project-10

June 25, 2021

Contents

Chapter 1

Introduction

In this project, we implement `K-winners Take All` and `Lateral Inhibition` mechanisms; then train a simple Convolutional Spiking Neural Network (SNN) using MNIST and Caltech datasets. Multiple parameters' effect are tested on the training process.

When using `KWTA` only K neurons (winners) can spike for each input image, and these K spikes should be selected from different feature maps ($K \leq NumFilters$). Winner neurons activate the lateral inhibition and prevent adjacent neurons in other filters to spike.

The steps for the training are as follows:

1. Resize the images to the same shape
2. Apply a DoG filter on the images
3. Encode the images using Time2FirstSpike encoder
4. Apply Convolution on the encoded data
5. Update the weights using `STDP` rule modulated with `KWTA` and `Lateral Inhibition`.

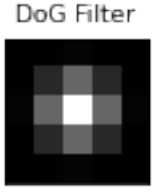
Note: A Gaussian kernel is used for Lateral Inhibition.

Chapter 2

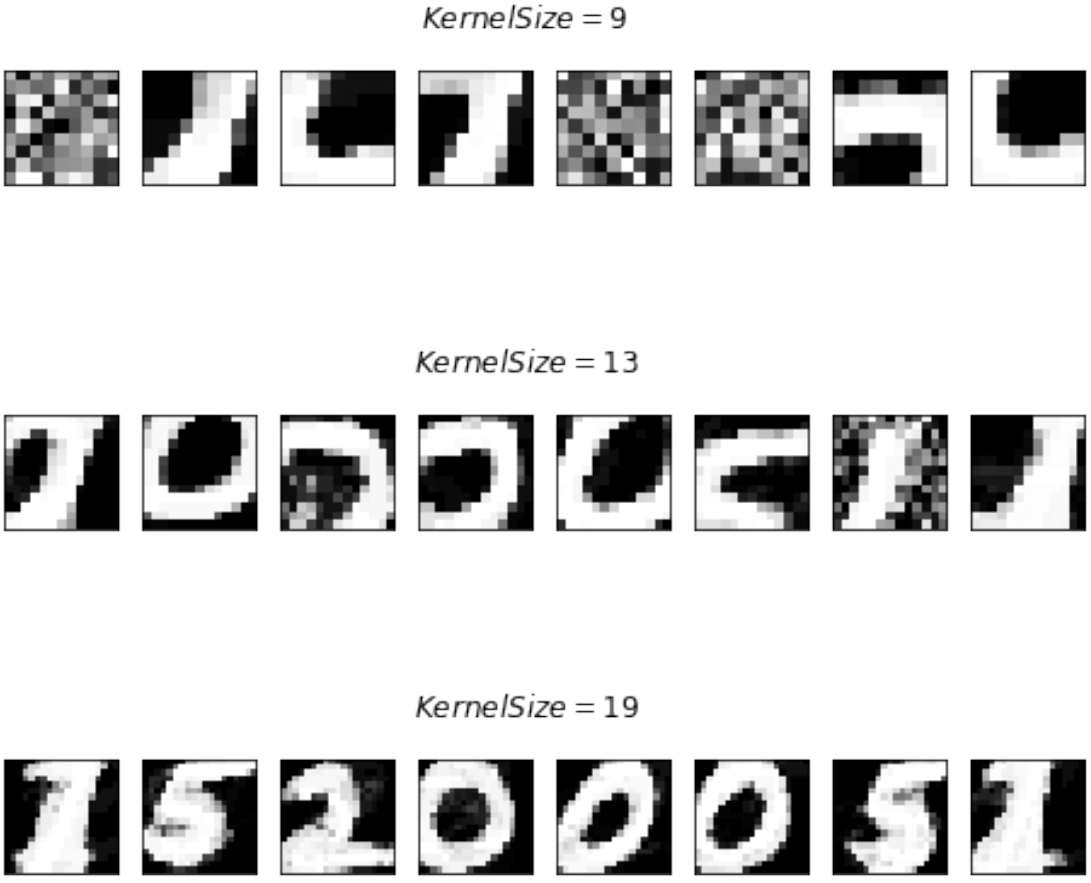
Part 1 (MNIST dataset)

We use only four digits $(0, 1, 2, 5)$ of this dataset to simplify the process.

