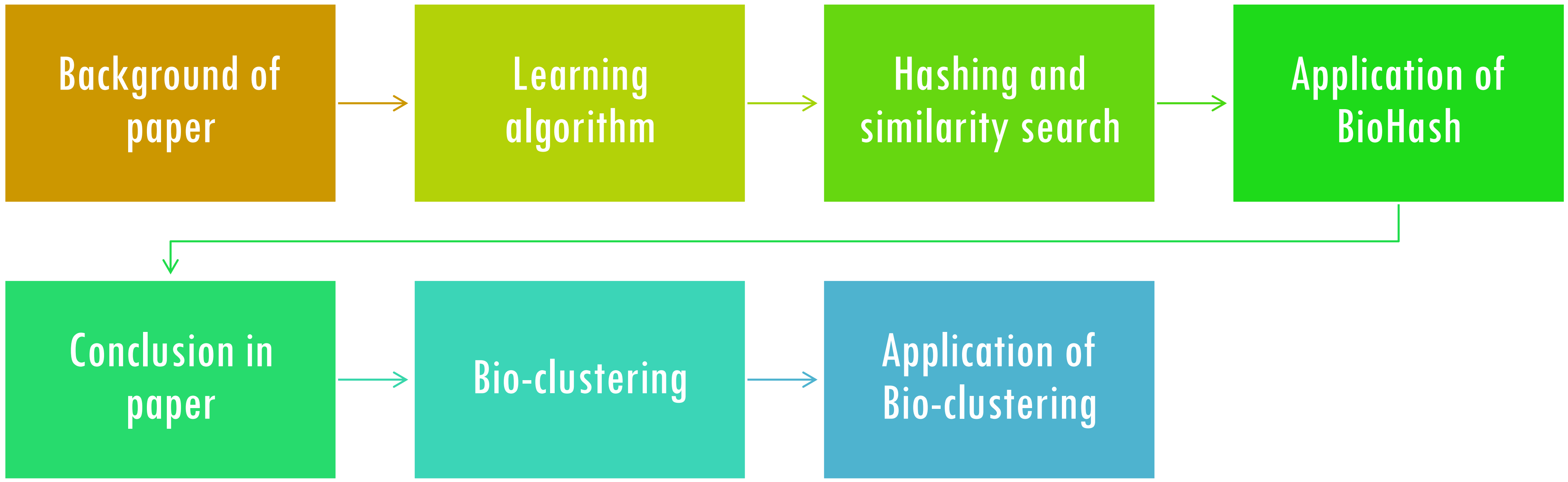
BIO-INSPIRED HASHING FOR UNSUPERVISED SIMILARITY SEARCH

Chaitanya K. Ryali
John J. Hopfield
Leopold Grinberg
Dmitry Krotov

GROUP 8

V S S Anirudh Sharma	EE18B036
Nune Sahithi	EE18B022
Bhargava M	EE18B112
Khyathi Sri K	EE18B137
Laashritha M	EE18B139

WE'LL TAKE YOU THROUGH…



THE NAME

- **BIO-INSPIRED**: Inspired from Drosophila fruit fly olfactory system.
  Biologically plausible.

- **HASHING**: Function Maps data of arbitrary size to fixed-size values, called hash codes

- **UNSUPERVISED**: Users do not need to supervise the model.
  Model discovers patterns and information that was previously undetected.

- **SIMILARITY SEARCH**: Giving out R data points like given input.

MOTIVATION

- The fruit fly Drosophila's olfactory circuit inspired **FlyHash Algorithm**
- ***Does not learn*** from Data, leaving scope for improvement
- Then came **SOLHash:** learns from data
- **Biologically Implausible**
- Not scalable due to **heavy computations**

