A Gentle Introduction to Graph Convolution Networks (GCN)

Hugh Nguyen
Data Scientist Co-op
RBC – Joint Security Operations Center

Date: 2018/12/06

AGENDA

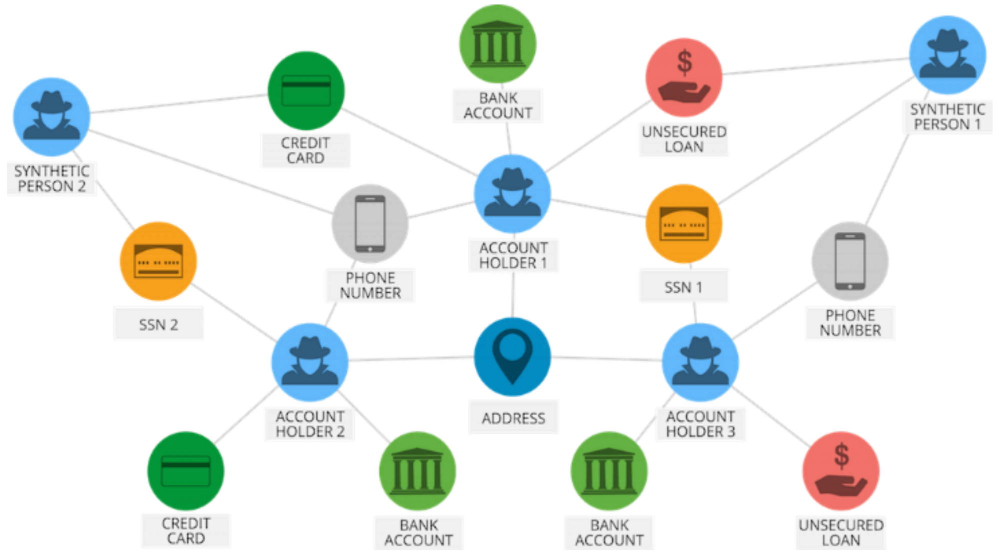
1. What is Convolution Neural Network? Why does it fail on graph?

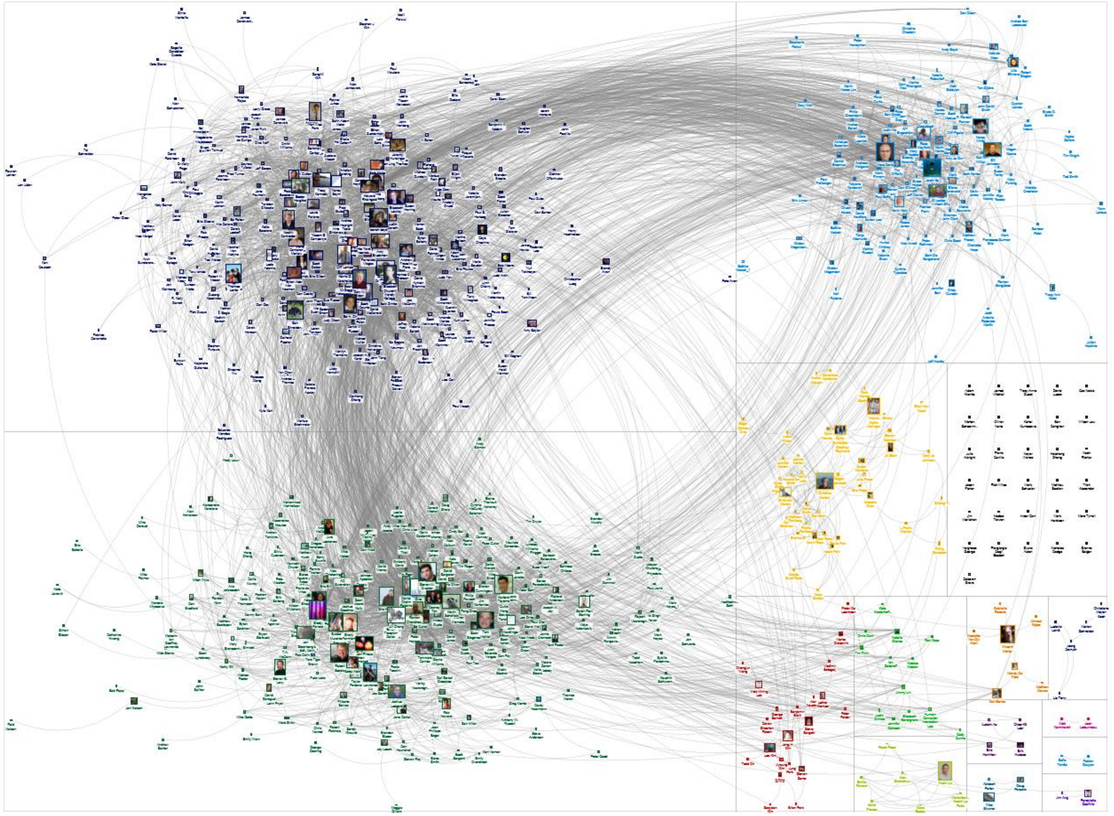
2. Spectral Graph Theory 101

3. Graph Convolution Networks (GCN) and some of its applications

4. Semi-supervised Nodes Classification Performance Review

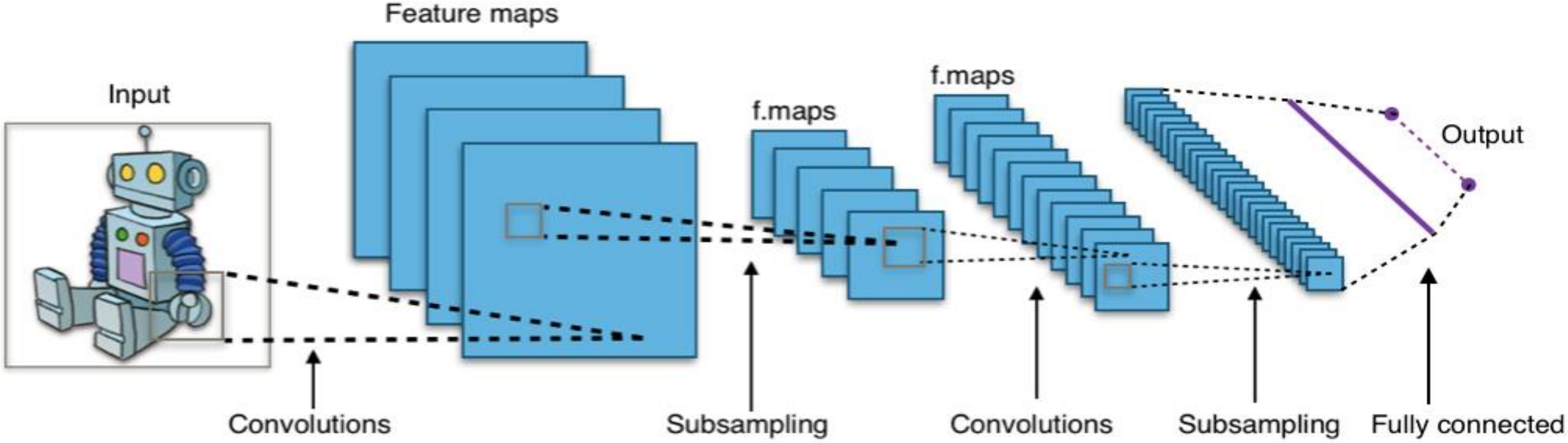
NETWORK IS EVERYWHERE

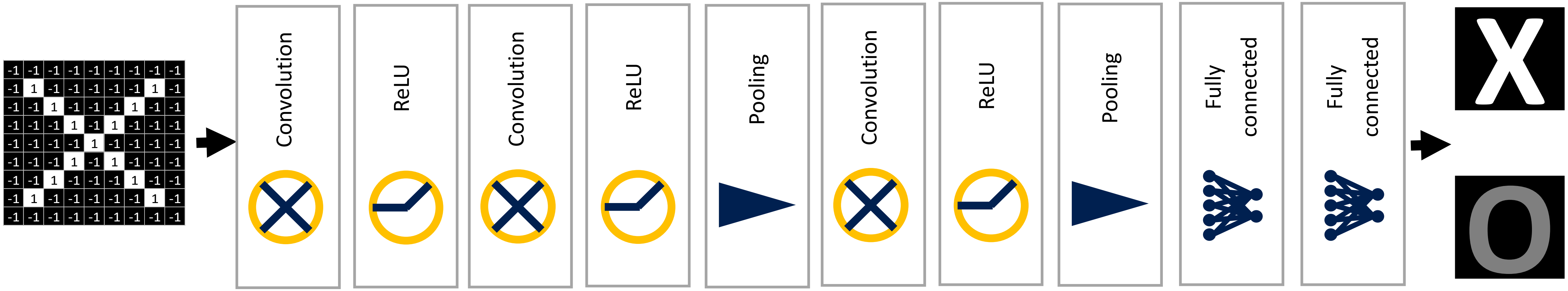




Convolutional Neural Network (CNN)

CONVOLUTION NEURAL NETWORK (CNN)





Source: Brandon Rohrer (Facebook)

FILTERING

Consider this image: (2D Matrix)

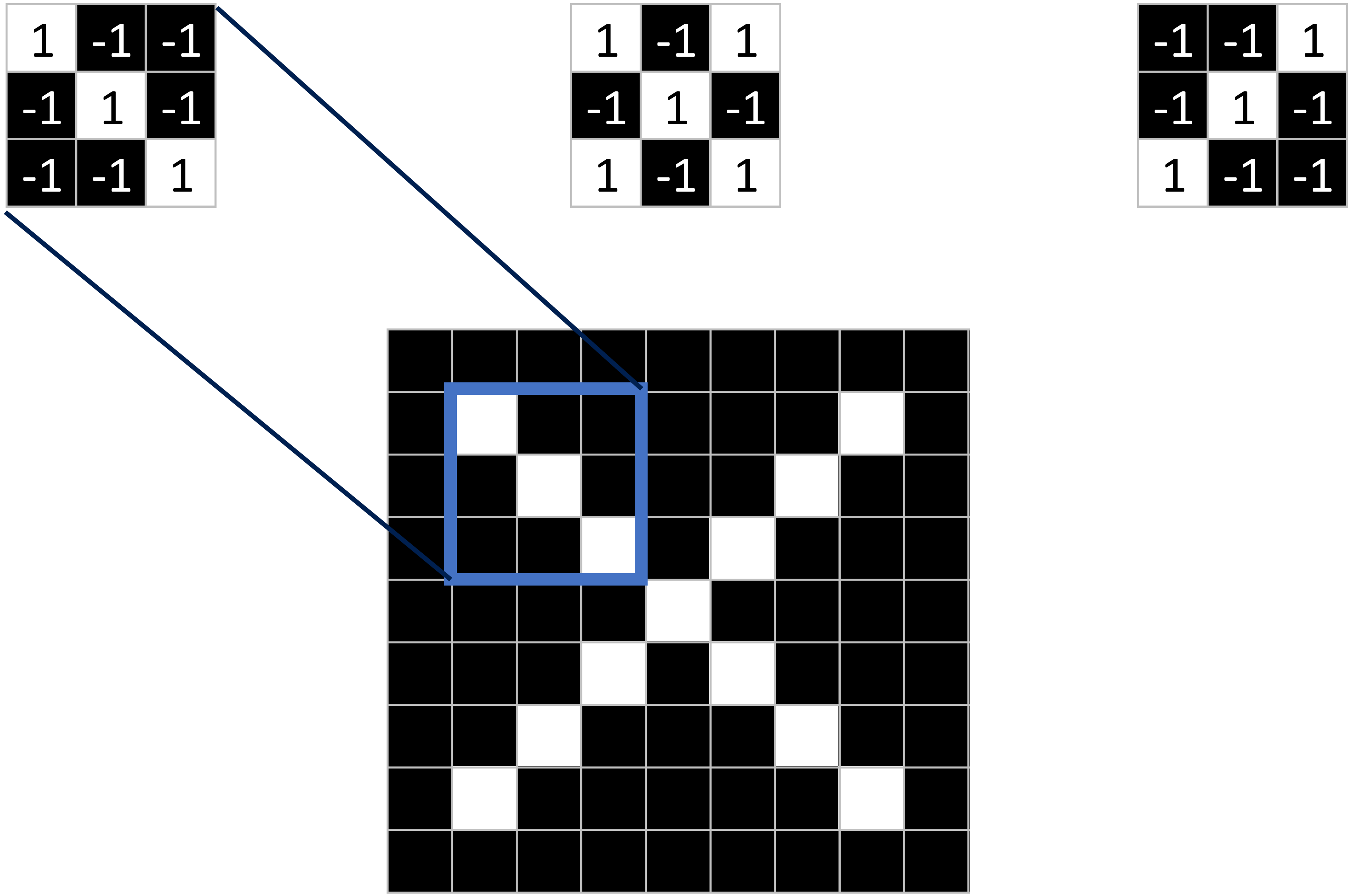
-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

FILTERING OPERATION

Features match pieces of the image

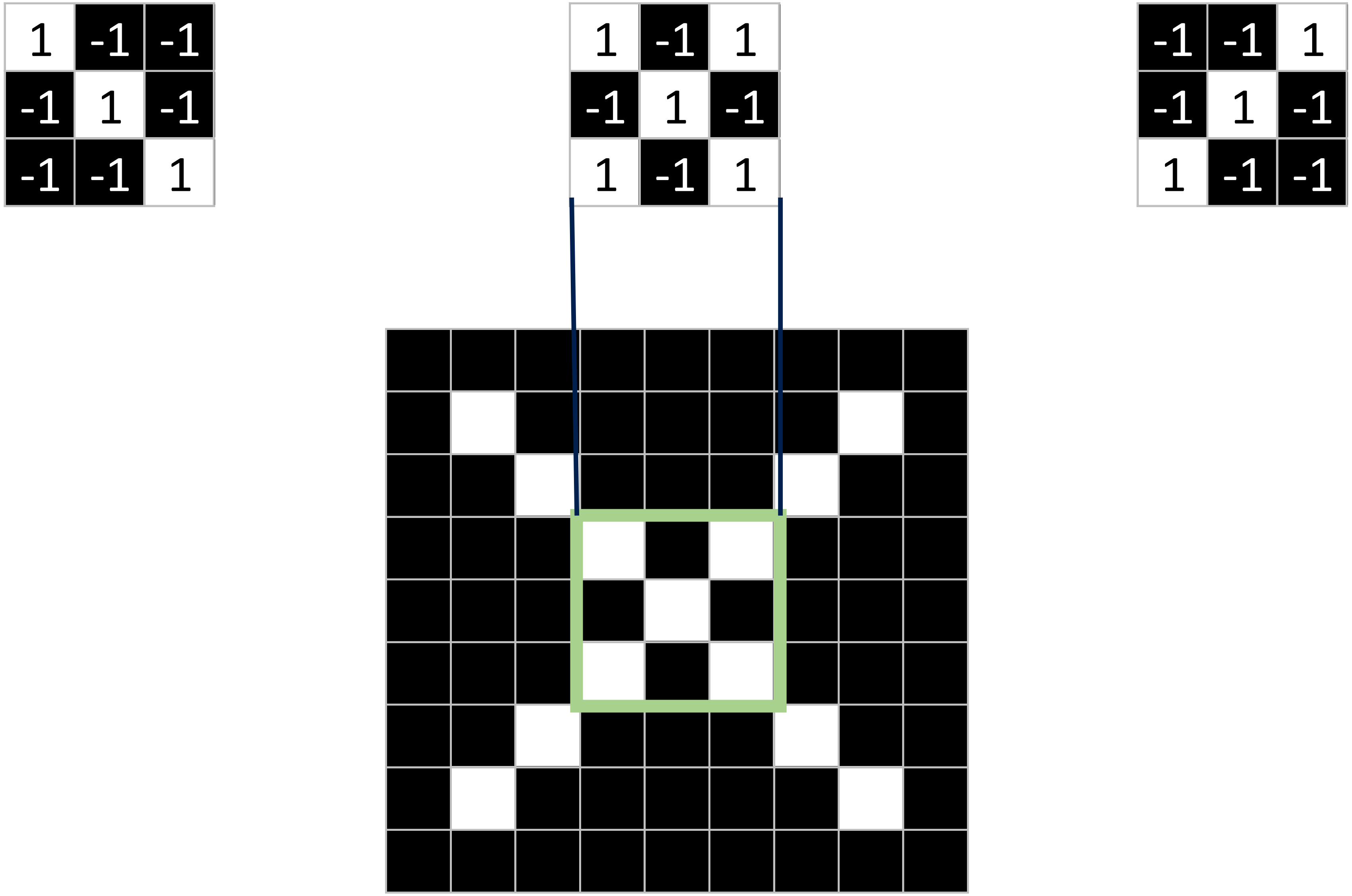
$$
\begin{bmatrix}
-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
-1 & 1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 \\
-1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 \\
-1 & -1 & -1 & 1 & -1 & 1 & -1 & -1 & -1 \\
-1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 \\
-1 & -1 & -1 & 1 & -1 & 1 & -1 & -1 & -1 \\
-1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 \\
-1 & 1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 \\
-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1
\end{bmatrix}
=
\begin{bmatrix}
1 & -1 & -1 \\
-1 & 1 & -1 \\
-1 & -1 & 1
\end{bmatrix}
+
\begin{bmatrix}
1 & -1 & 1 \\
-1 & 1 & -1 \\
1 & -1 & 1
\end{bmatrix}
+
\begin{bmatrix}
-1 & -1 & 1 \\
-1 & 1 & -1 \\
1 & -1 & -1
\end{bmatrix}
$$