Default Parameters:
Input and Learning Parameters:
$NumImages \approx 3000$
$ImageSize = 28 \times 28$
$FilterSize = 21 \times 21$
$EncodeTime = 5ms$
$NumFilters = 8$
$K = \frac{1}{4}NumFilter$
$LearningRate = [0.02, 0.2]$
$DoG_{KernelSize} = 5, DoG_{\sigma_1} = 1.5, DoG_{\sigma_2} = 15$
Lateral Inhibition Params:
$KernelSize = ConvolutionFilterSize$
$\sigma = \frac{1}{3}ConvolutionFilterSize$
$Multiplier = 15$
Convolution Layer Neuron Parameters:
$\tau = 10ms$
$\tau_s = 5ms$
$threshold = -57.5mv$
$u_{rest} = -60mv$
You can see the dog filter, filtering and encoding output below.



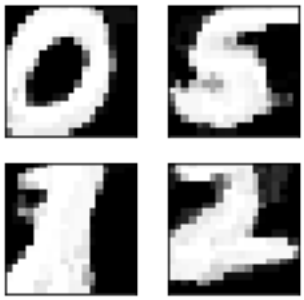
2.1 KernelSize



If KernelSize is smaller than the input images, the filters tend to learn parts of the input images like curves or oriented lines of digits. When $KernelSize = 19$, filters can learn the entire digits separately.

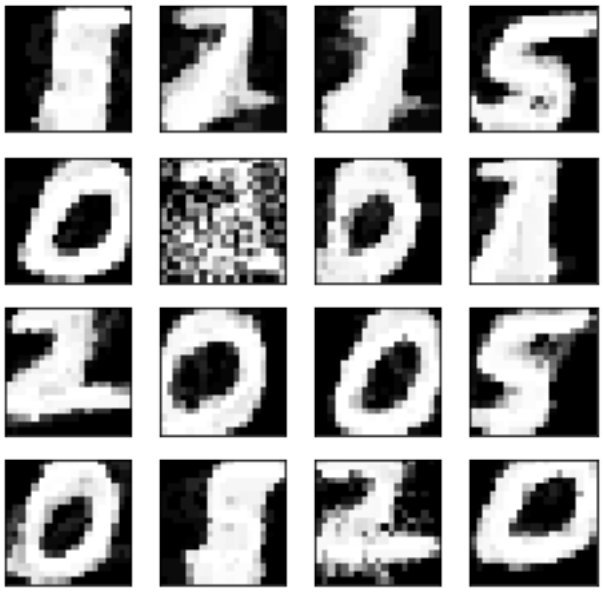
2.2 #Filters

$F = 4$



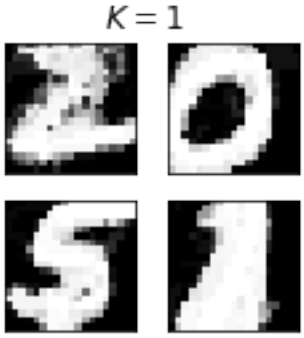
$F = 8$

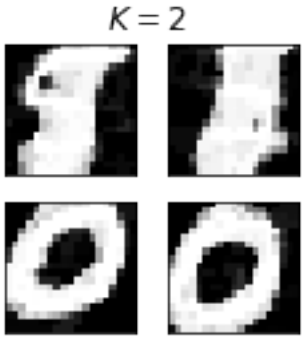
$F = 16$

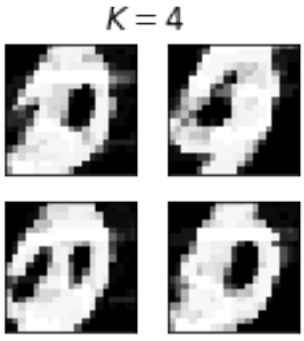


Since we only use four digits, 4 filters could be enough to learn them. By increasing the filters number, some filters might not learn anything, and some of them may learn shapes that are similar to two or more digits.