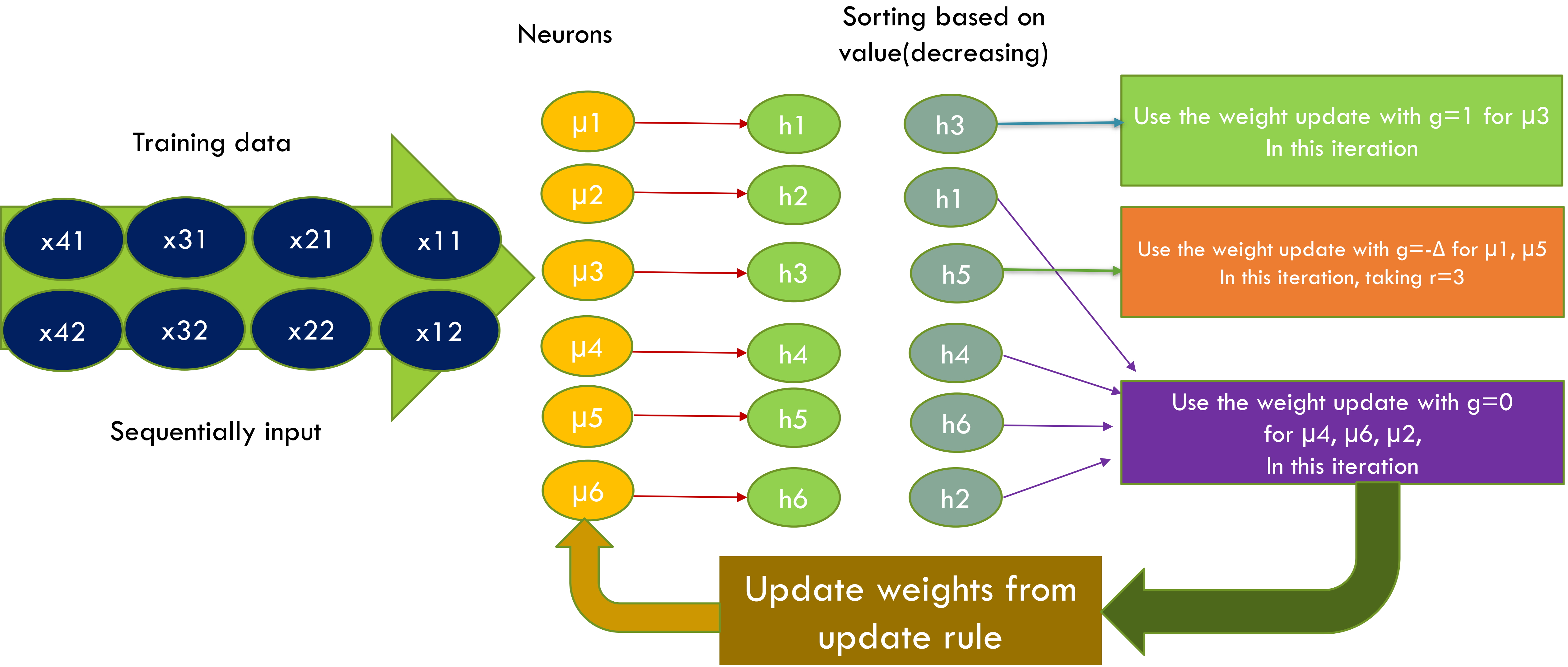
INTRODUCING BIO HASH

- Data driven
- Locality sensitive hashing algorithm
- Sparse high dimensional codes
- Biologically plausible
- Hebbian-like learning
- Useful In Unsupervised Similarity Search

THE ALGORITHM

A visual guide to the bio-inspired hashing and similarity search

Learning the weights



Where Wμ=[w1 w2]μ different for each neuron

LEARNING THE WEIGHTS

- Eqn-1: Synaptic plasticity rule (synaptic weight update)
- Eqn-2: For learning of the best activated unit and unlearning of rth units.
- Eqn-3: The energy function that gets minimized by this algorithm