FILTERING OPERATION



Source: Brandon Rohrer

FILTERING OPERATION



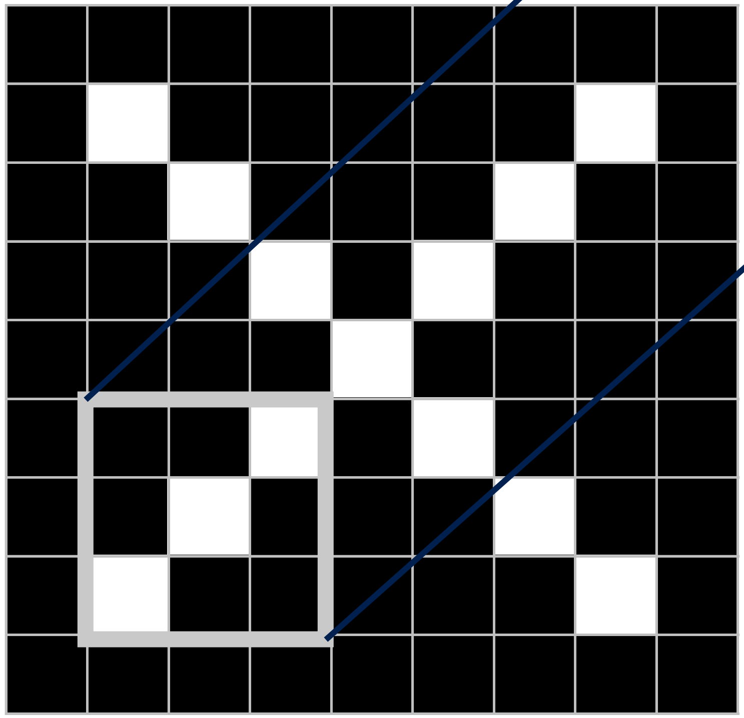
Source: Brandon Rohrer

FILTERING OPERATION

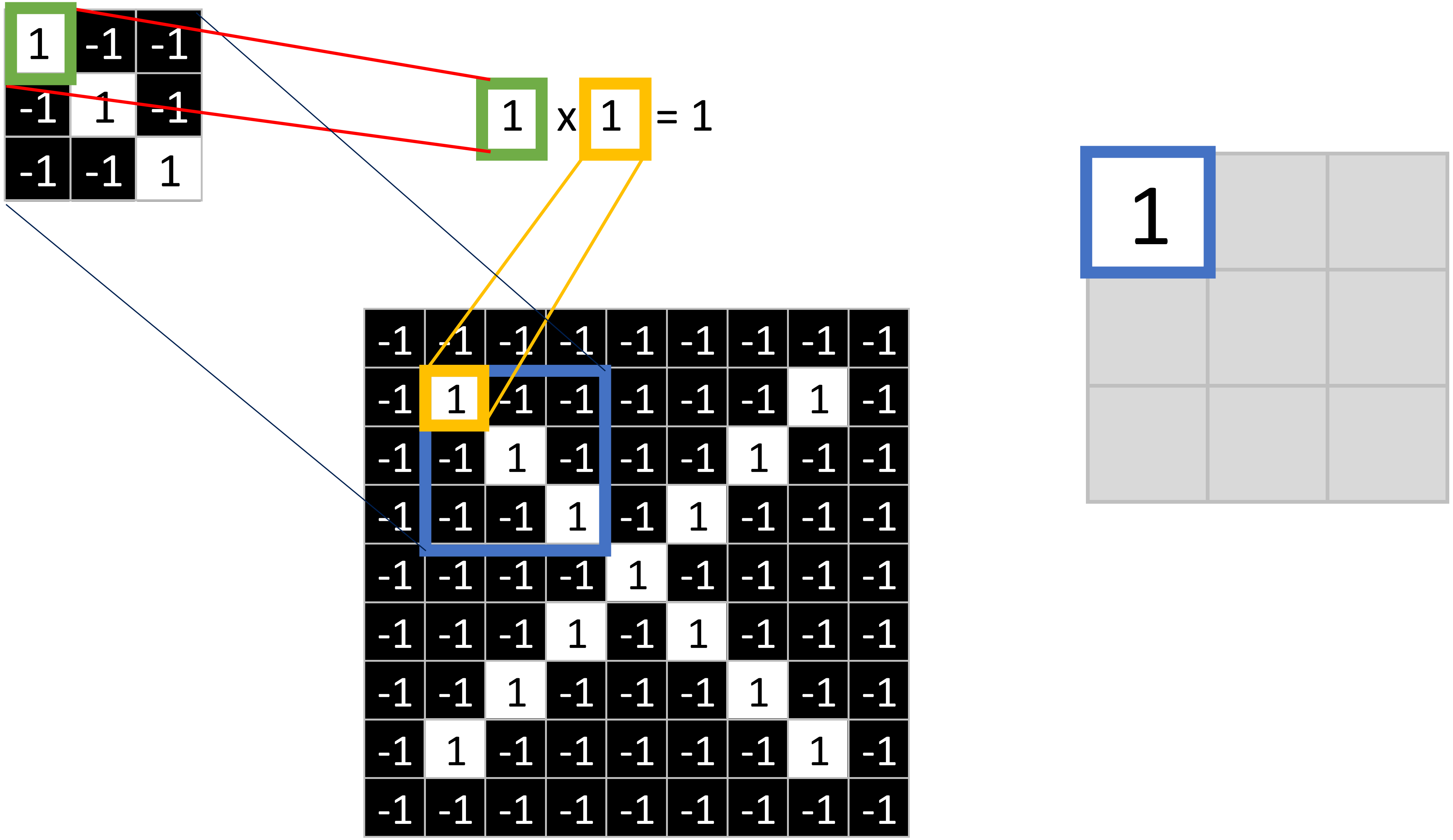
1	-1	-1
-1	1	-1
-1	-1	1

1	-1	1
-1	1	-1
1	-1	1

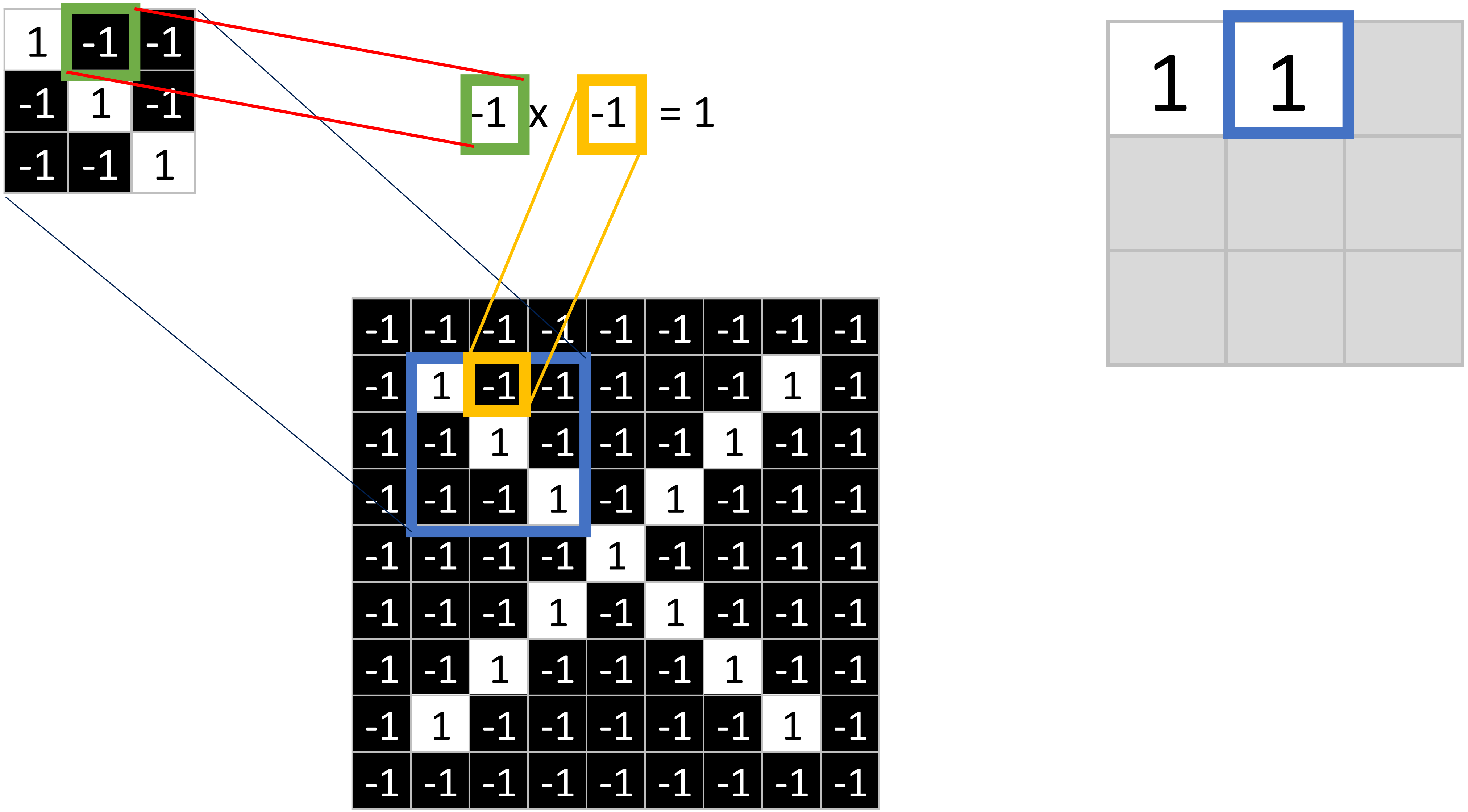
-1	-1	1
-1	1	-1
1	-1	-1



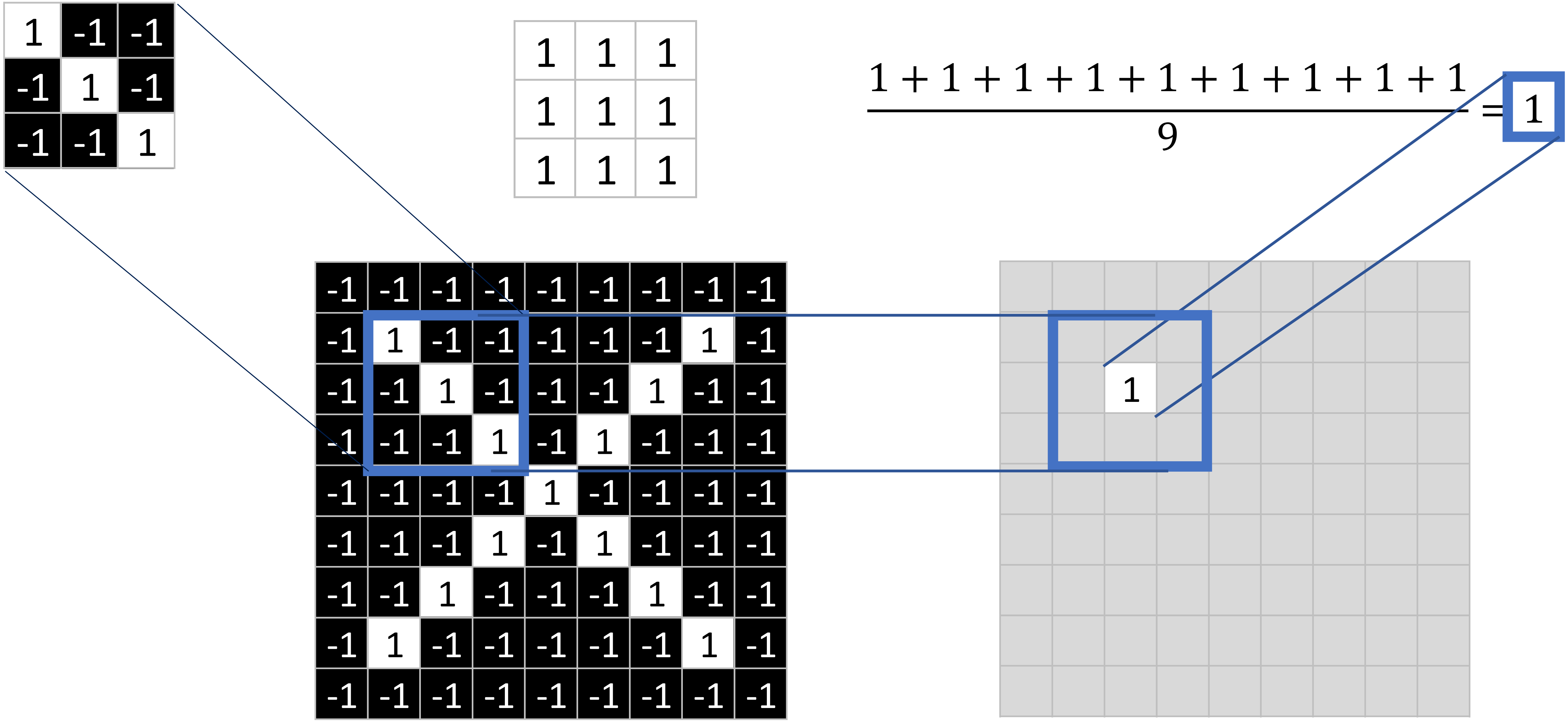
FILTERING OPERATION



FILTERING OPERATION



FILTERING OPERATION



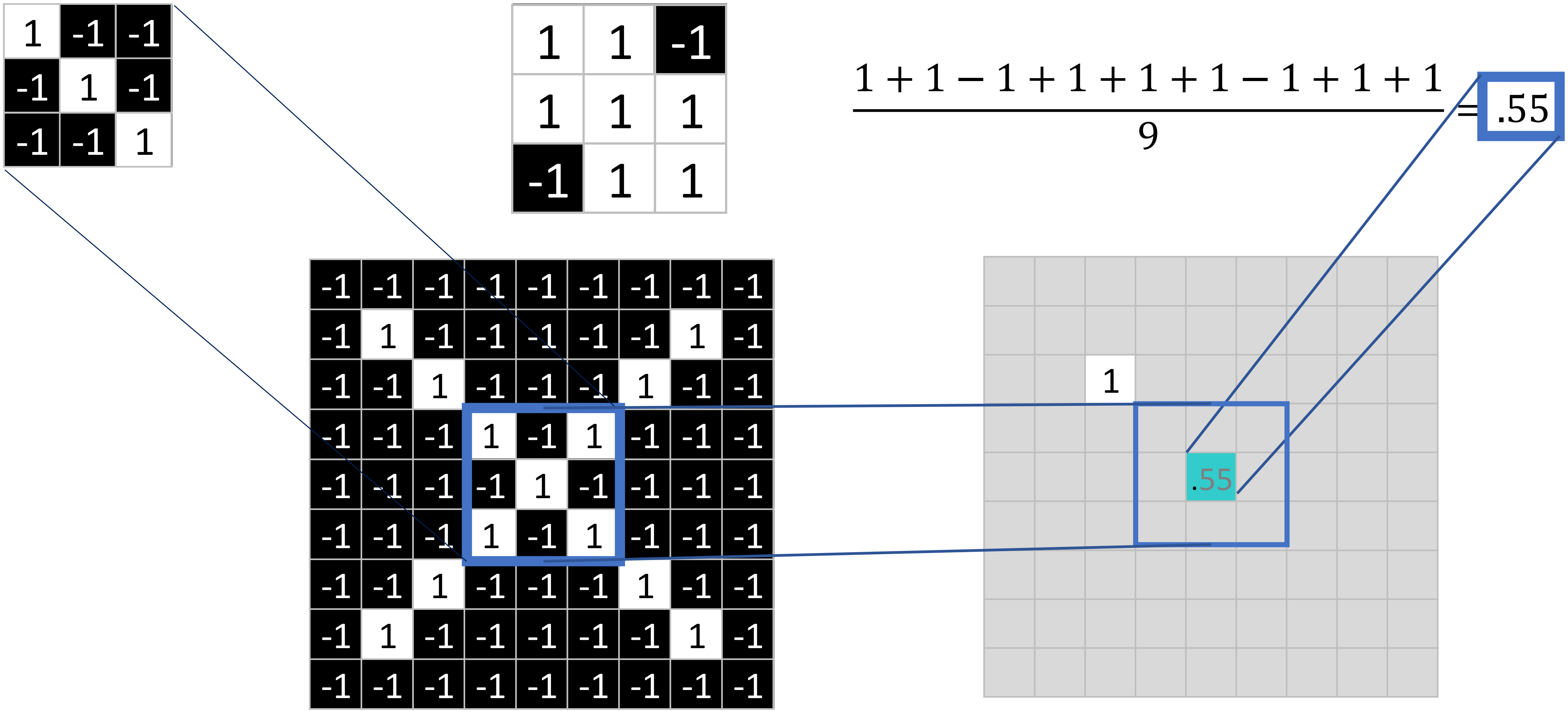
FILTERING OPERATION

1	-1	-1
-1	1	-1
-1	-1	1

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

1	1	-1
1	1	1
-1	1	1

FILTERING OPERATION



FILTERING OPERATION

1	-1	-1
-1	1	-1
-1	-1	1

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

FILTERING OPERATION

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

$\otimes$

1	-1	-1
-1	1	-1
-1	-1	1

=

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

FILTERING OPERATION

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

$\otimes$

1	-1	-1
-1	1	-1
-1	-1	1

=

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

$\otimes$

1	-1	1
-1	1	-1
1	-1	1

=

0.33	-0.55	0.11	-0.11	0.11	-0.55	0.33
-0.55	0.55	-0.55	0.33	-0.55	0.55	-0.55
0.11	-0.55	0.55	-0.77	0.55	-0.55	0.11
-0.11	0.33	-0.77	1.00	-0.77	0.33	-0.11
0.11	-0.55	0.55	-0.77	0.55	-0.55	0.11
-0.55	0.55	-0.55	0.33	-0.55	0.55	-0.55
0.33	-0.55	0.11	-0.11	0.11	-0.55	0.33

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

$\otimes$

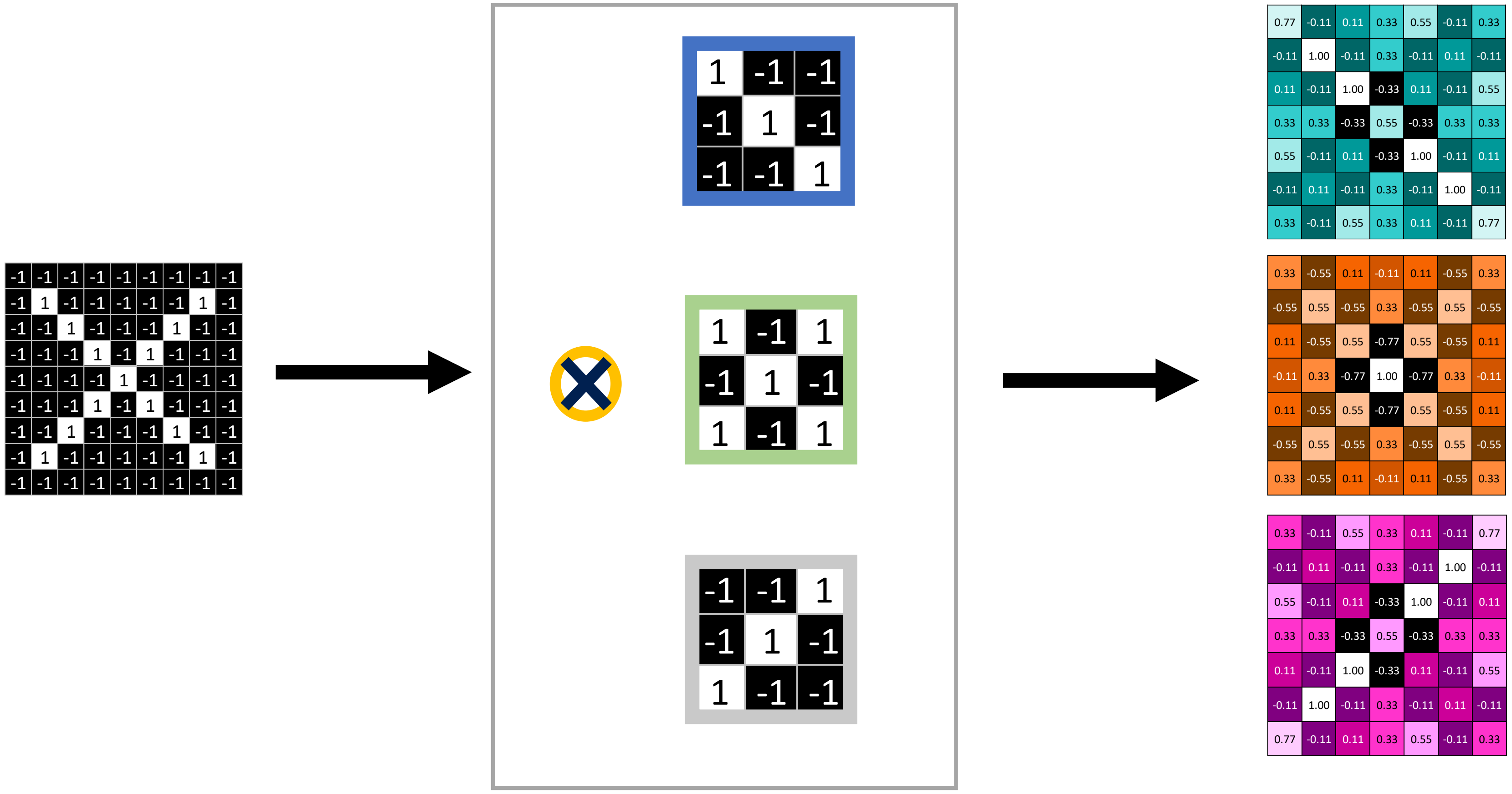
-1	-1	1
-1	1	-1
1	-1	-1

=

0.33	-0.11	0.55	0.33	0.11	-0.11	0.77
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.77	-0.11	0.11	0.33	0.55	-0.11	0.33