2.3 K



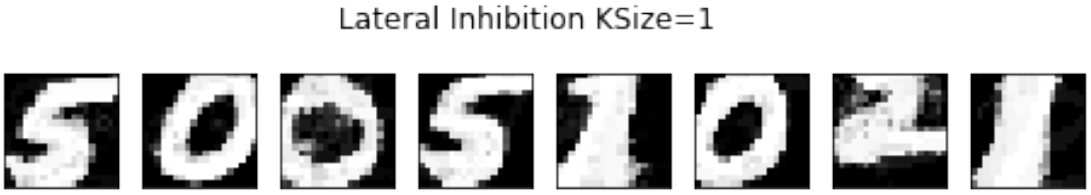




If $\frac{F}{K} = NumClasses$, every feature tends to learn one digit. But when $\frac{F}{K} \leq NumClasses$, filters tend to learn shared and/or mixed features of digits.

2.4 Lateral Inhibition

2.4.1 KernelSize

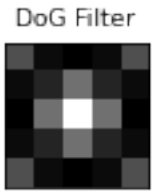


From the above figures we conclude that the Lateral Inhibition KernelSize does not change the learned weights meaningfully in our setting. A possible reason is that our encoding time is short (also the pixel values are almost binary) and often all winners are found in the same time step, so the inhibition will not get the chance to make its effect visible. Due to this reason other parameters of lateral inhibition possibly do not have any visible effect.

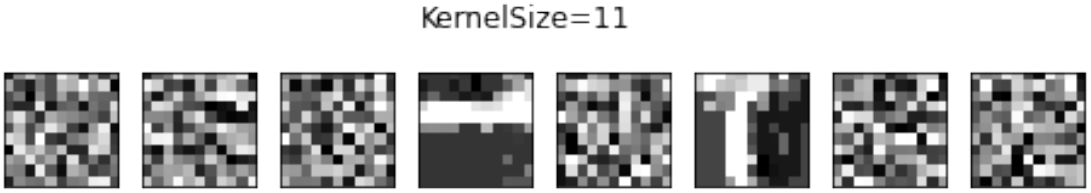
Chapter 3

Part 2 (Caltech dataset: Faces_easy)

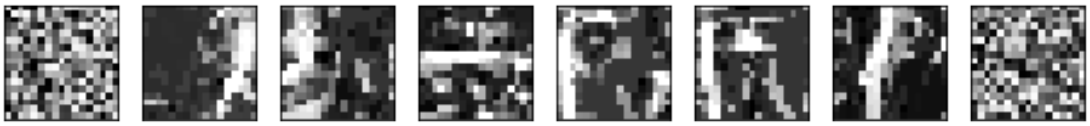
In this section we only provide some outputs without discussing the effect of parameters.
The images' size are 62×62.