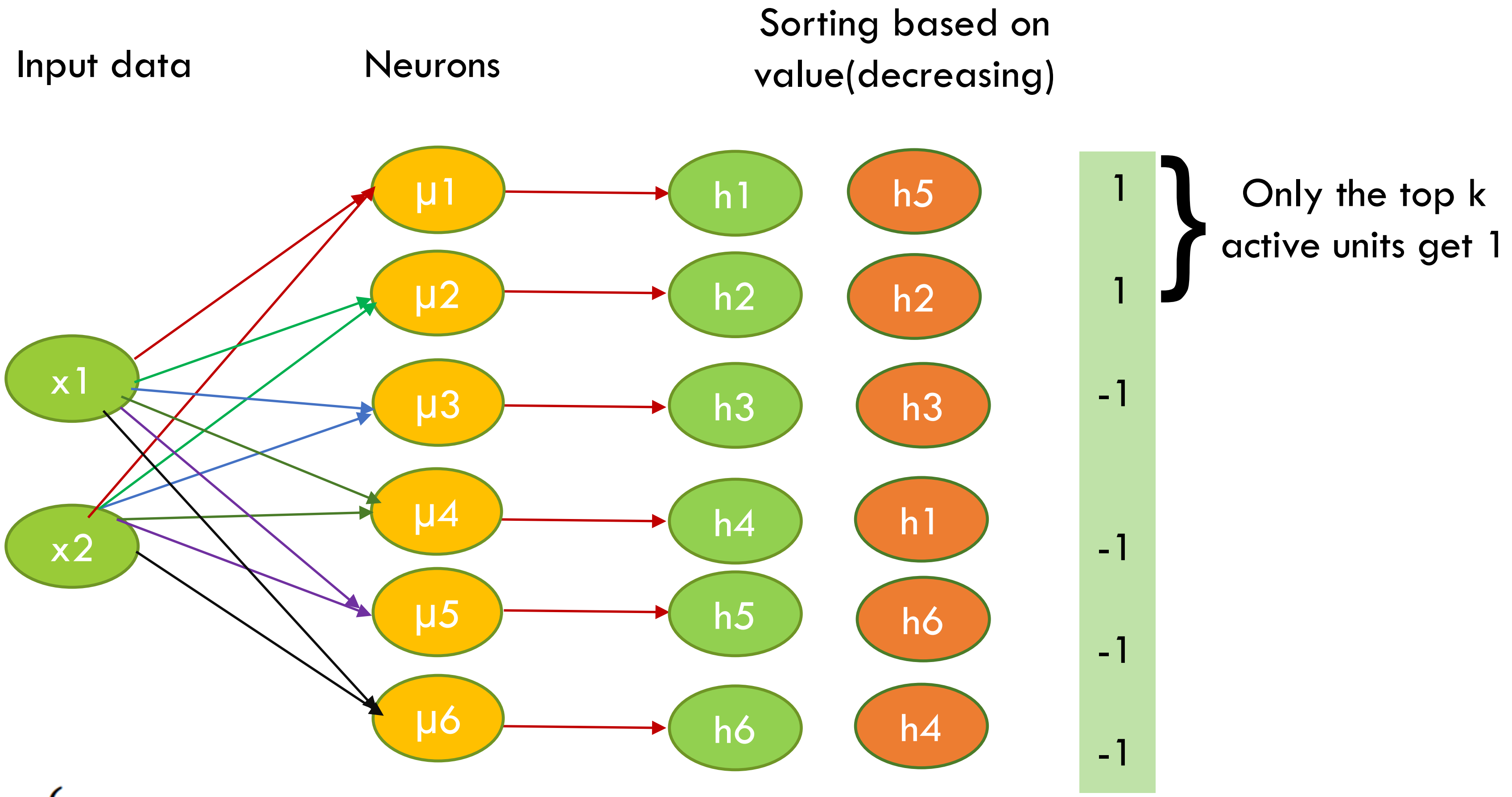
$$\tau \frac{dW_{\mu i}}{dt} = g\Big[\mathrm{Rank}\big(\langle W_\mu, x\rangle_\mu\big)\Big]\Big(x_i - \langle W_\mu, x\rangle_\mu W_{\mu i}\Big), \tag{1}$$

where $W_\mu = (W_{\mu 1}, W_{\mu 2} \ldots W_{\mu d})$, and

$$g[\mu] = \begin{cases} 1, & \mu = 1 \\ -\Delta, & \mu = r \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$E = -\sum_A \sum_{\mu=1}^{m} g\Big[\mathrm{Rank}\big(\langle W_\mu, x^A\rangle_\mu\big)\Big] \frac{\langle W_\mu, x^A\rangle_\mu}{\langle W_\mu, W_\mu\rangle_\mu^{\frac{p-1}{p}}}, \tag{3}$$