CONVOLUTION LAYER

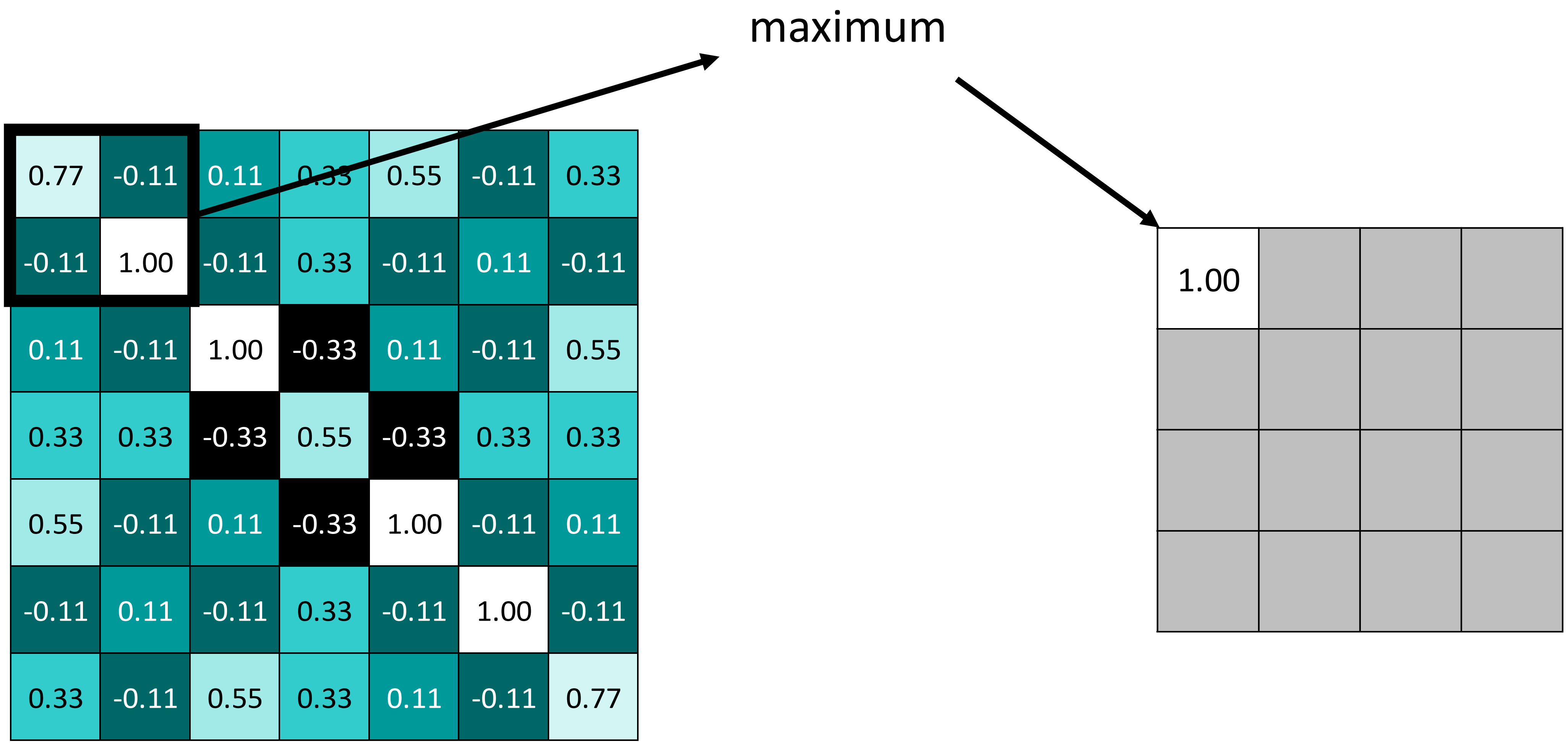
A stack of filters forms a convolution layer



POOLING OPERATION

1. Select a pooling window size; go with 2
2. Select a stride; go with 2

POOLING OPERATION



POOLING OPERATION

maximum

0.77	-0.11	0.11	0.33	0.55	0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

1.00	0.33		

POOLING OPERATION

maximum

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

1.00	0.33	0.55	0.33

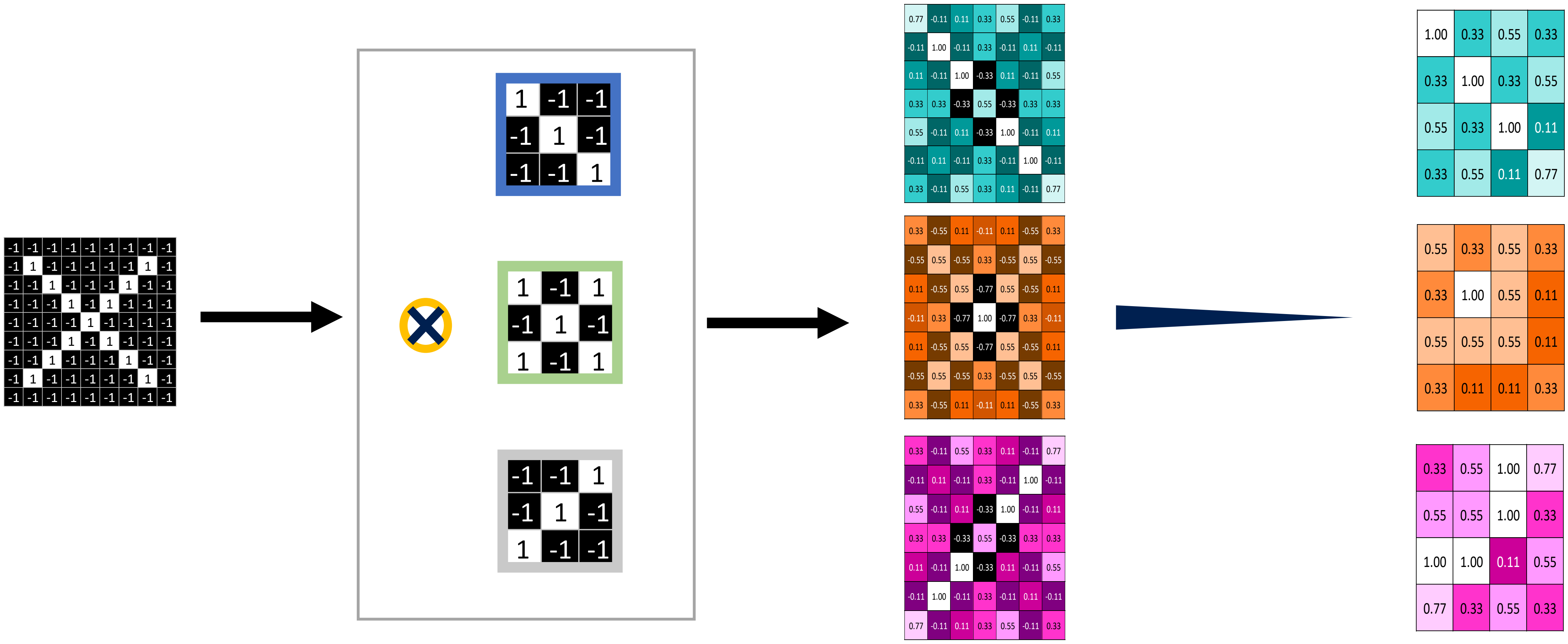
POOLING OPERATION

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

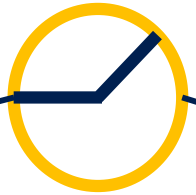
max pooling

1.00	0.33	0.55	0.33
0.33	1.00	0.33	0.55
0.55	0.33	1.00	0.11
0.33	0.55	0.11	0.77

POOLING OPERATION



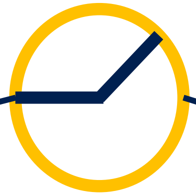
ACTIVATION FUNCTION – ReLU



0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

0.77						

ACTIVATION FUNCTION – ReLU

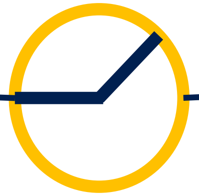


0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

0.77	0					

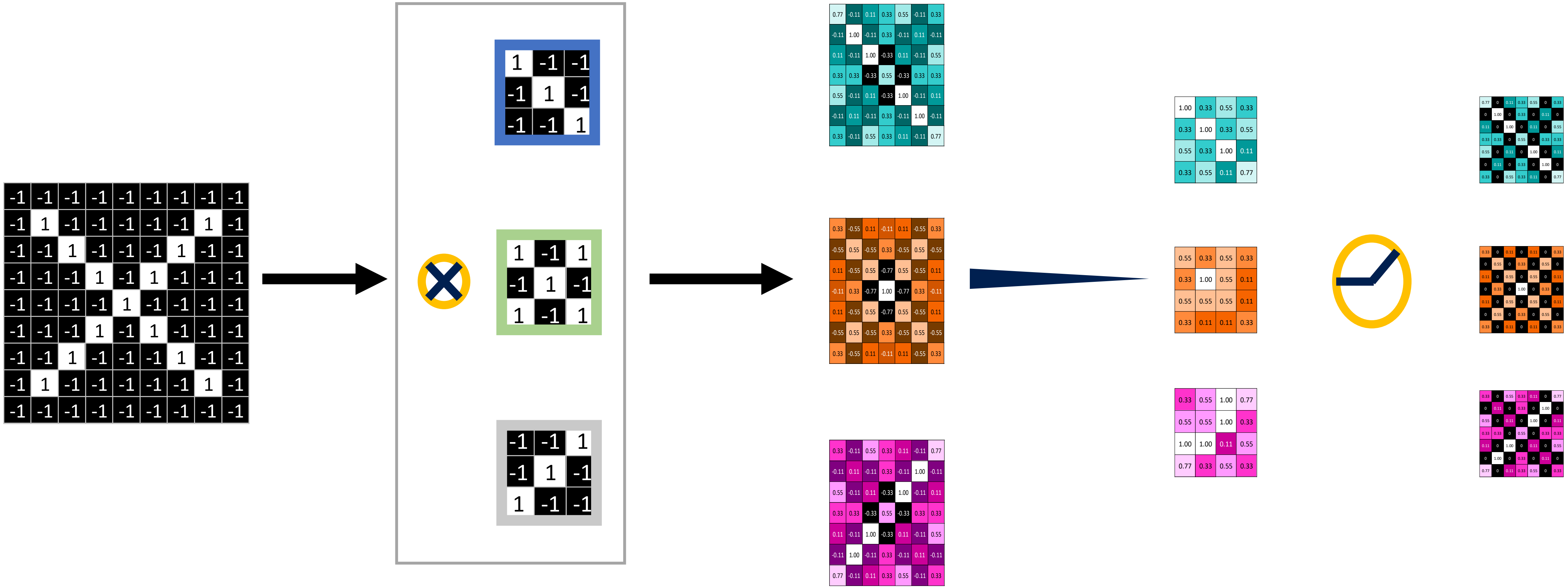
ACTIVATION FUNCTION – ReLU

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77



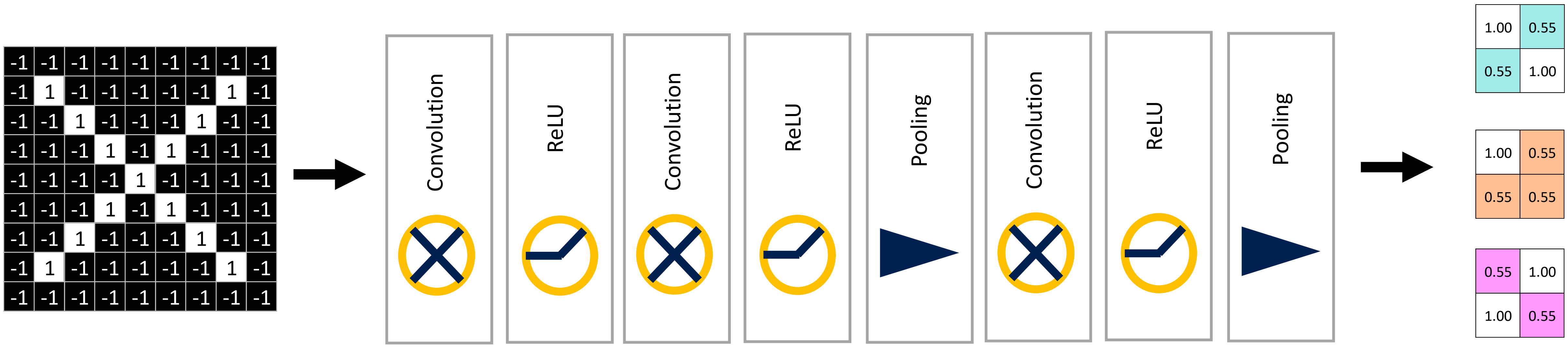
0.77	0	0.11	0.33	0.55	0	0.33
0	1.00	0	0.33	0	0.11	0
0.11	0	1.00	0	0.11	0	0.55
0.33	0.33	0	0.55	0	0.33	0.33
0.55	0	0.11	0	1.00	0	0.11
0	0.11	0	0.33	0	1.00	0
0.33	0	0.55	0.33	0.11	0	0.77

SO FAR ...



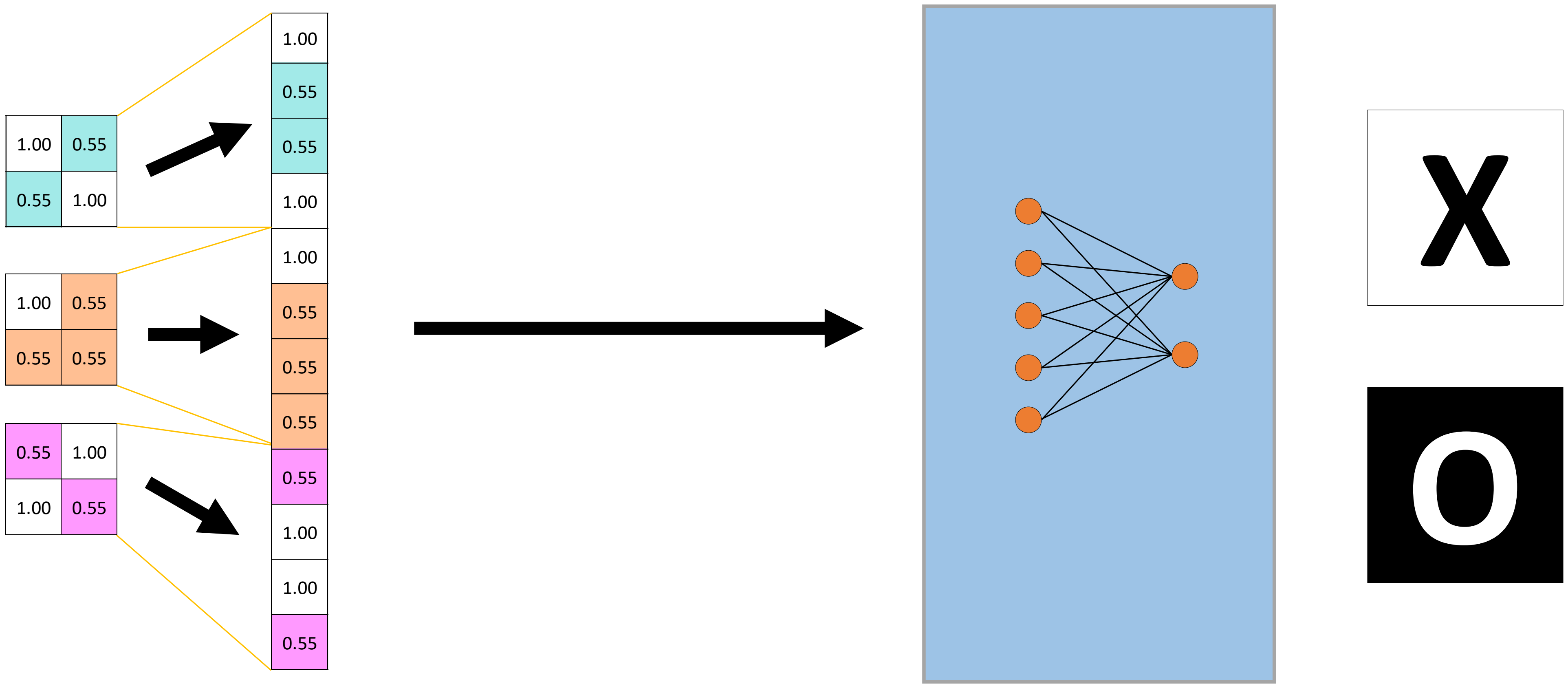
-1 -1 -1 -1 -1 -1 -1 -1 -1
-1 1 -1 -1 -1 -1 -1 1 -1
-1 -1 1 -1 -1 -1 1 -1 -1
-1 -1 -1 1 -1 1 -1 -1 -1
-1 -1 -1 -1 1 -1 -1 -1 -1
-1 -1 -1 1 -1 1 -1 -1 -1
-1 -1 1 -1 -1 -1 1 -1 -1
-1 1 -1 -1 -1 -1 -1 1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1
1 -1 -1
-1 1 -1
-1 -1 1
1 -1 1
-1 1 -1
1 -1 1
-1 -1 1
-1 1 -1
1 -1 -1

SO FAR ...