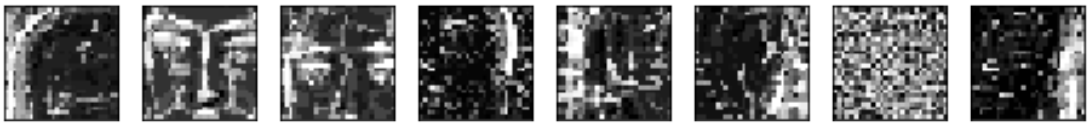
3.1 Parameter Set 1



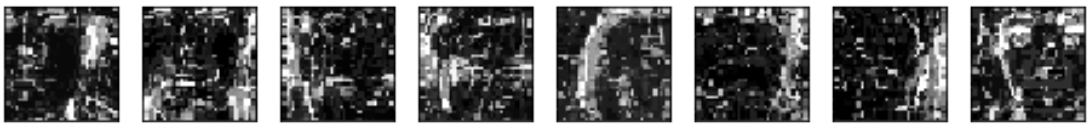
KernelSize=17



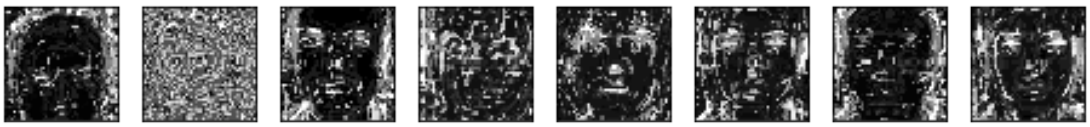
KernelSize=27



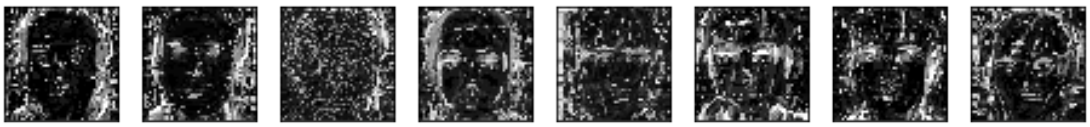
KernelSize=37



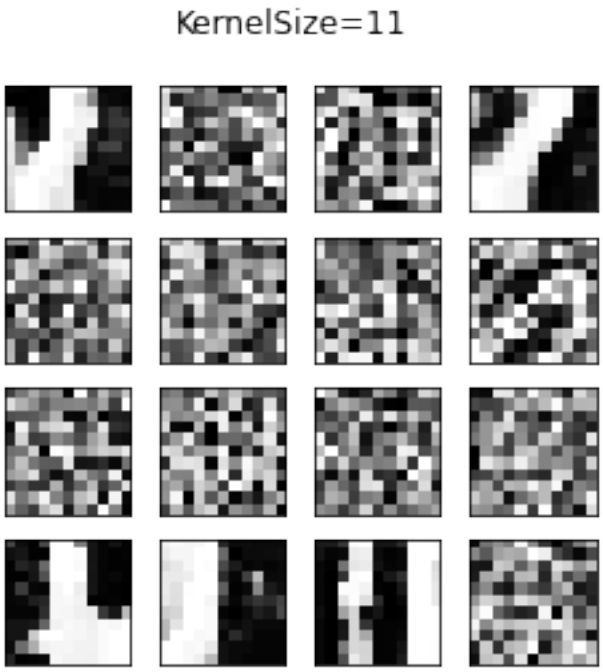
KernelSize=47

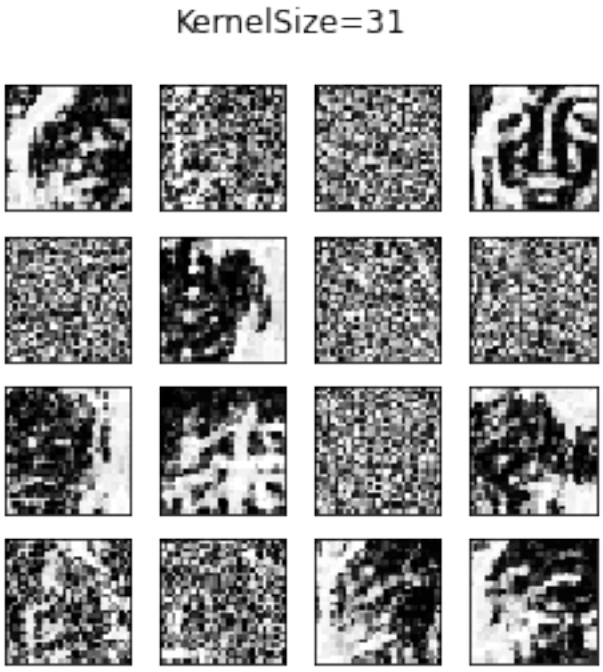


KernelSize=55

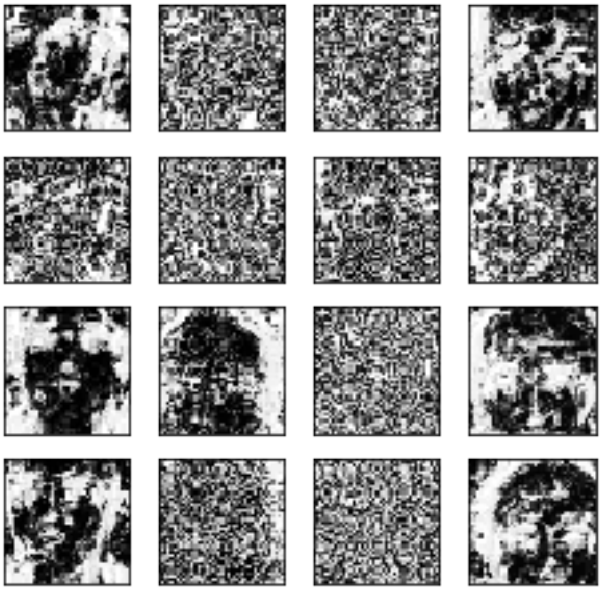


3.2 Parameter Set 2

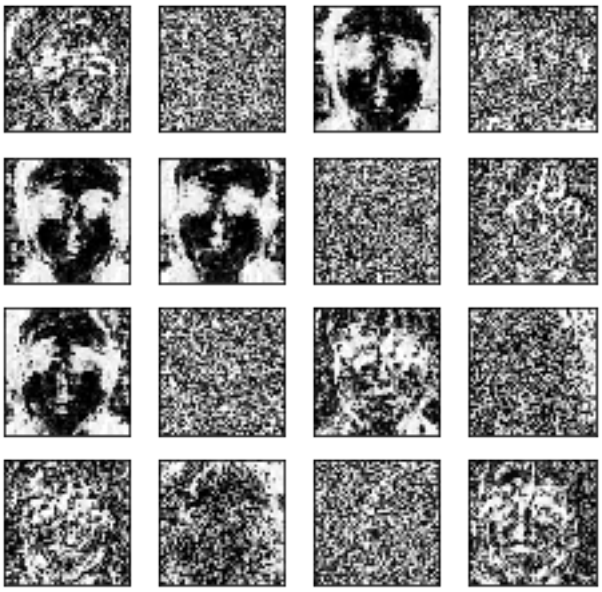




KernelSize=43

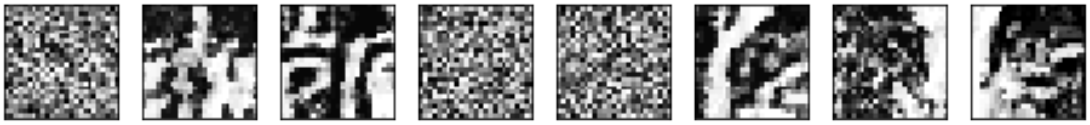


KernelSize=51

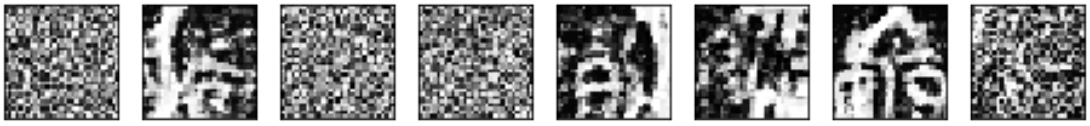


3.3 Parameter Set 3

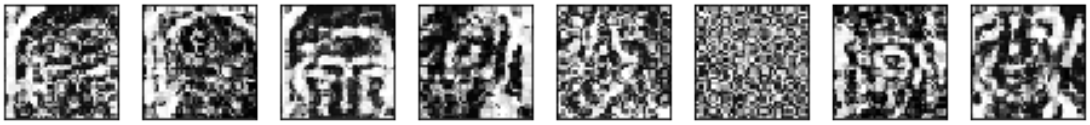
KernelSize=25



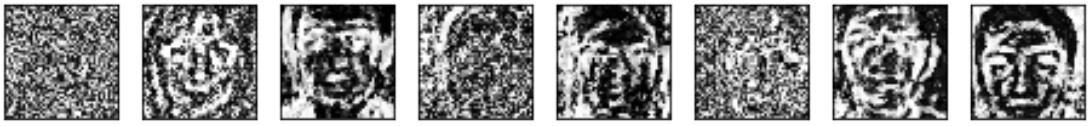
KernelSize=31



KernelSize=39