Hash coding



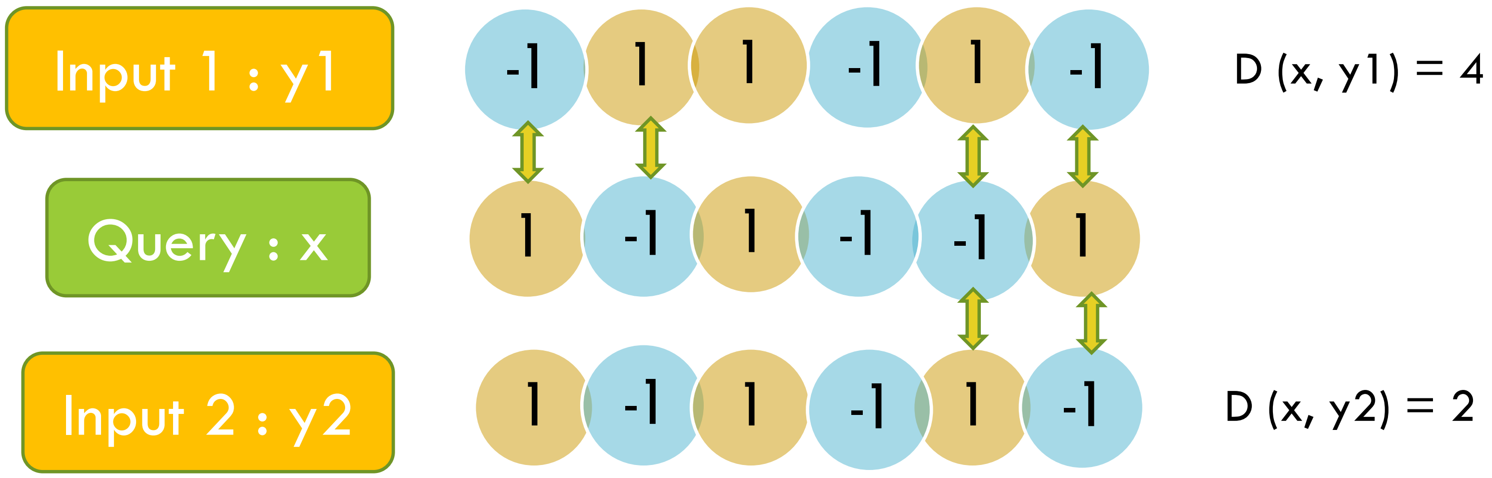
$$y_\mu = \begin{cases} 1, & \langle W_\mu, x \rangle_\mu \text{ is in top } k \\ -1, & \text{otherwise.} \end{cases} \qquad (6)$$

Finally hash code for h(x) = [h1 h2 h3 h4 h5 h6] = [-1, +1, −1, −1 ,+1, −1]

Where Wμ=[w1 w2]μ different for each neuron

SIMILARITY SEARCH

- This is a Locality Sensitive Hashing
- We select top R results from total inputs with the **least hamming distance of hash codes** with the given input to be searched against.



APPLICATIONS OF SIMILARITY SEARCH

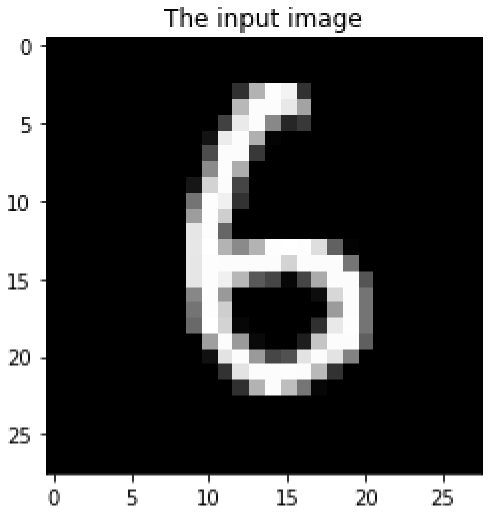
- Face Identification
- Plagiarism check
- Gene impression similarity
- Comparing Fingerprints
- **Handwriting matching**

APPLICATION: HANDWRITING MATCHING

Finding similar handwritten numbers

RESULT FROM CODE

- 150 search inputs
- Efficiency per input = right_outputs / total_outputs
- Average efficiency achieved =89.555%