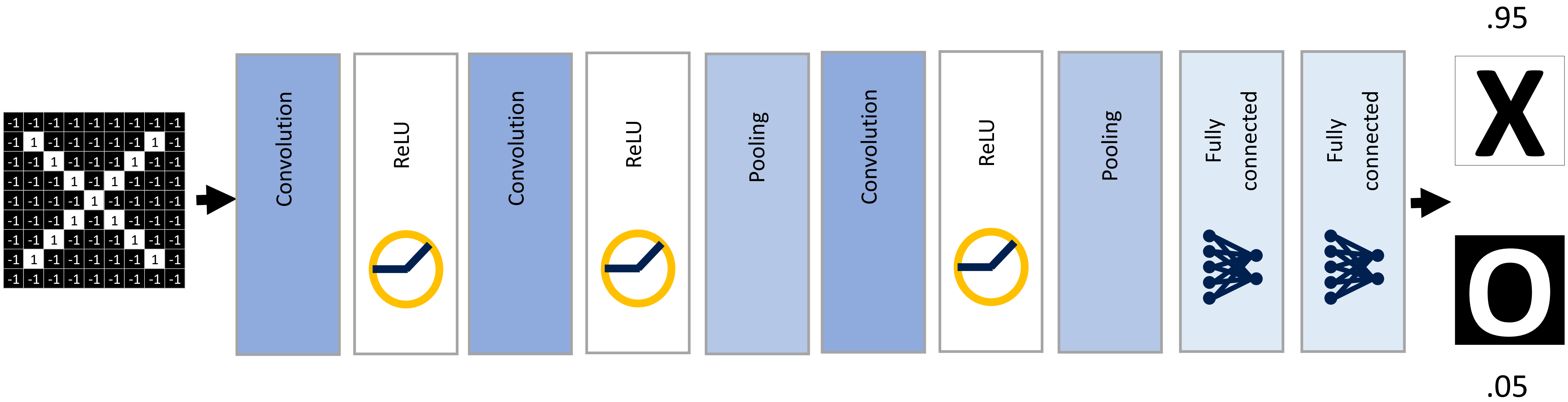


-1 -1 -1 -1 -1 -1 -1 -1 -1
-1 1 -1 -1 -1 -1 -1 1 -1
-1 -1 1 -1 -1 -1 1 -1 -1
-1 -1 -1 1 -1 1 -1 -1 -1
-1 -1 -1 -1 1 -1 -1 -1 -1
-1 -1 -1 1 -1 1 -1 -1 -1
-1 -1 1 -1 -1 -1 1 -1 -1
-1 1 -1 -1 -1 -1 -1 1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1
Convolution
ReLU
Convolution
ReLU
Pooling
Convolution
ReLU
Pooling
1.00 0.55
0.55 1.00
1.00 0.55
0.55 0.55
0.55 1.00
1.00 0.55

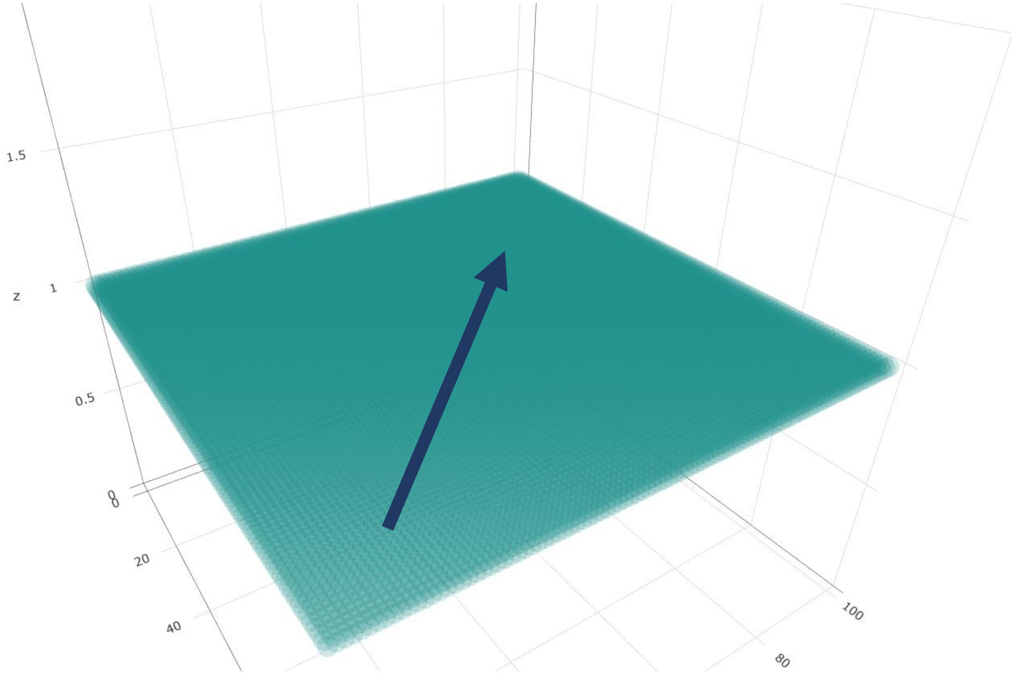
FULLY CONNECTED LAYER



CNN FULL STACK



CNN PROS



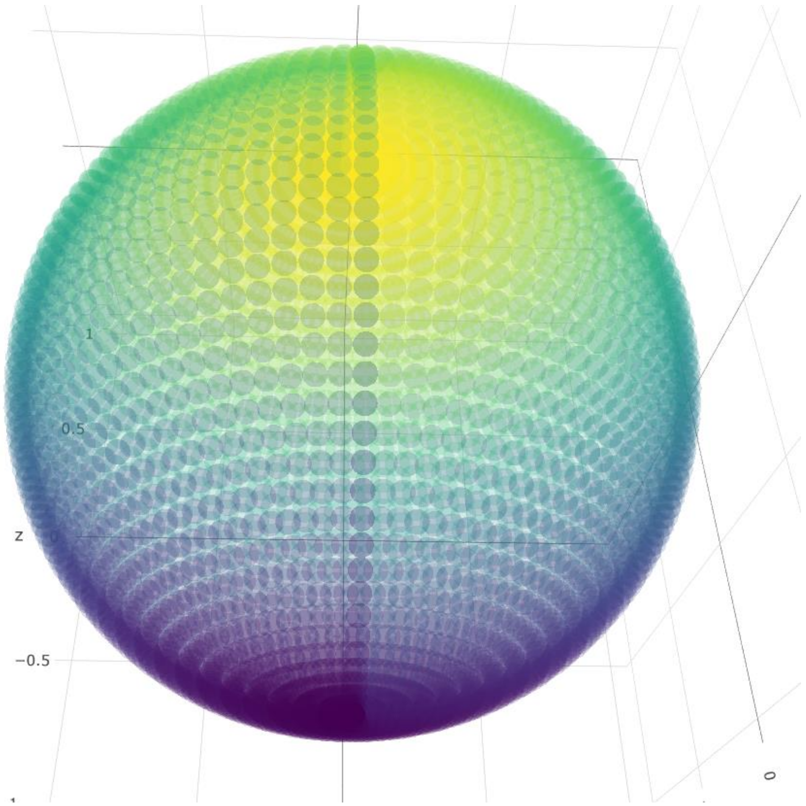
$$d(\mathbf{p},\mathbf{q}) = d(\mathbf{q},\mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

Linear

$$f(x) * g(x) = \int_{-\infty}^{\infty} f(y)\, g(x-y)\, dy$$

CNN PROS



$$a = \sin^2(\Delta\varphi/2) + \cos\varphi_1 \cdot \cos\varphi_2 \cdot \sin^2(\Delta\lambda/2)$$

$$c = 2 \cdot \operatorname{atan2}(\sqrt{a}, \sqrt{(1-a)})$$

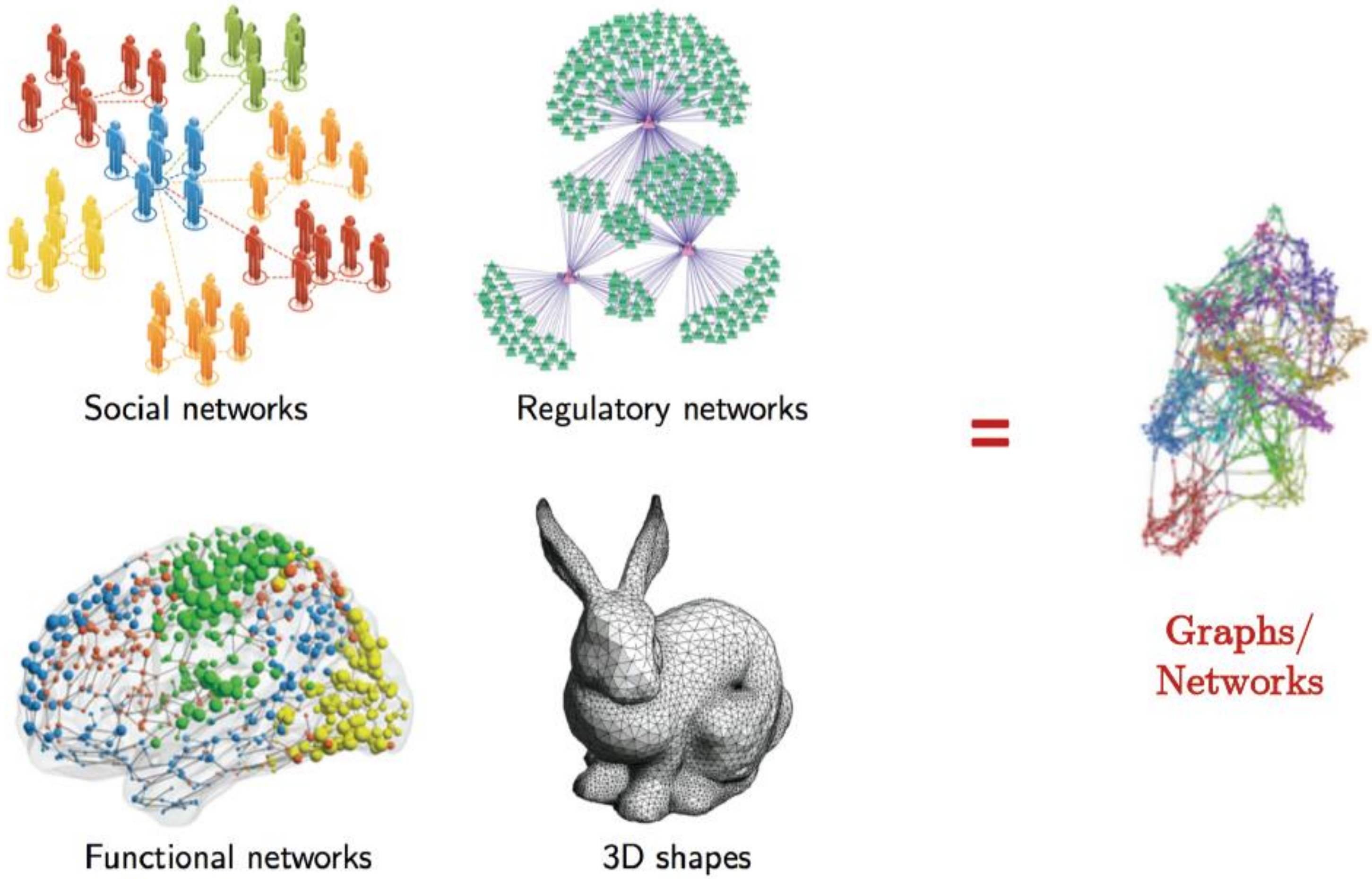
$$d = R \cdot c$$

φ: latitude
λ: longitude
R: Earth radius (6371 km)

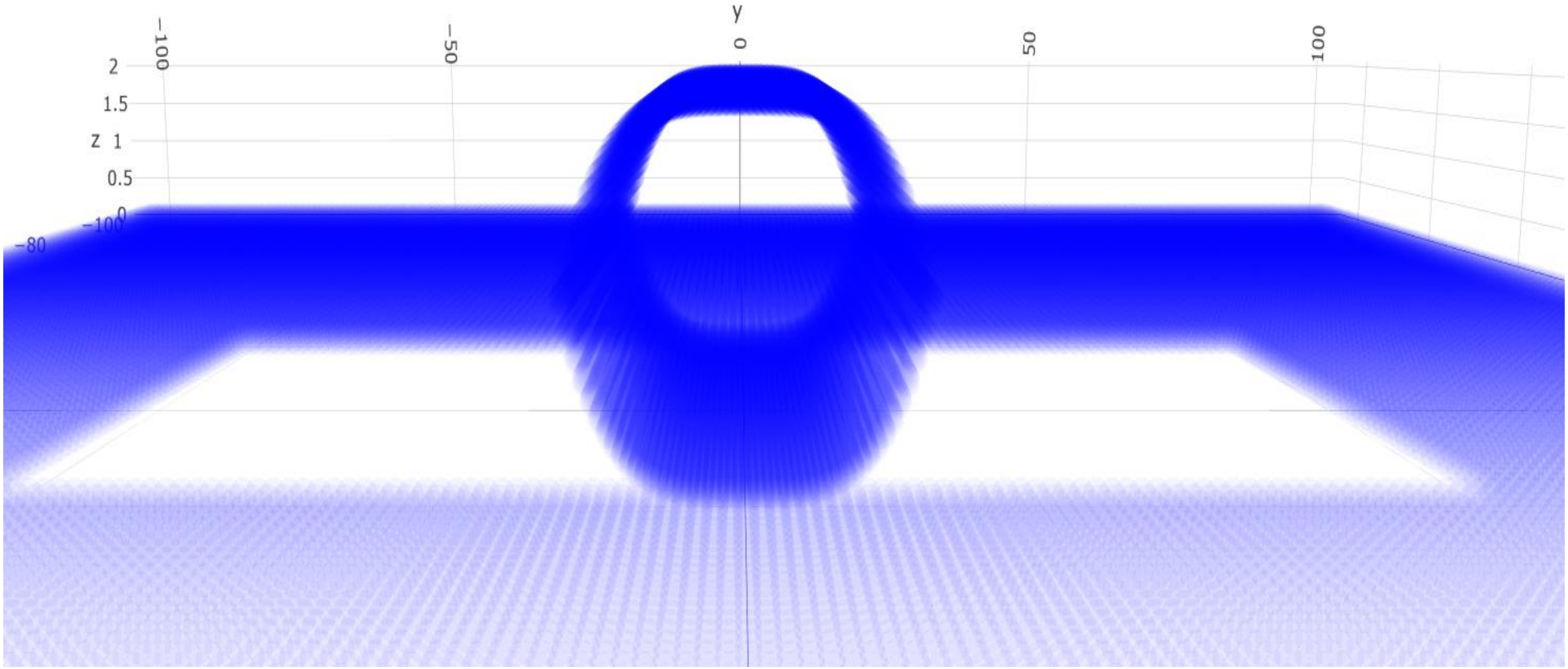
Spherical

$$(f * g)(\theta, \phi) = \sum_{l} \sum_{m=-l}^{l} \hat{f}(l, m) \cdot \hat{g}(l, 0) \langle Y_l^m, \rho_{R(\theta,\phi)}(Y_l^0) \rangle$$

WAIT WHAT … ?

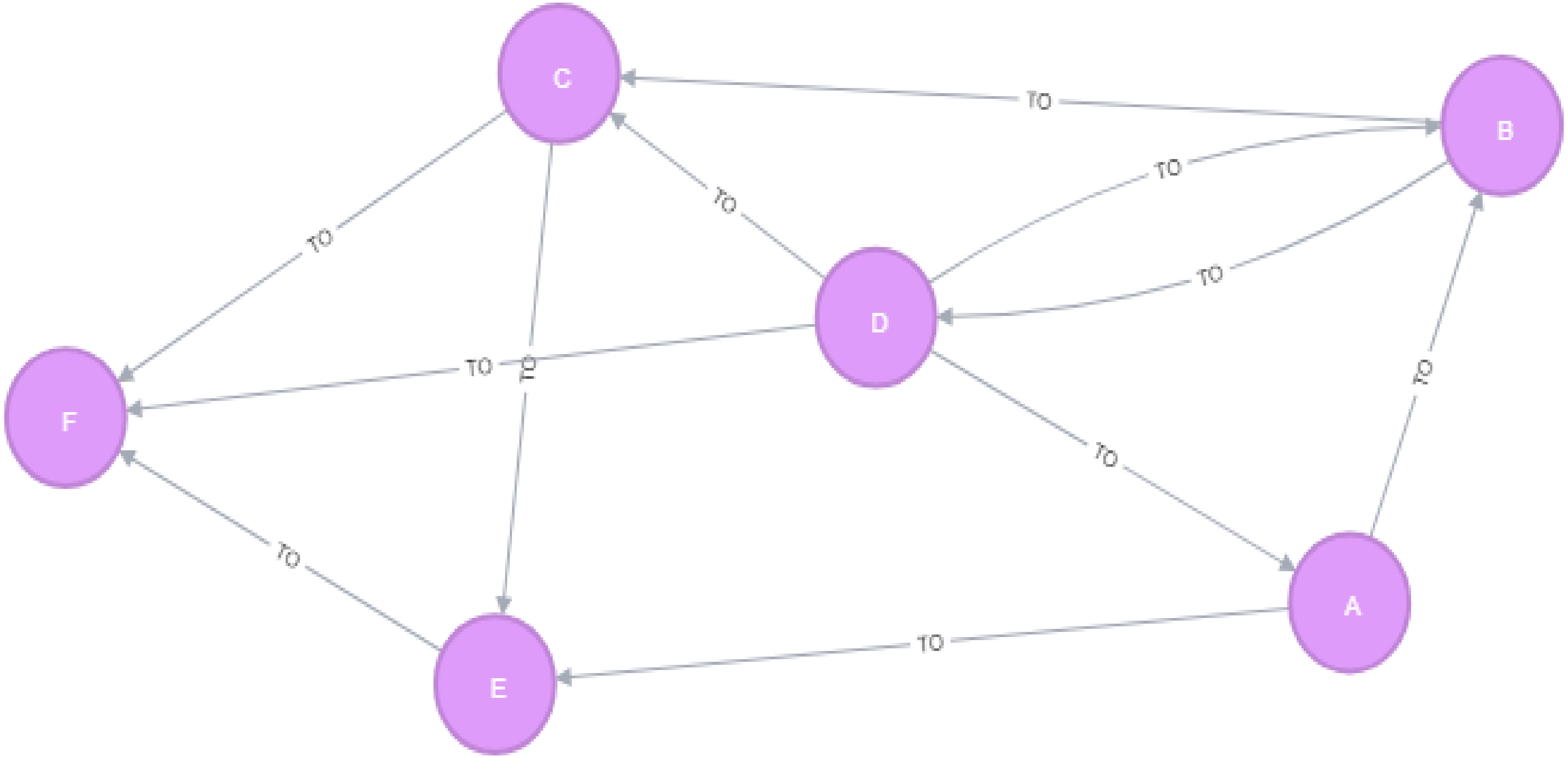

Social networks
Regulatory networks
Functional networks
3D shapes
=
Graphs/
Networks

CNN CANNOT HANDLE NON-EUCLIDEAN DOMAIN

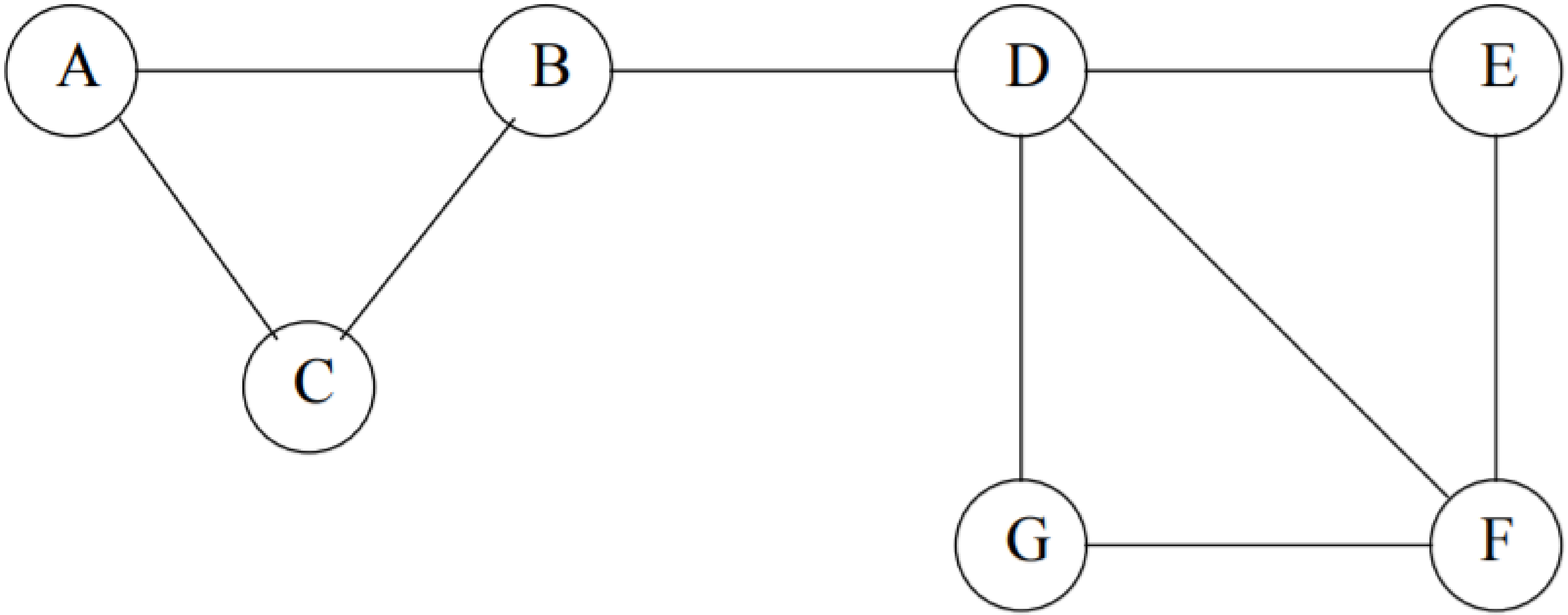


Spectral Graph Theory 101

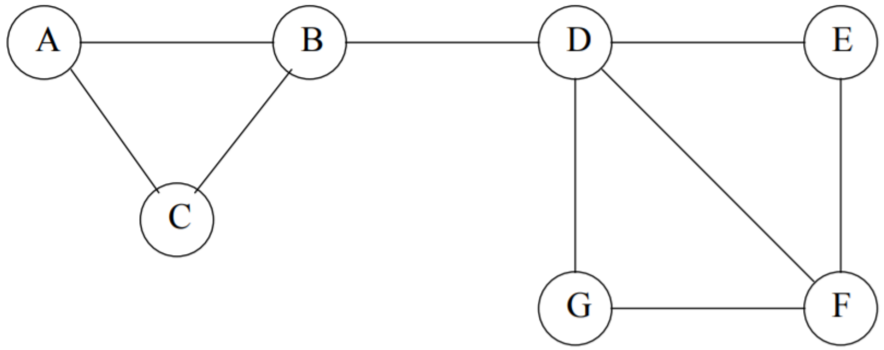
WHAT IS A GRAPH?