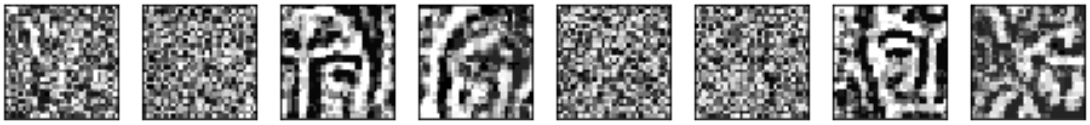
KernelSize=47



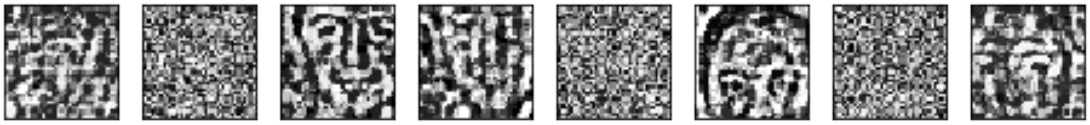
3.4 Parameter Set 4

KernelSize=25

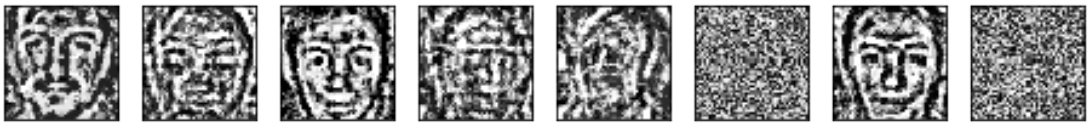
KernelSize=31



KernelSize=39

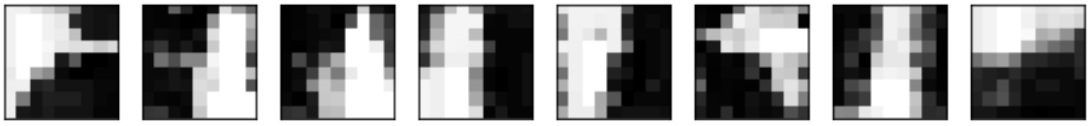


KernelSize=47

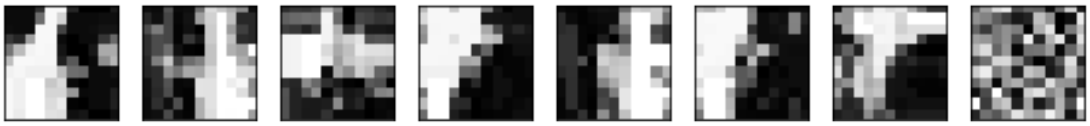


3.5 Parameter Set 5

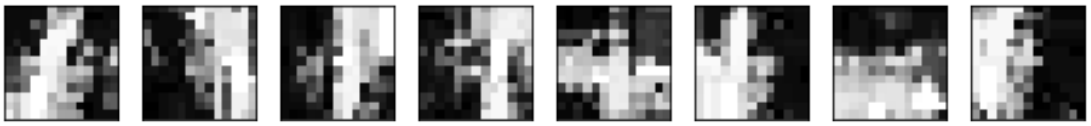
KernelSize=9



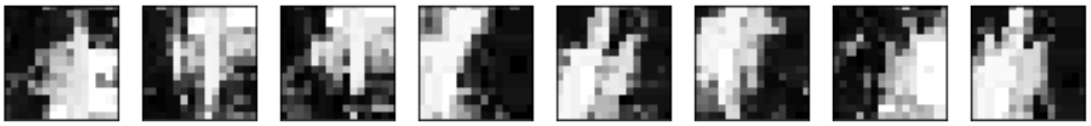
KernelSize=11



KernelSize=13

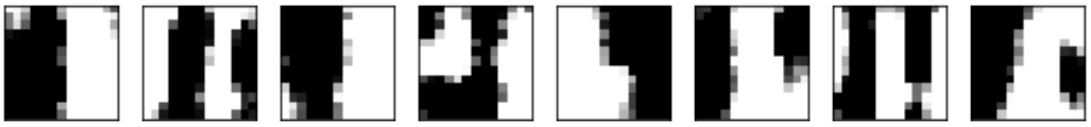


KernelSize=15

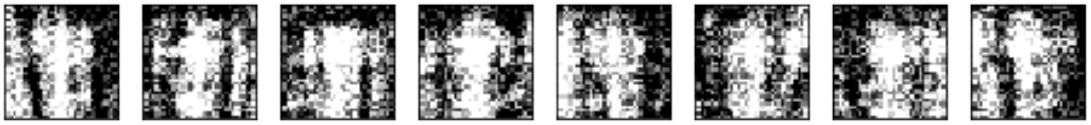


3.6 Parameter Set 6 (no KWTA and Lateral Inhibition)

KernelSize=13

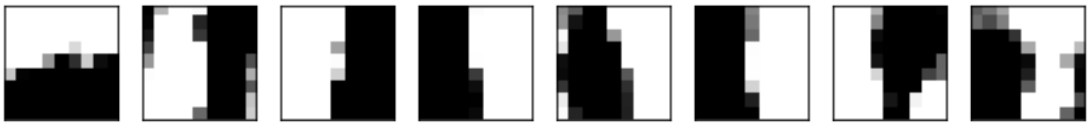


KernelSize=35

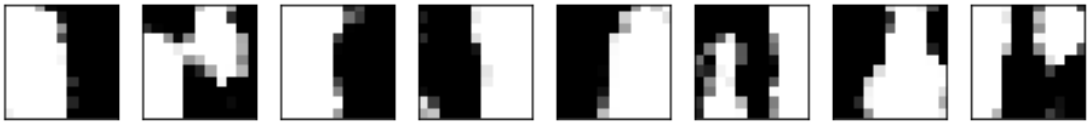


KernelSize=49

KernelSize=9



KernelSize=11



KernelSize=13

KernelSize=15

Chapter 4

Summary

1. By increasing `K`, redundancy of the learned features will increase, and the features will be less specific. For example, in MNIST dataset with four digits, when we set $F = K$, all filters learn almost the same feature which is not desired. This is due to the fact that every digit will update every feature. So the output features will be almost the same and not be specific to one digit.

2. We expect to get $\frac{F}{K}$ different types of features. Again, considering the MNIST experiments, when we set $K = 2$ and $F = 8$, in most cases (different weight initializations) each digit will have two dedicated features.

3. According to previous facts, we can control the redundancy and specificness of the learned weights with `K` and `F`.

4. Filters with small KernelSize tend to learn simple features such as oriented lines or a combination of multiple lines or simple curves (like a part of a digit). However, filters with larger KernelSize tend to learn the whole data.

5. Without using `KWTA` and `Lateral Inhibition` the learned features are more general and may not be specific of the input data. In this setting large features noisy and redundant, but small features learn general oriented lines and simple curves successfully.

6. Based on the previous observation, we could say that `KWTA` performs better with large kernel sizes (large receptive fields).