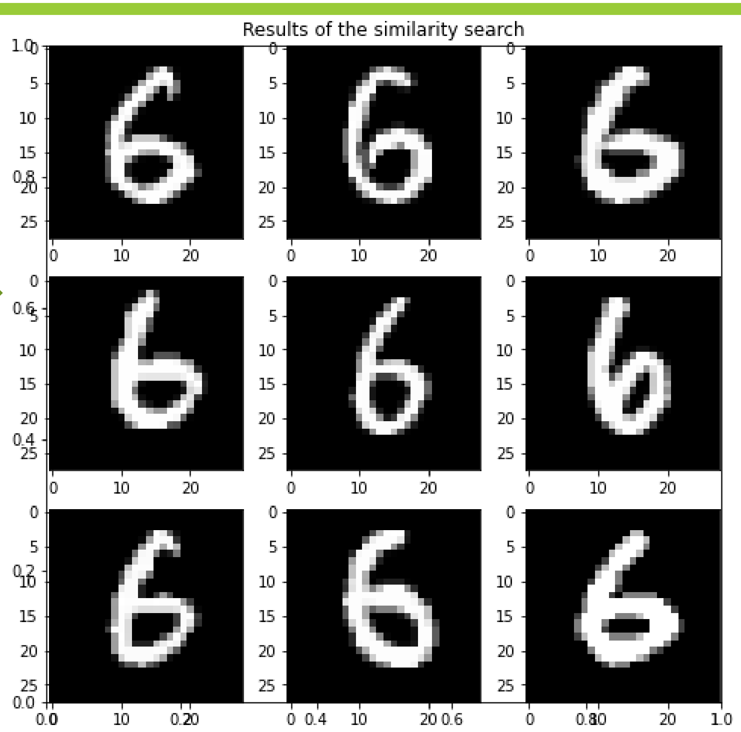
6000 training inputs from MNIST

1000 neurons

Activity = 40%

Search input not in this training set

9 most similar results from training set



INTUITION

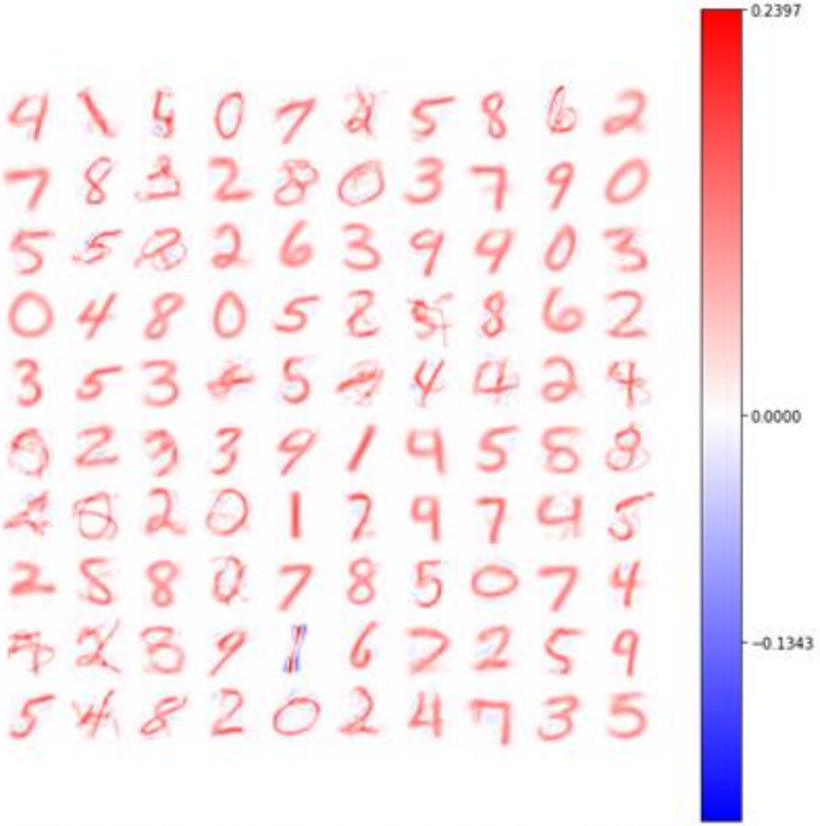
Putting the equations aside

HEURISTICS