MATRIX REPRESENTATION OF A GRAPH

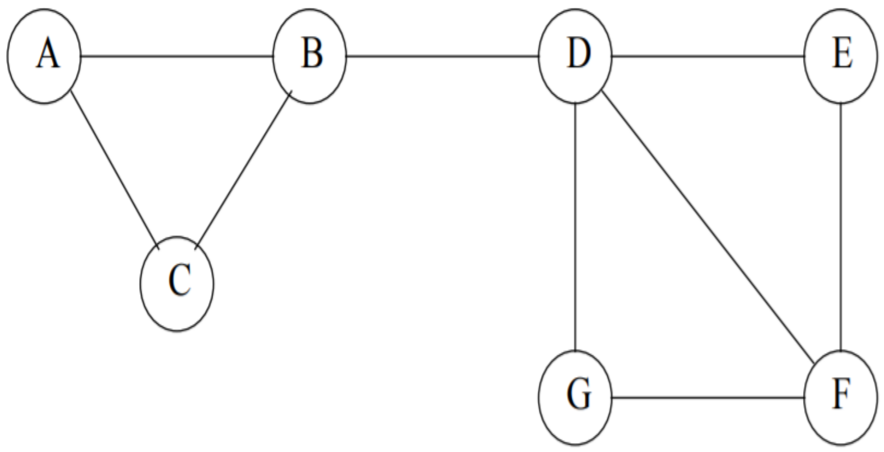


ADJACENCY MATRIX



$$
\begin{array}{c}
\\
A\\ B\\ C\\ D\\ E\\ F\\ G
\end{array}
\begin{array}{c}
\begin{array}{ccccccc} A & B & C & D & E & F & G \end{array}\\
\begin{bmatrix}
0 & 1 & 1 & 0 & 0 & 0 & 0\\
1 & 0 & 1 & 1 & 0 & 0 & 0\\
1 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 1 & 0 & 0 & 1 & 1 & 1\\
0 & 0 & 0 & 1 & 0 & 1 & 0\\
0 & 0 & 0 & 1 & 1 & 0 & 1\\
0 & 0 & 0 & 1 & 0 & 1 & 0
\end{bmatrix}
\end{array}
$$

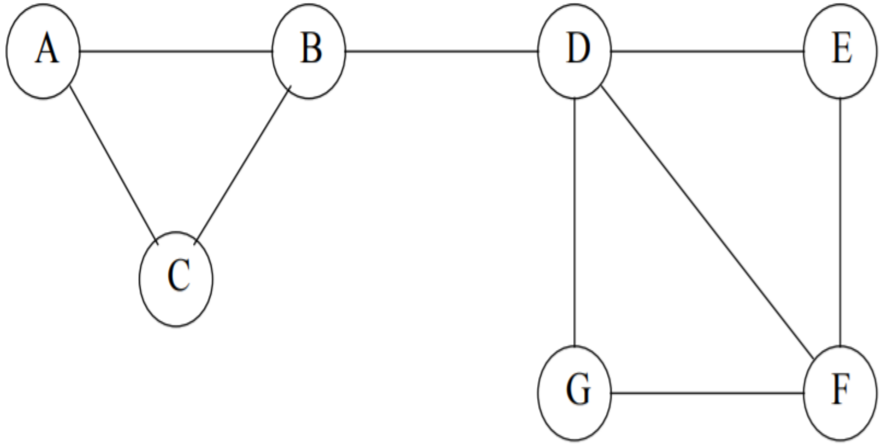
DEGREE MATRIX



$$\begin{array}{c c}
 & \begin{array}{ccccccc} A & B & C & D & E & F & G \end{array} \\
\begin{array}{c} A \\ B \\ C \\ D \\ E \\ F \\ G \end{array} &
\begin{bmatrix}
2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 3 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 4 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 3 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 2
\end{bmatrix}
\end{array}$$

LAPLACIAN MATRIX

L = D – A

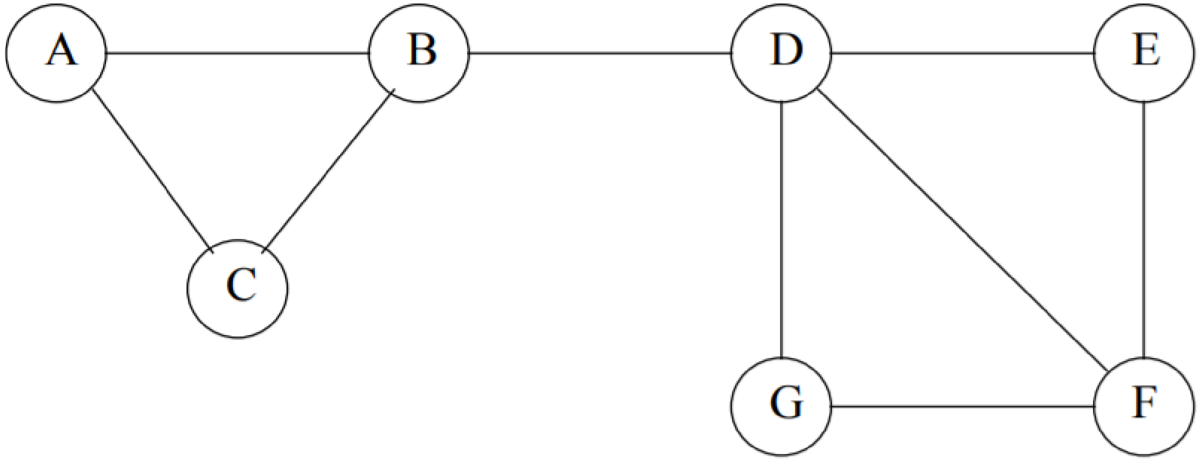


$$
\begin{array}{c c}
 & \begin{array}{ccccccc} A & B & C & D & E & F & G \end{array} \\
\begin{array}{c} A \\ B \\ C \\ D \\ E \\ F \\ G \end{array} &
\begin{bmatrix}
2 & -1 & -1 & 0 & 0 & 0 & 0 \\
-1 & 3 & -1 & -1 & 0 & 0 & 0 \\
-1 & -1 & 2 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 4 & -1 & -1 & -1 \\
0 & 0 & 0 & -1 & 2 & -1 & 0 \\
0 & 0 & 0 & -1 & -1 & 3 & -1 \\
0 & 0 & 0 & -1 & 0 & -1 & 2
\end{bmatrix}
\end{array}
$$

SPECTRAL GRAPH PARTIONING

$$\begin{array}{c|ccccccc} & A & B & C & D & E & F & G \\ \hline A & 2 & -1 & -1 & 0 & 0 & 0 & 0 \\ B & -1 & 3 & -1 & -1 & 0 & 0 & 0 \\ C & -1 & -1 & 2 & 0 & 0 & 0 & 0 \\ D & 0 & -1 & 0 & 4 & -1 & -1 & -1 \\ E & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ F & 0 & 0 & 0 & -1 & -1 & 3 & -1 \\ G & 0 & 0 & 0 & -1 & 0 & -1 & 2 \end{array}$$

Vertices								
	ei.value	0	0.398	2	3	3.34	4	5.26
A	ei.vector	1	-1.38	0	too lazy	too lazy	too lazy	too lazy
B		1	-0.833	0	too lazy	too lazy	too lazy	too lazy
C		1	-1.384	0	too lazy	too lazy	too lazy	too lazy
D		1	0.602	0	too lazy	too lazy	too lazy	too lazy
E		1	1	-1	too lazy	too lazy	too lazy	too lazy
F		1	1	0	too lazy	too lazy	too lazy	too lazy
G		1	1	1	too lazy	too lazy	too lazy	too lazy



EIGEN DECOMPOSITION

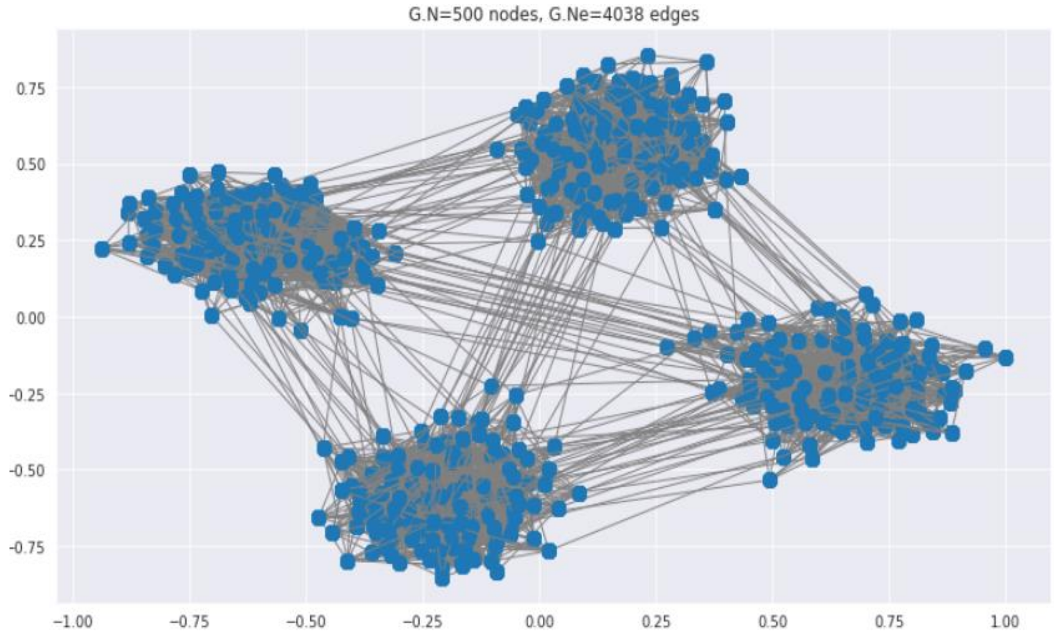
We consider here only undirected graphs, such that the Laplacian matrix is real symmetric, thus diagonalizable in an orthonormal eigenbasis

$$\mathbf{L} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\top},$$

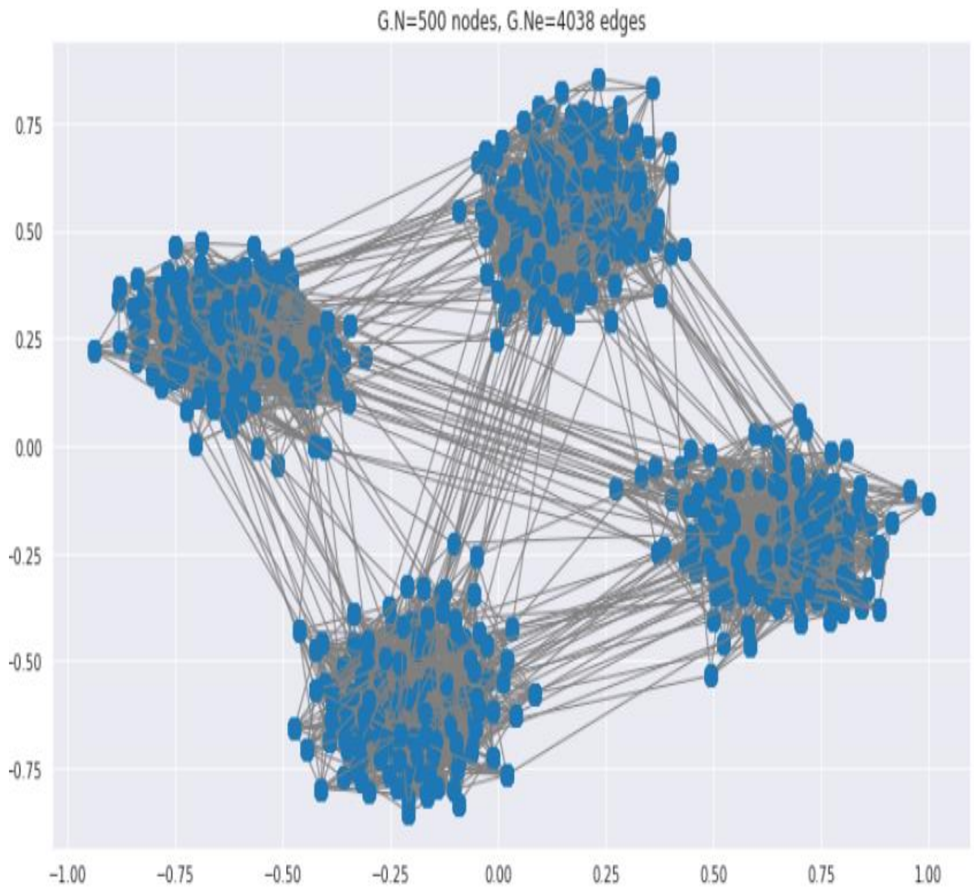
where $\mathbf{U} = (\mathbf{u}_1 | \ldots | \mathbf{u}_N) \in \mathbb{R}^{N \times N}$ is the matrix of orthonormal eigenvectors and $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$ is the diagonal matrix of associated sorted eigenvalues:

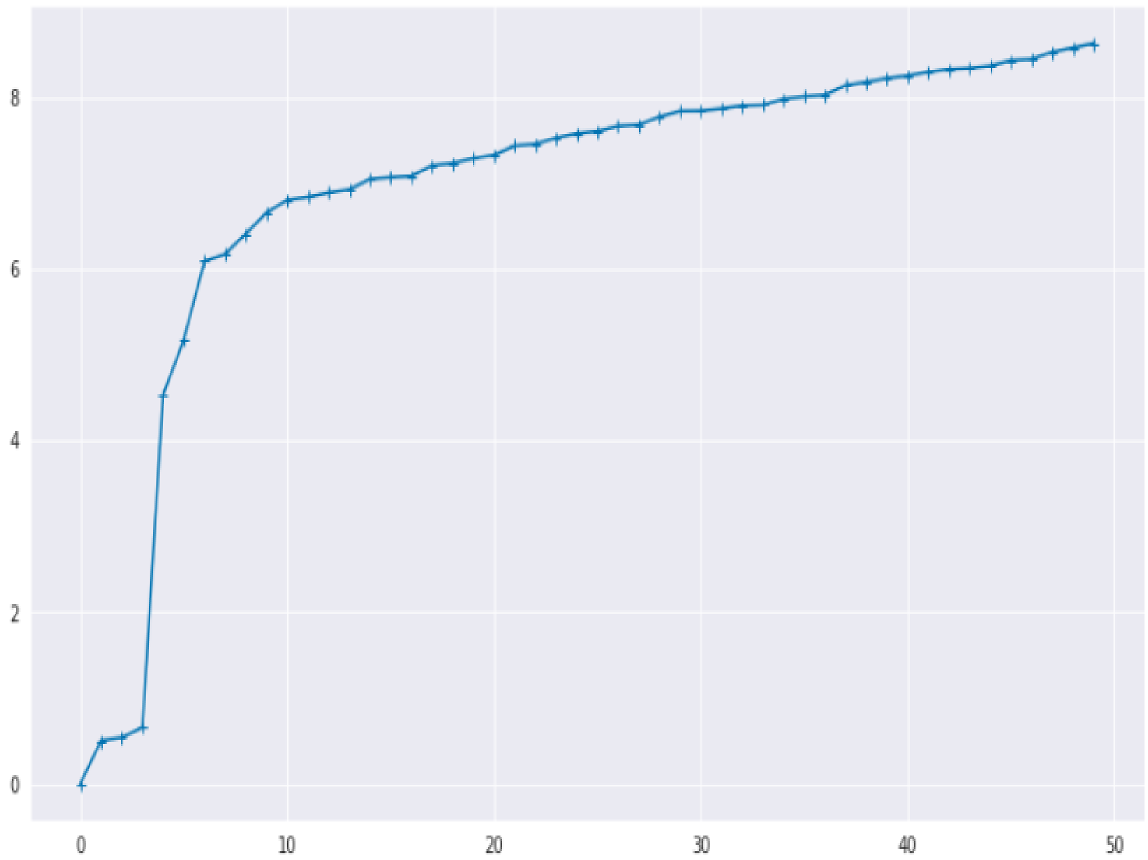
$$\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N.$$

NOTE: that λ_1 is necessarily 0 and that $\lambda_2 > 0$ iff the graph is connected.



