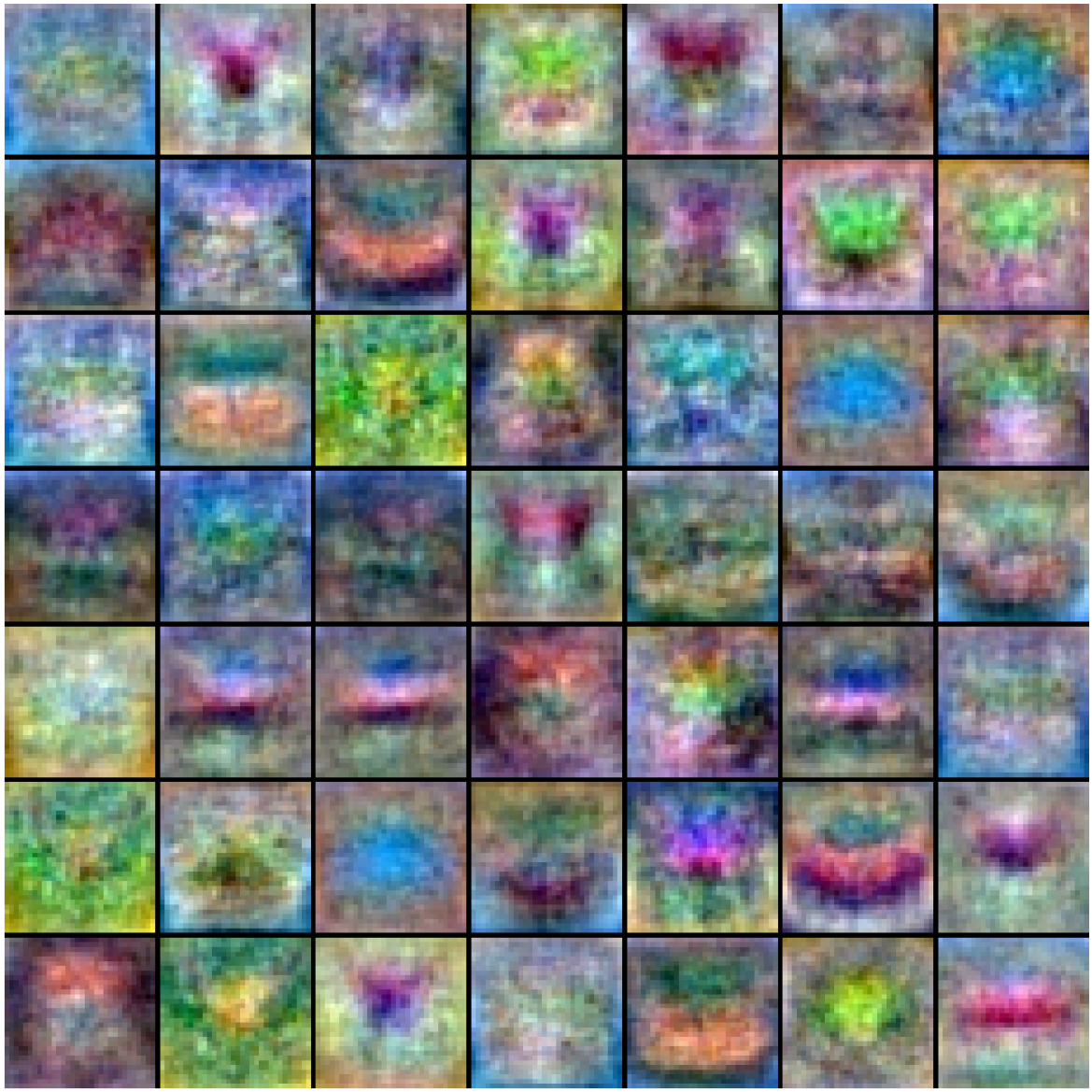

Figure 22: weight size:7, center distance:50

Figure 23: weight size:7, center distance:55