Synaptic Weights: Particles moving in a data space

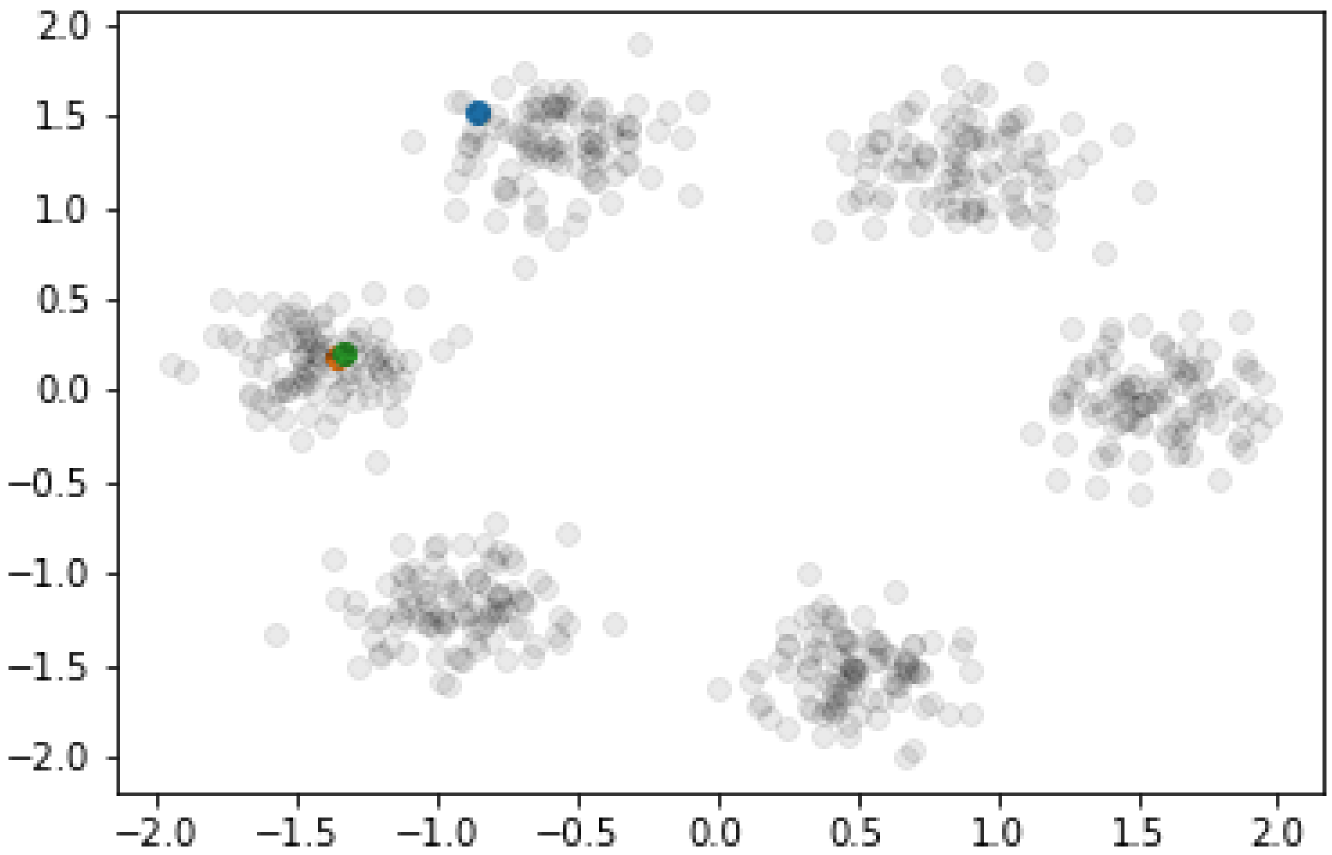
Acting forces:

- Attraction: to the peaks of the data distribution (**Cooperation**)
- Repulsion: between the hidden units (**Competition**)



HEAT MAP OF LEARNED WEIGHTS

HOW DO SYNAPSES SETTLE IN DATA SPACE?