FOURIER (SPECTRAL) DOMAIN



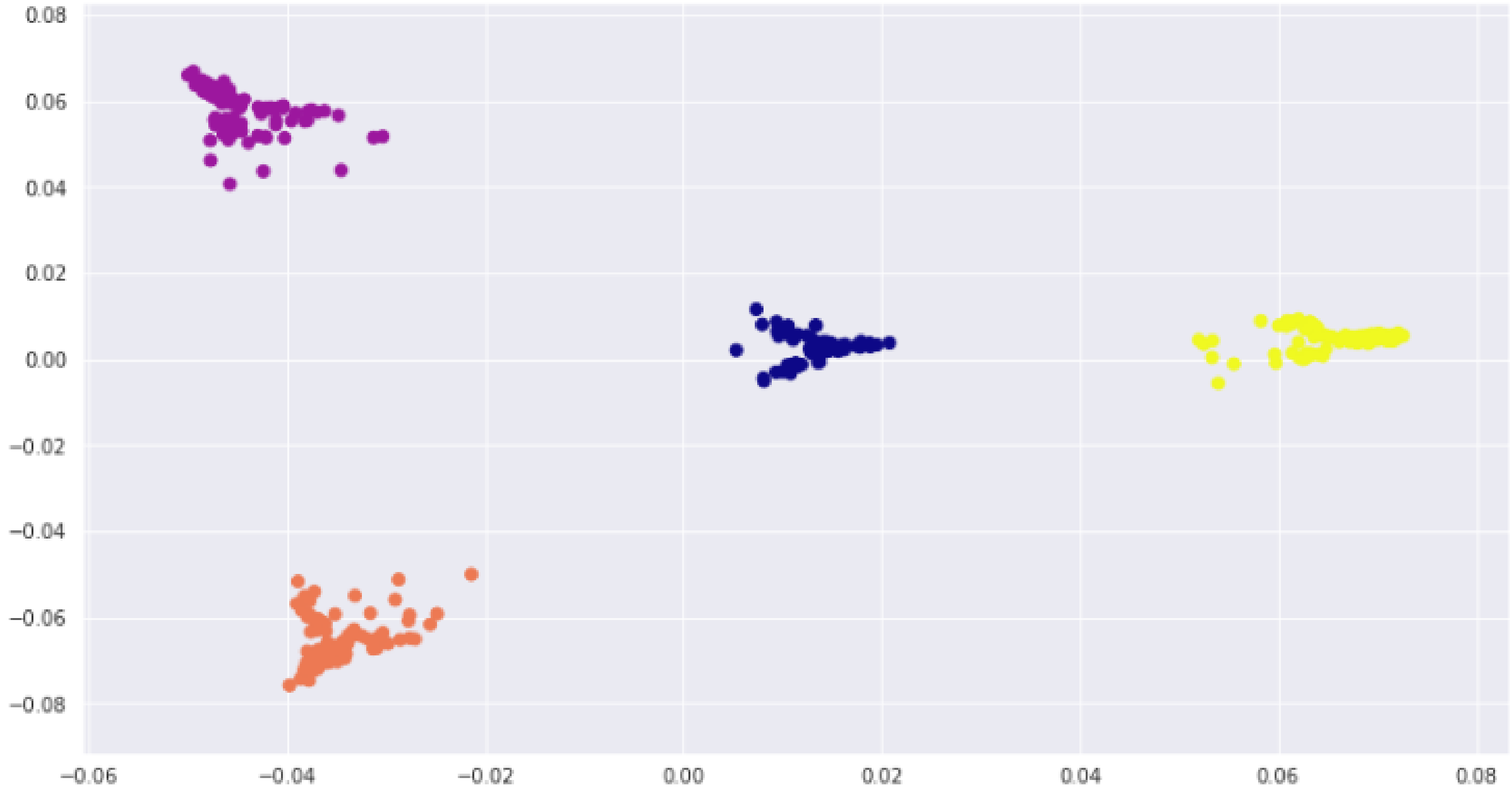


Eigenvalues – lambda 0 to n

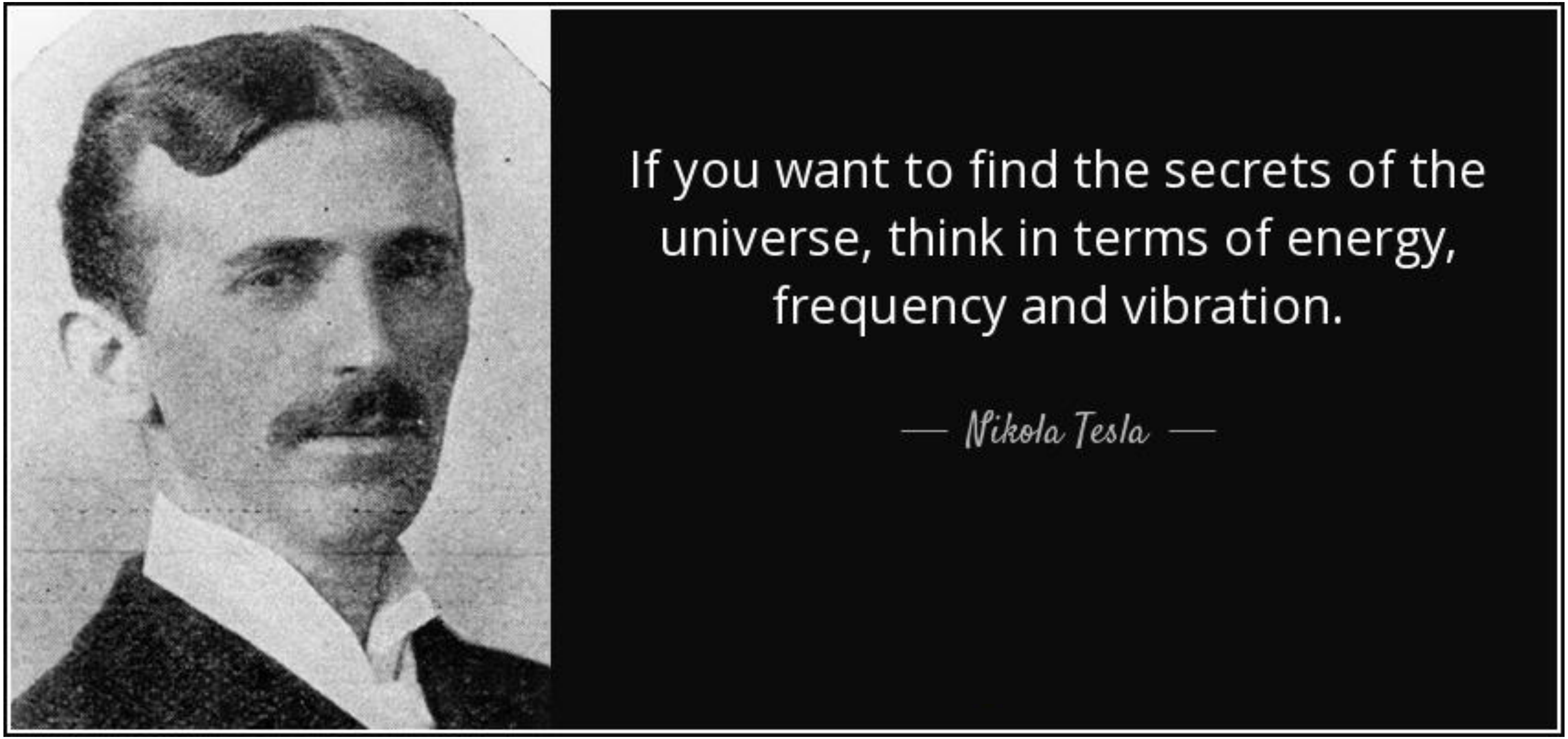
FOURIER (SPECTRAL) DOMAIN

```
plt.scatter(U[:,2], U[:,3], c=truth, cmap='plasma')
```

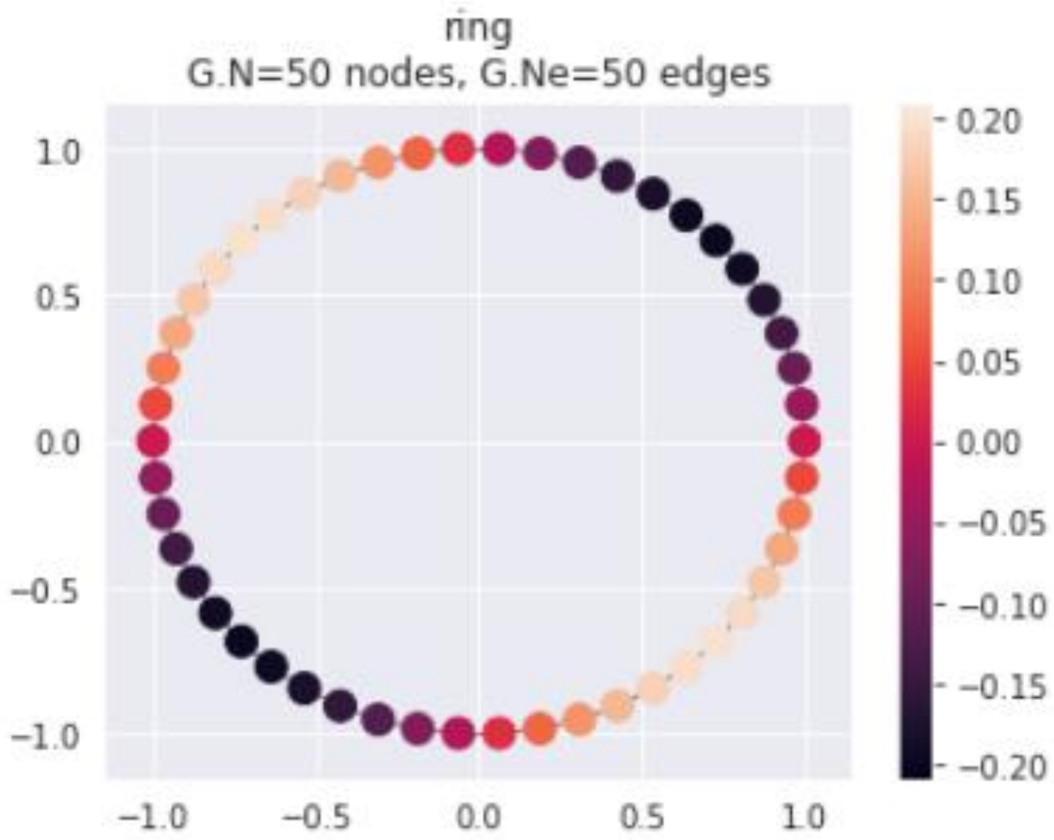
```
<matplotlib.collections.PathCollection at 0x7fc74a81b438>
```

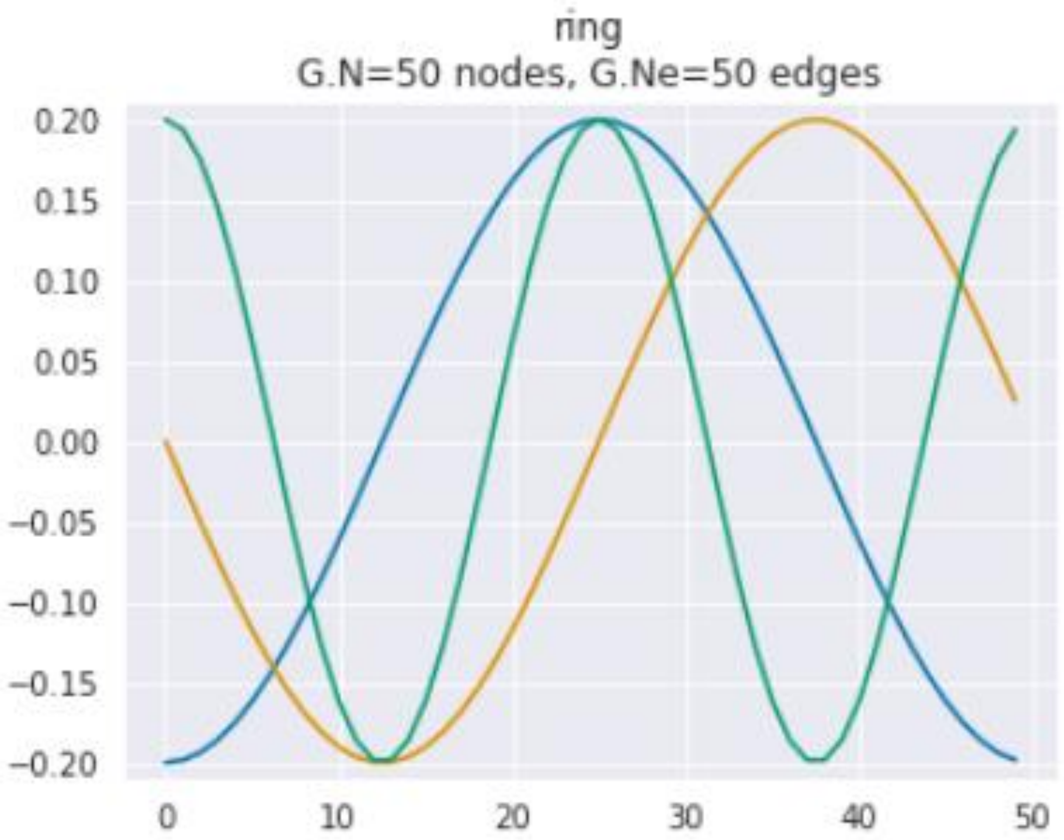


FOURIER BASIS OF A GRAPH



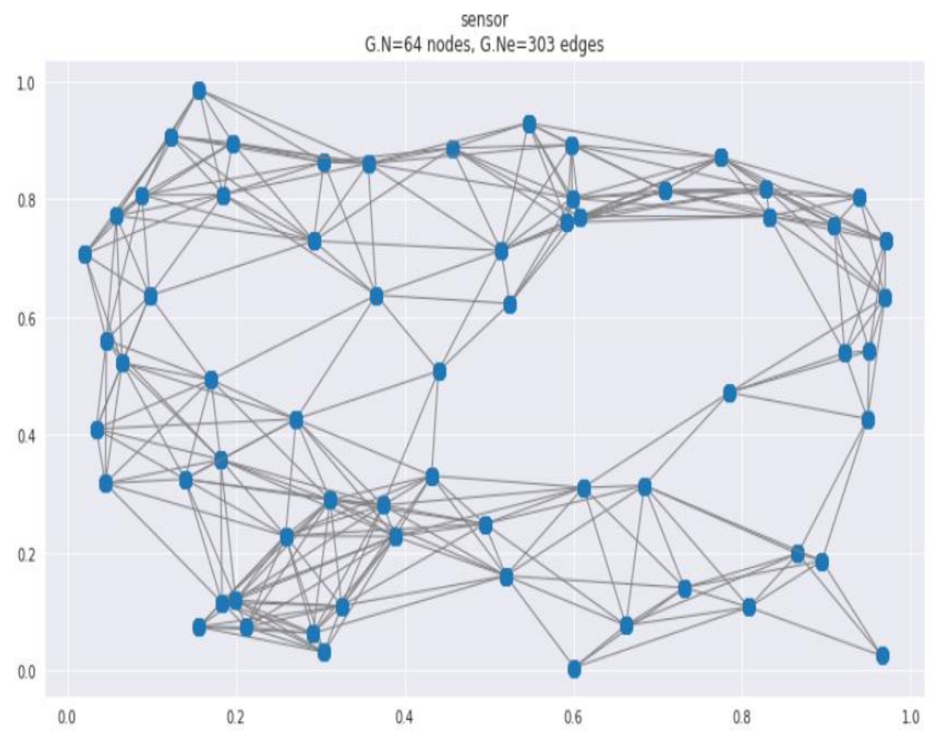
FOURIER BASIS OF A GRAPH





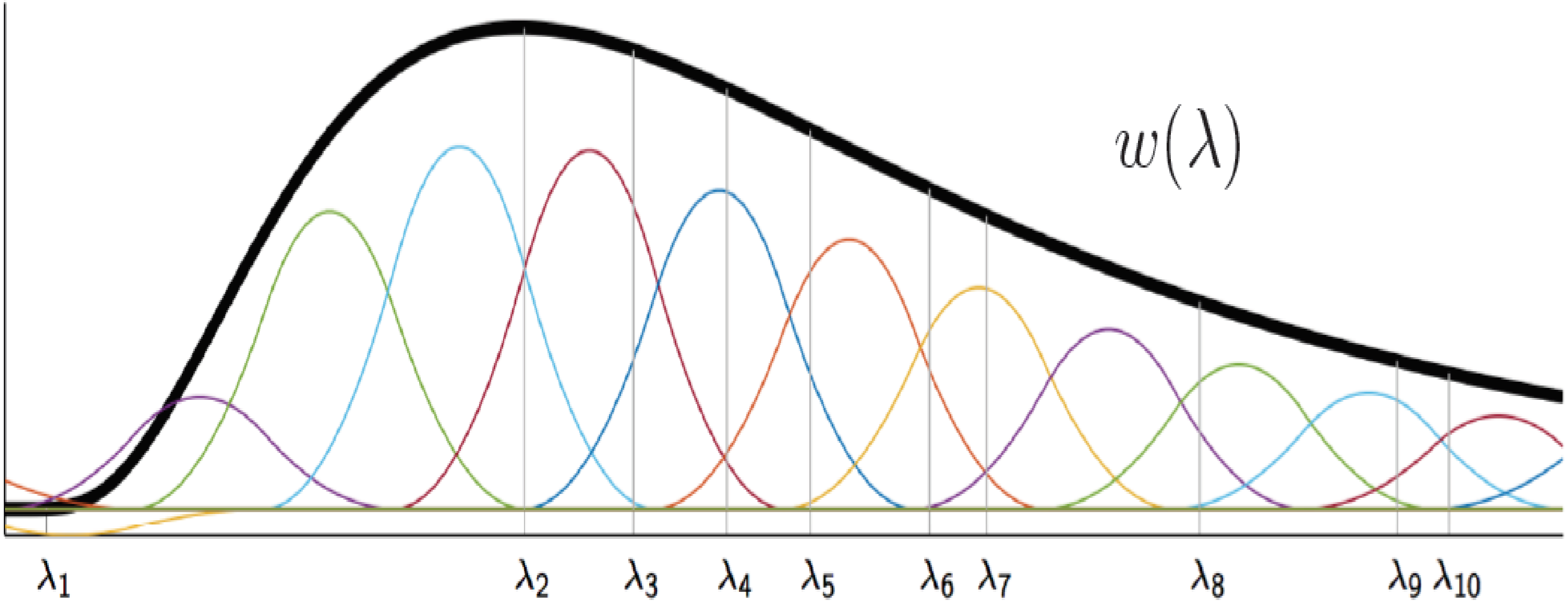
The first 4 eigenvectors

FOURIER BASIS OF A GRAPH



The first 13 eigenvectors

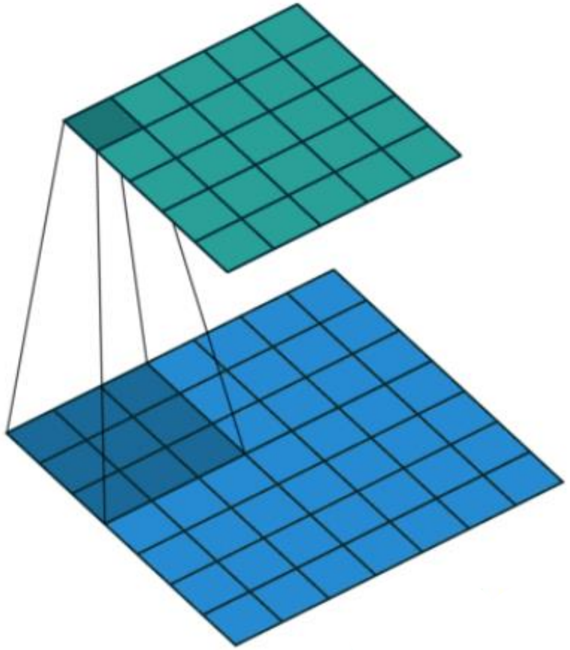
FOURIER BASIS OF A GRAPH

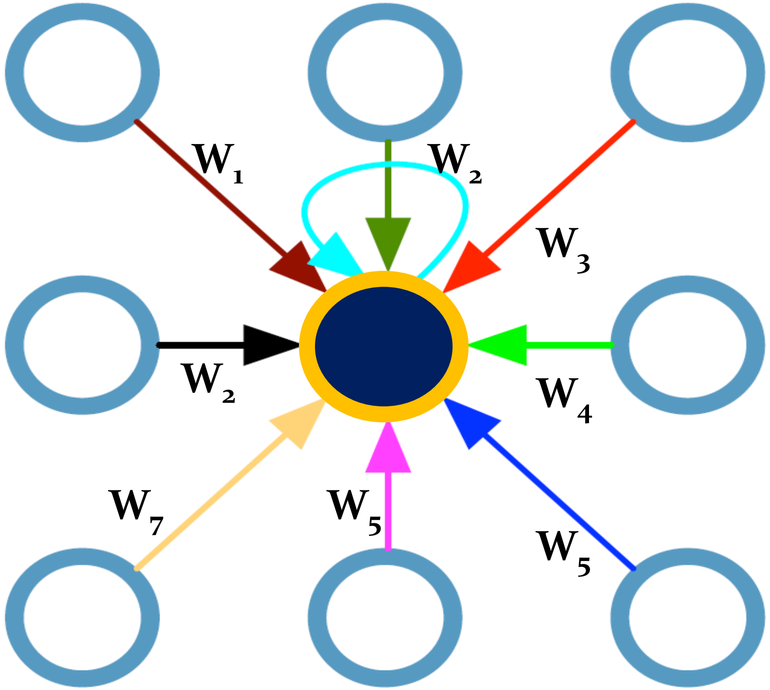


Source: Stanford Spectral Graph Theory

Graph Convolution Network (GCN)

RECALL – ON GRID

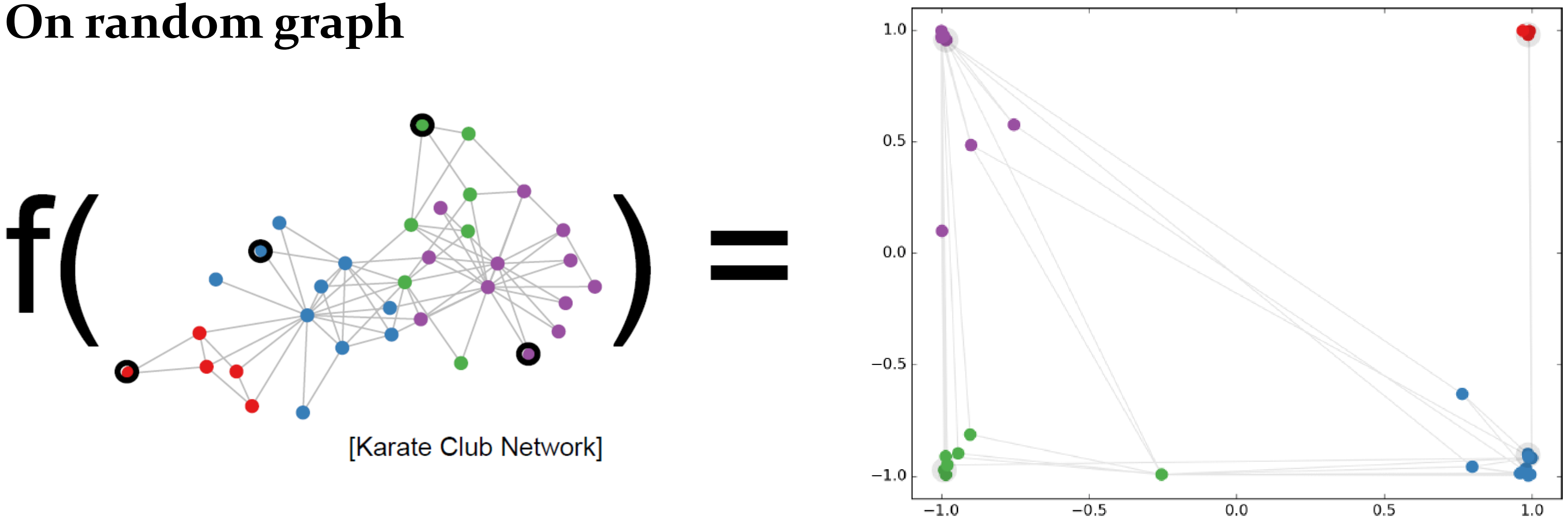




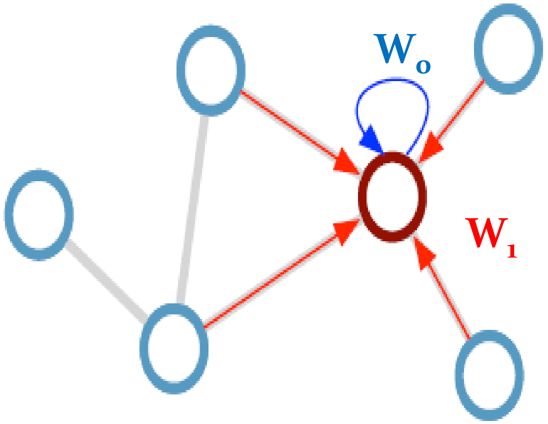
$$\mathbf{h}^{(l+1)} = \sigma\left(\mathbf{W}_0^{(l)}\mathbf{h}_0^{(l)} + \mathbf{W}_1^{(l)}\mathbf{h}_1^{(l)} + \cdots + \mathbf{W}_8^{(l)}\mathbf{h}_8^{(l)}\right)$$

SEMI-SUPERVISED LEARNING WITH GCN

On random graph



SEMI-SUPERVISED LEARNING WITH GCN

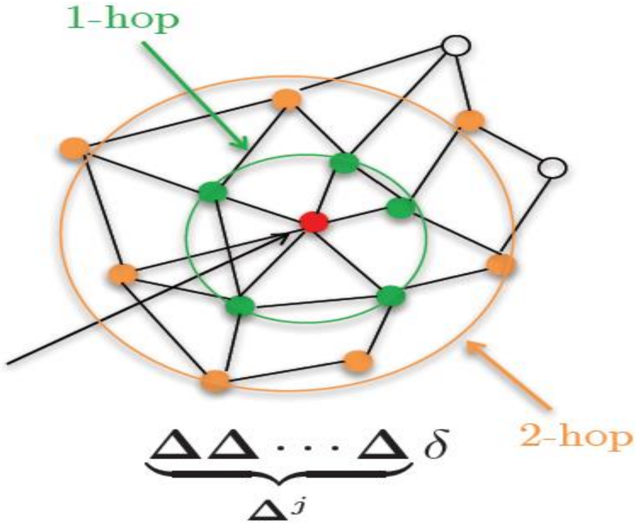


Update Rule: Localized 1st **Order Chebyshev Approximation** of Spectral Filter

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\mathbf{h}_i^{(l)}\mathbf{W}_0^{(l)} + \sum_{j\in\mathcal{N}_i}\frac{1}{c_{ij}}\mathbf{h}_j^{(l)}\mathbf{W}_1^{(l)}\right)$$

$\mathcal{N}_i$: neighbor indices

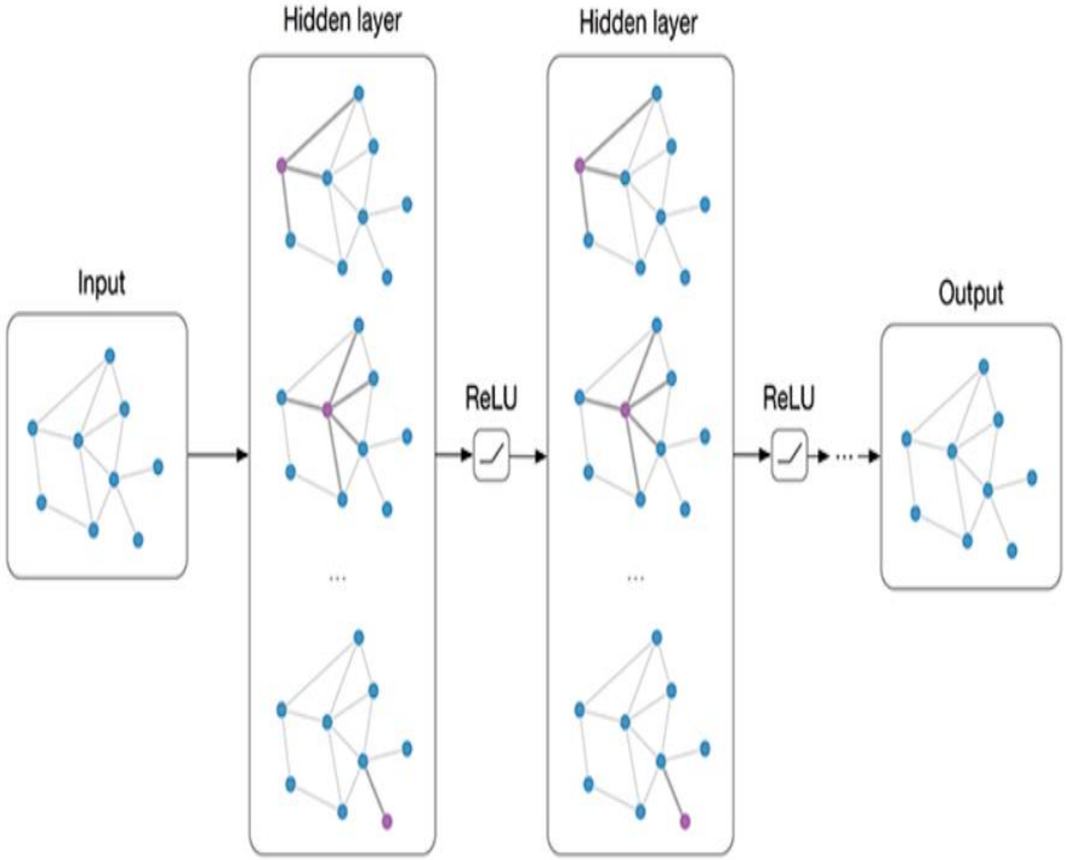
c_{ij} : norm. constant (per edge)



SEMI-SUPERVISED LEARNING WITH GCN

Vectorization Form:

$$\underbrace{H^{(l+1)}}_{\text{Output to next layer / result}} = \underbrace{\sigma}_{\text{3. Nonlinearity}}[\underbrace{\widehat{D}^{-\frac{1}{2}}\widehat{A}\,\widehat{D}^{-\frac{1}{2}}}_{\text{1. Normalize graph structure}}\underbrace{H^{(l)}W^{(l)}}_{\text{2. Multiply node properties and weights}}]$$



<u>Renormalize Trick: (Kipf, 2018)</u>

$$L_n = D^{-1/2}\, L\, D^{-1/2} = I_n - D^{-1/2} A\, D^{-1/2}$$

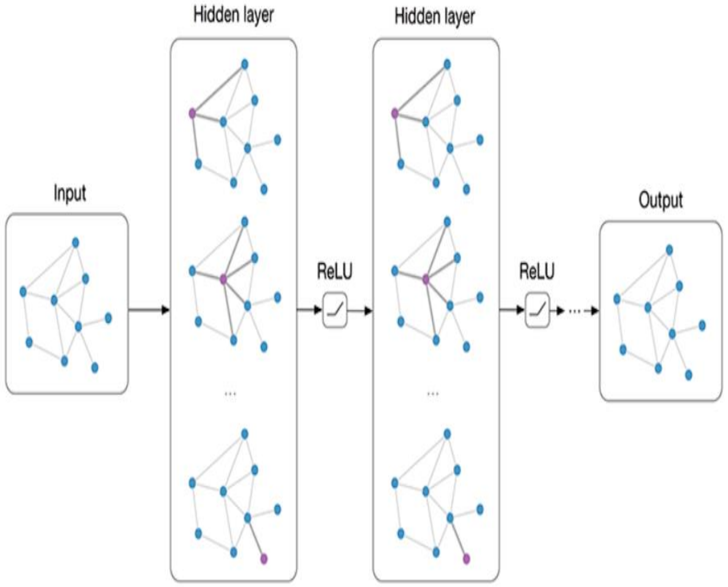
$$L_n = \tilde{D}^{-1/2}\,\tilde{A}\,\tilde{D}^{-1/2}$$

CONVOLUTION IN SPECTRAL DOMAIN

1. Get embedding for every nodes

2. Train classifier on every node

2. Evaluate loss on labels nodes only



$$\mathcal{L} = -\sum_{l \in \mathcal{Y}_L} \sum_{f=1}^{F} Y_{lf} \ln Z_{lf}$$

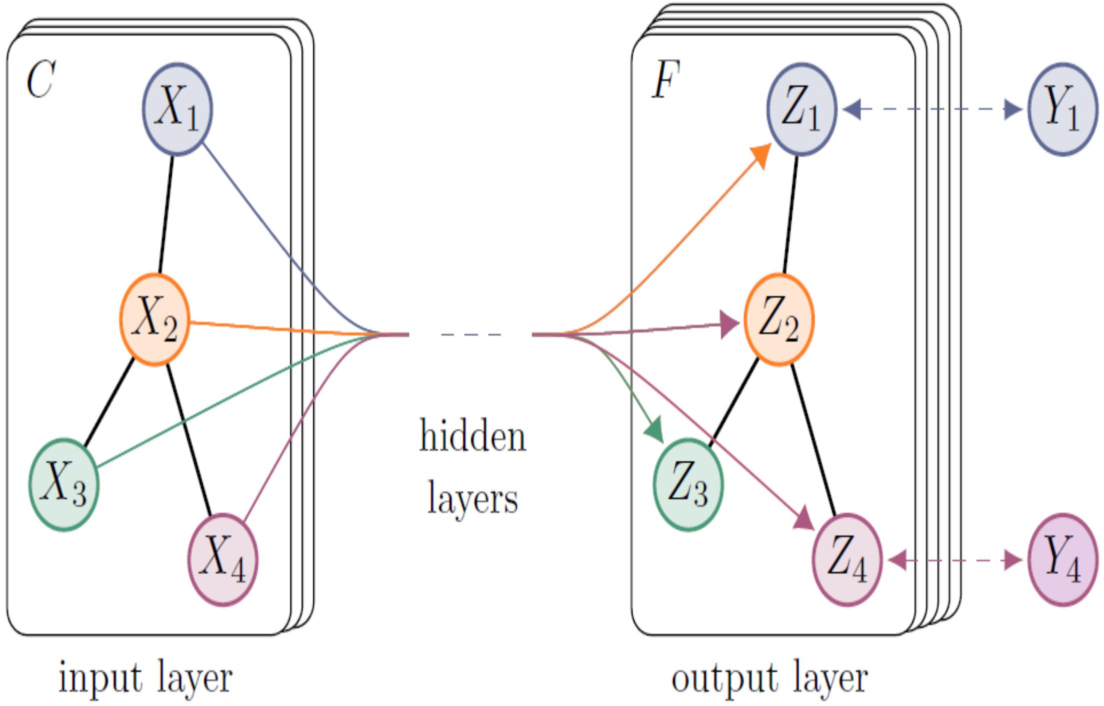
$\mathcal{Y}_L$ set of labeled node indices

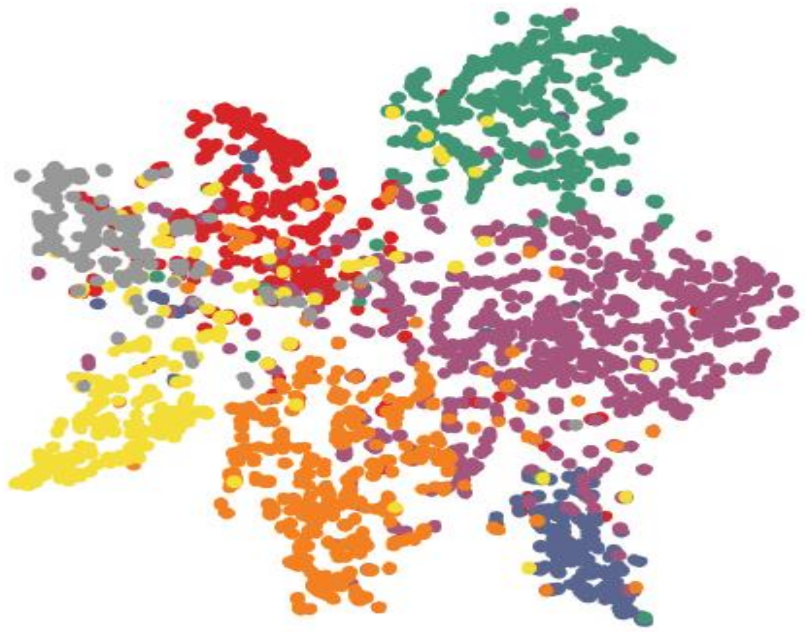
$\mathbf{Y}$ label matrix

$\mathbf{Z}$ GCN output (after softmax)

CONVOLUTION IN SPECTRAL DOMAIN

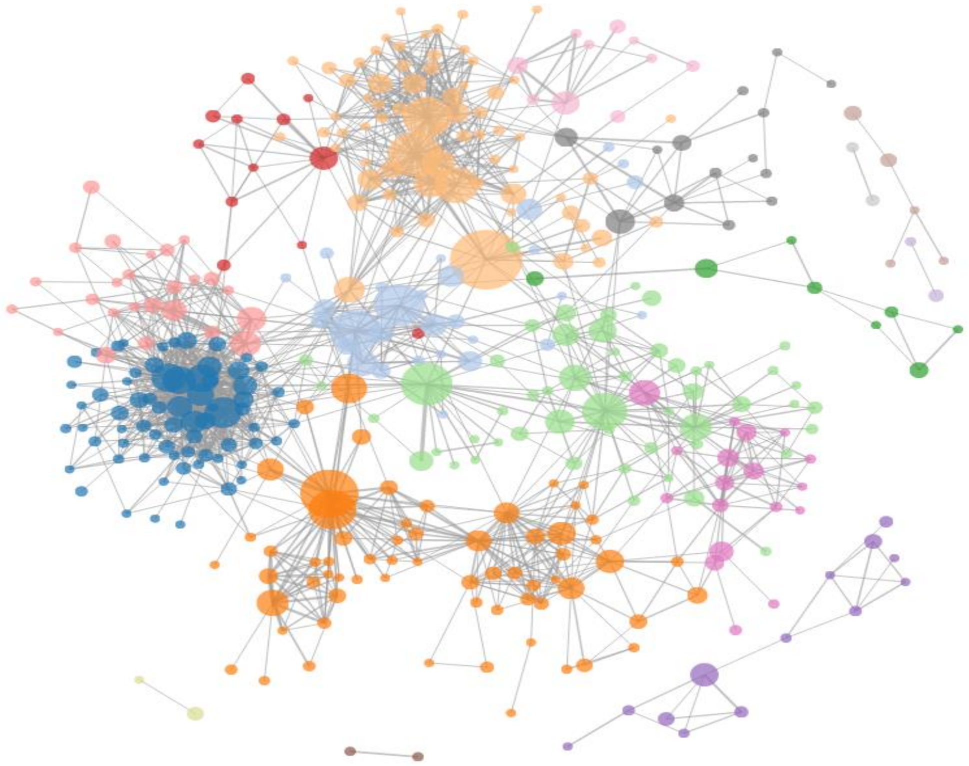
Example of a latent (embedded) representation coming out of a hidden layer:





PERFORMANCE ON CITATION NETWORK

Dataset	Type	Nodes	Edges	Classes	Features	Label rate
Citeseer	Citation network	3,327	4,732	6	3,703	0.036
Cora	Citation network	2,708	5,429	7	1,433	0.052
Pubmed	Citation network	19,717	44,338	3	500	0.003
NELL	Knowledge graph	65,755	266,144	210	5,414	0.001



Method	Citeseer	Cora	Pubmed	NELL
ManiReg [3]	60.1	59.5	70.7	21.8
SemiEmb [28]	59.6	59.0	71.1	26.7
LP [32]	45.3	68.0	63.0	26.5
DeepWalk [22]	43.2	67.2	65.3	58.1
ICA [18]	69.1	75.1	73.9	23.1
Planetoid* [29]	64.7 (26s)	75.7 (13s)	77.2 (25s)	61.9 (185s)
GCN (this paper)	**70.3** (7s)	**81.5** (4s)	**79.0** (38s)	**66.0** (48s)
GCN (rand. splits)	67.9 ± 0.5	80.1 ± 0.5	78.9 ± 0.7	58.4 ± 1.7

Thank You!

REFERENCES

1. CNN on Graphs, Xavier: http://helper.ipam.ucla.edu/publications/dlt2018/dlt2018_14506.pdf

2. Stanford – Mining Massive Datasets: http://snap.stanford.edu/class/cs246-2012/slides/11-graphs.pdf

3. Semi-supervised Learning with GCN, Thomas Kipf: https://arxiv.org/pdf/1609.02907.pdf

4. Graph Signal Processing: https://arxiv.org/pdf/1211.0053.pdf

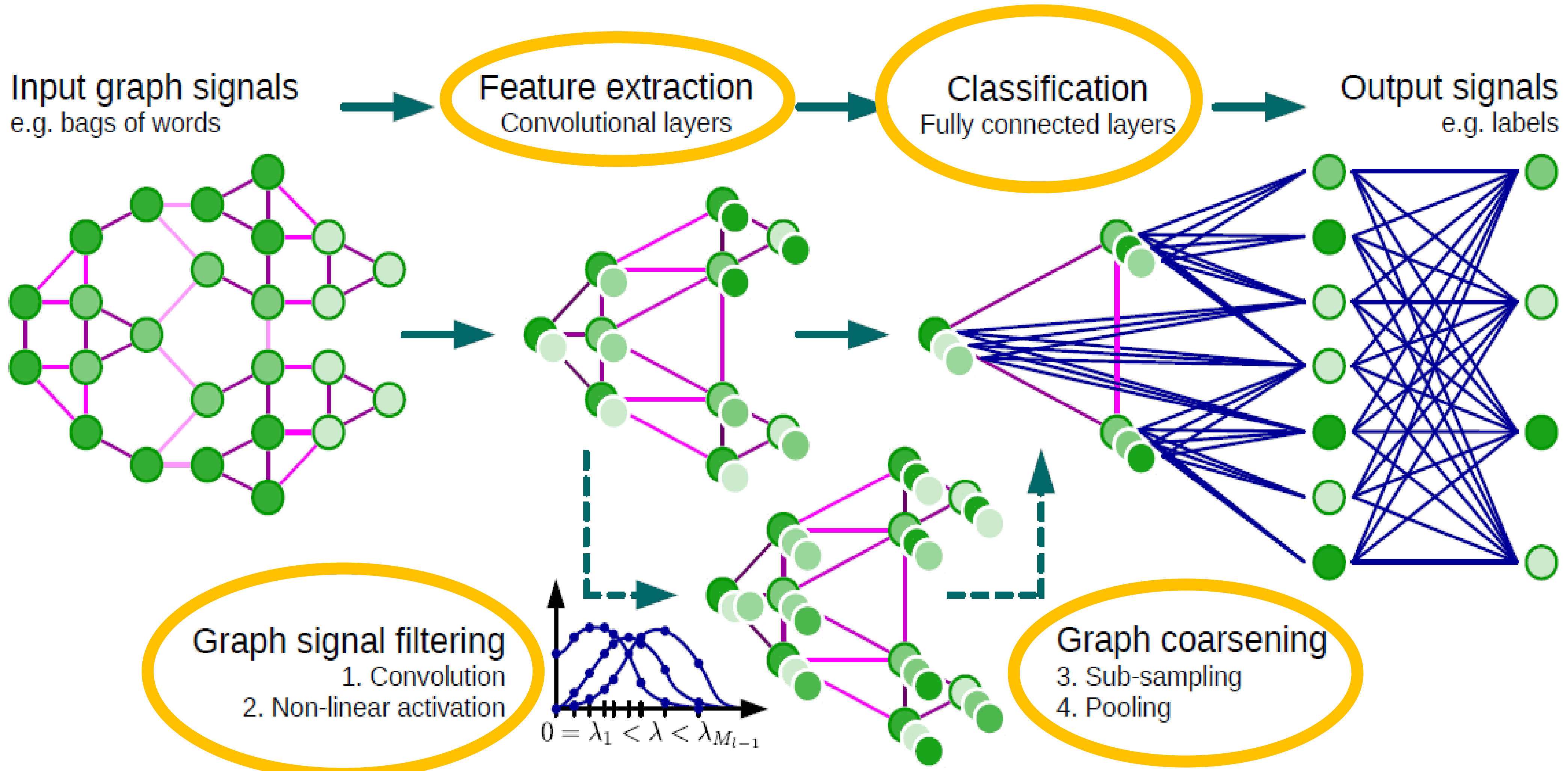
5. Udacity: High Performance Computing

6. Coursera: Graph Analytics for Big Data – UC San Diego

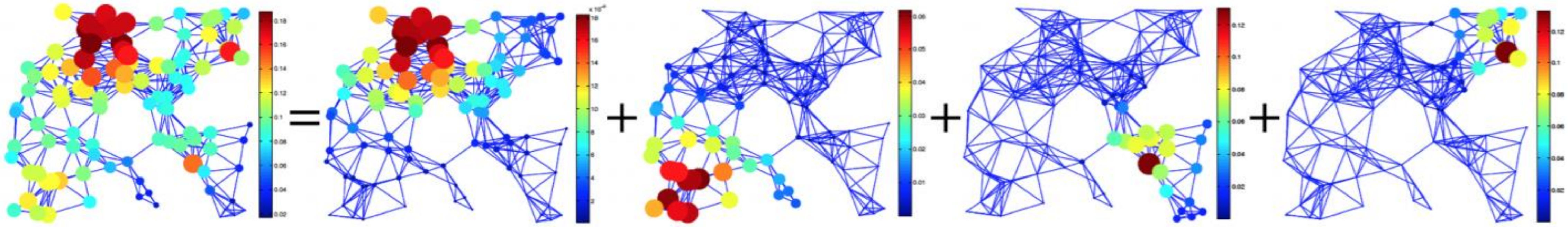
7. Coursera: Applied Social Network Analysis – University of Michigan

APPENDIX:

GCN ARCHITECTURE

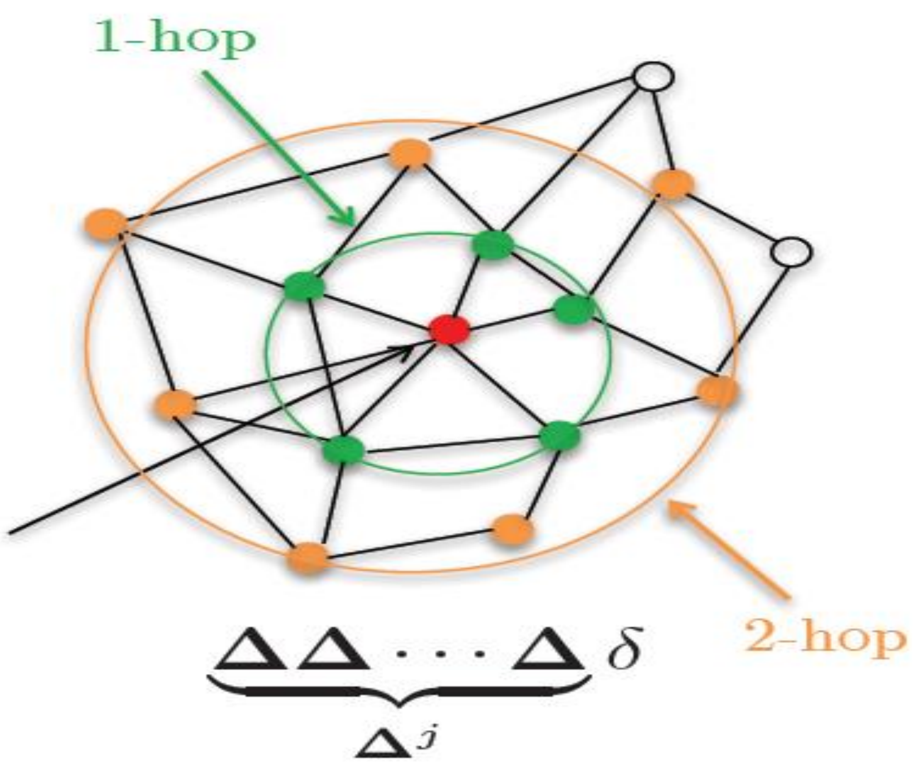


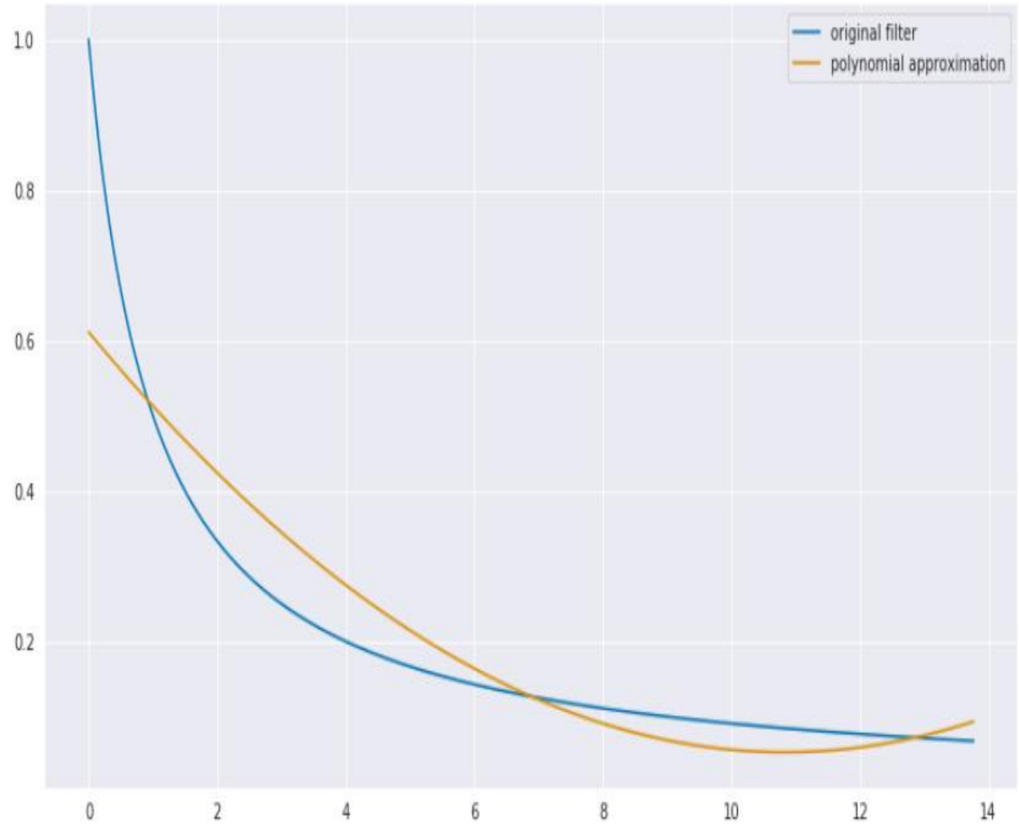
HEAT DIFUSION OF GRAPH



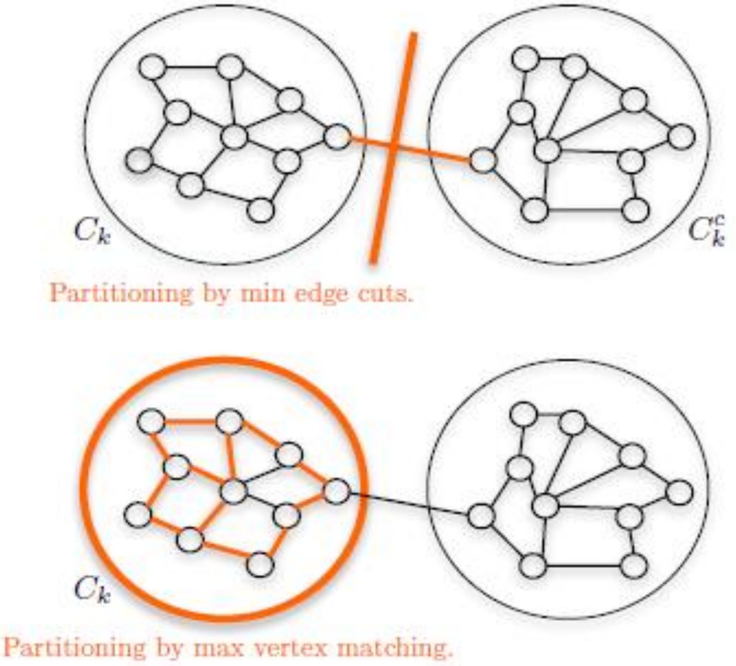
CONVOLUTION IN SPECTRAL DOMAIN

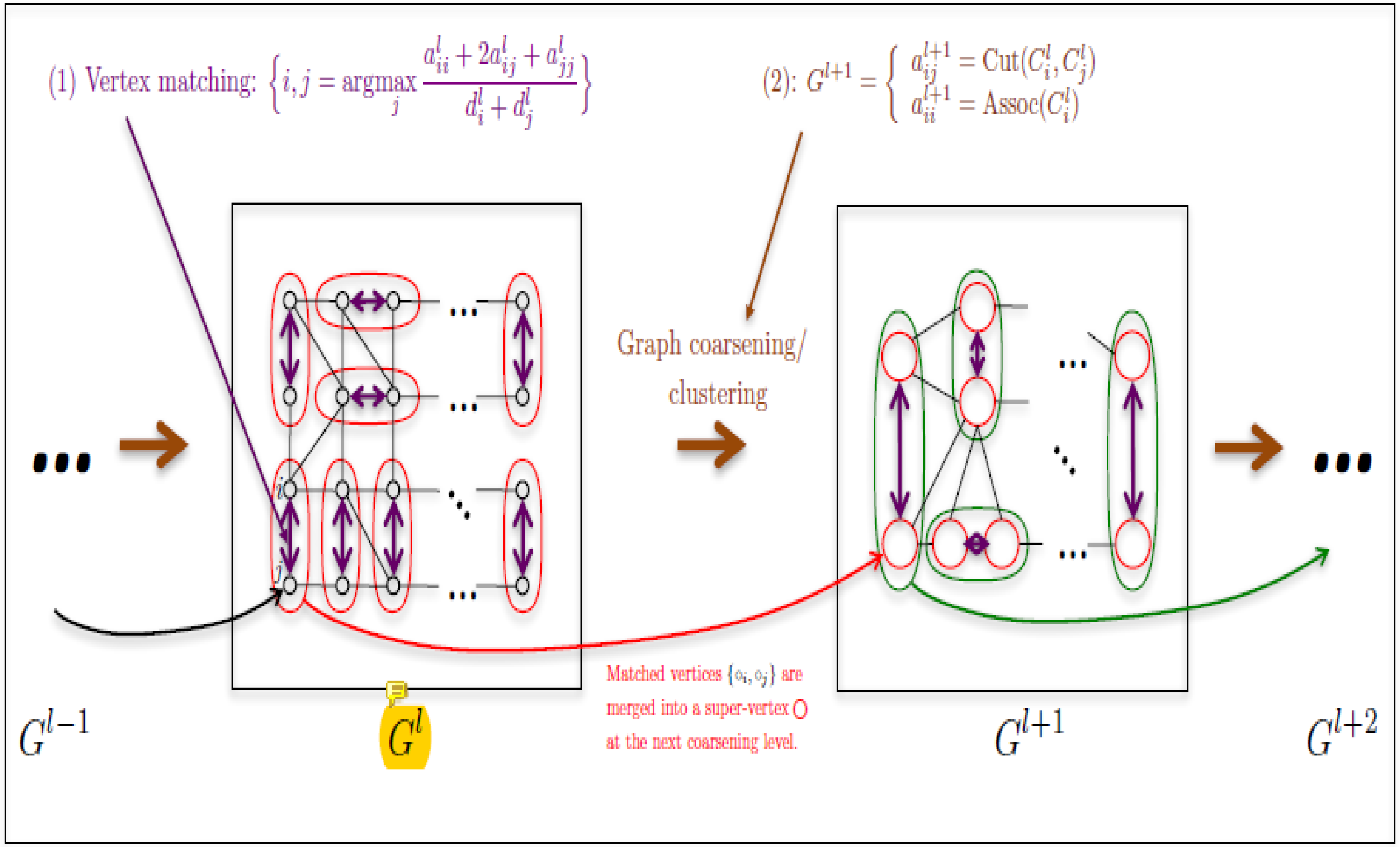
Chebyshev Approximation to Kth order where K << N



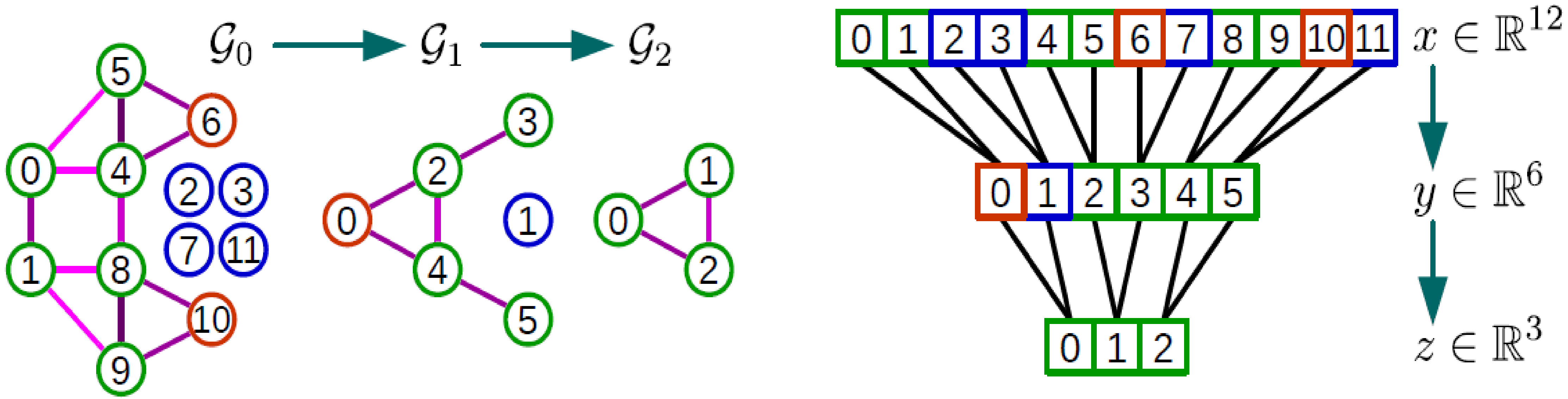


GRAPH COARSENING – MAX POOLING

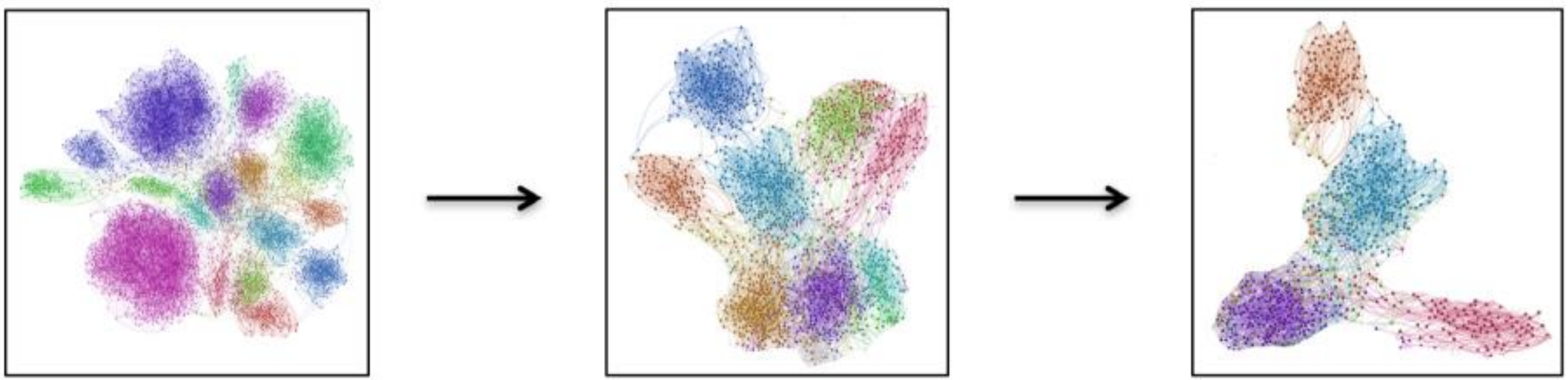




GRAPH COARSENING – MAX POOLING



GRAPH COARSENING – MAX POOLING



$G^{l=0} = G$ $G^{l=1}$ $G^{l=2}$

NP-Hard