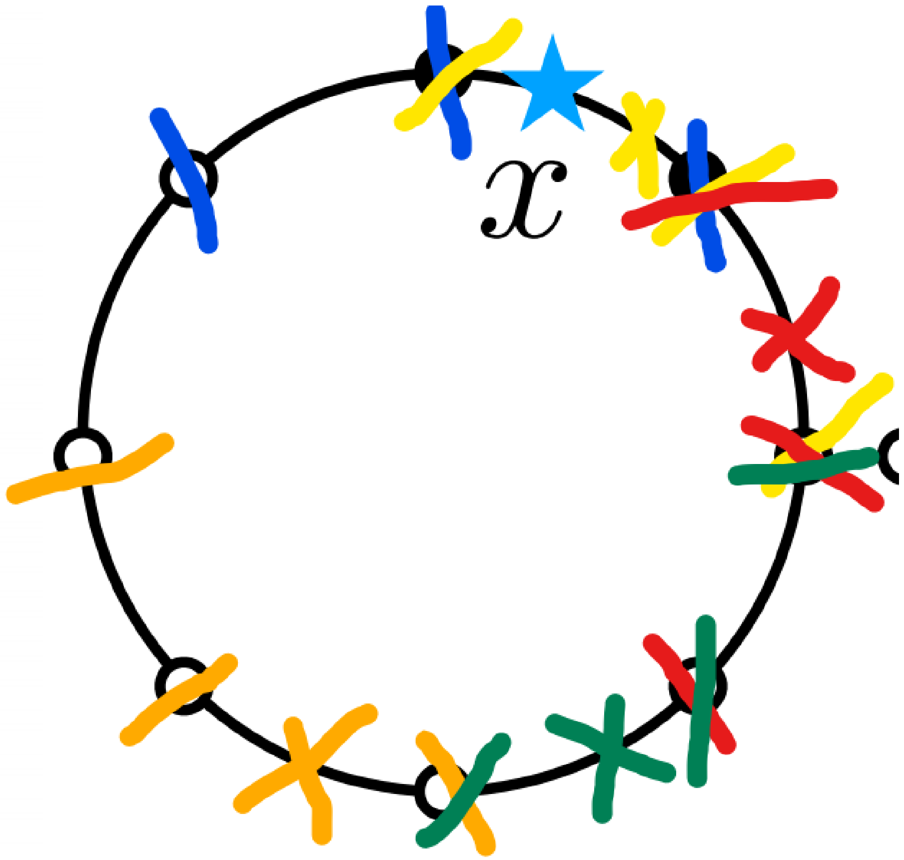


CONCLUSION IN PAPER

What the paper says

CONCLUSIONS IN PAPER

- BioHash better preserves local distances over global distances.
- **Effect of sparsity:**
  - There's optimal level of activity for each dataset (∝ k/m).
  - at lower sparsity levels, dissimilar images may
    become nearest neighbors though highly dissimilar images stay apart.



CONCLUSIONS IN PAPER

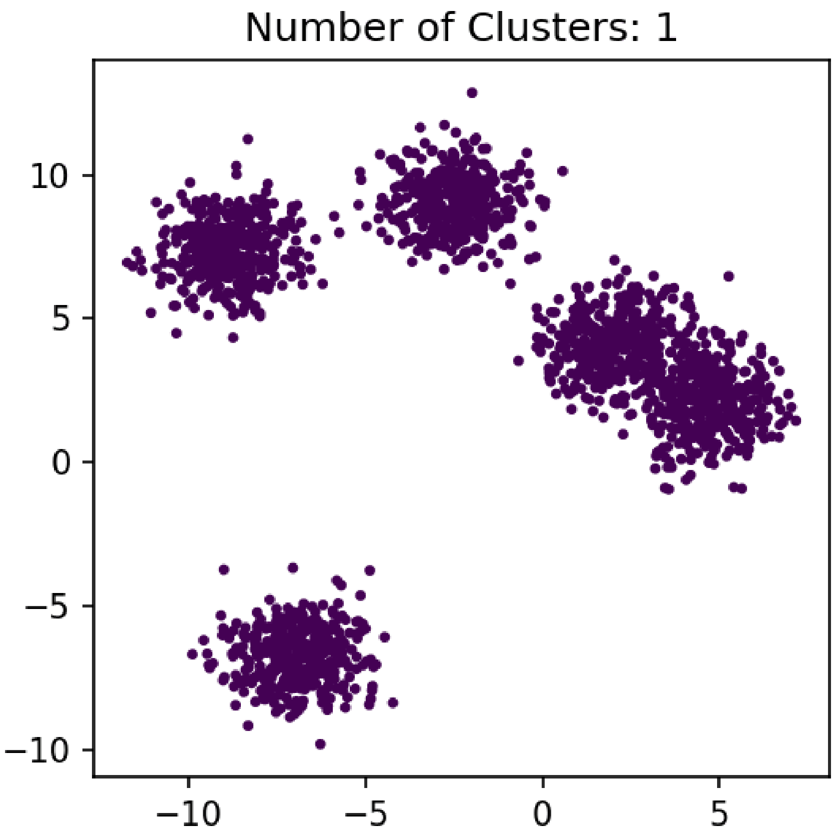
- Divisive normalization into learning to hash methods improves robustness to local intensity variations
- The biological plausibility of this work provides support toward the proposal that LSH might be the neural circuit algorithm featuring sparse expansive representations.
- **This can be used for Spherical k means when p=2 (Simple inner product) and Δ= 0. (No Anti-hebbian update)**

BIO-CLUSTERING

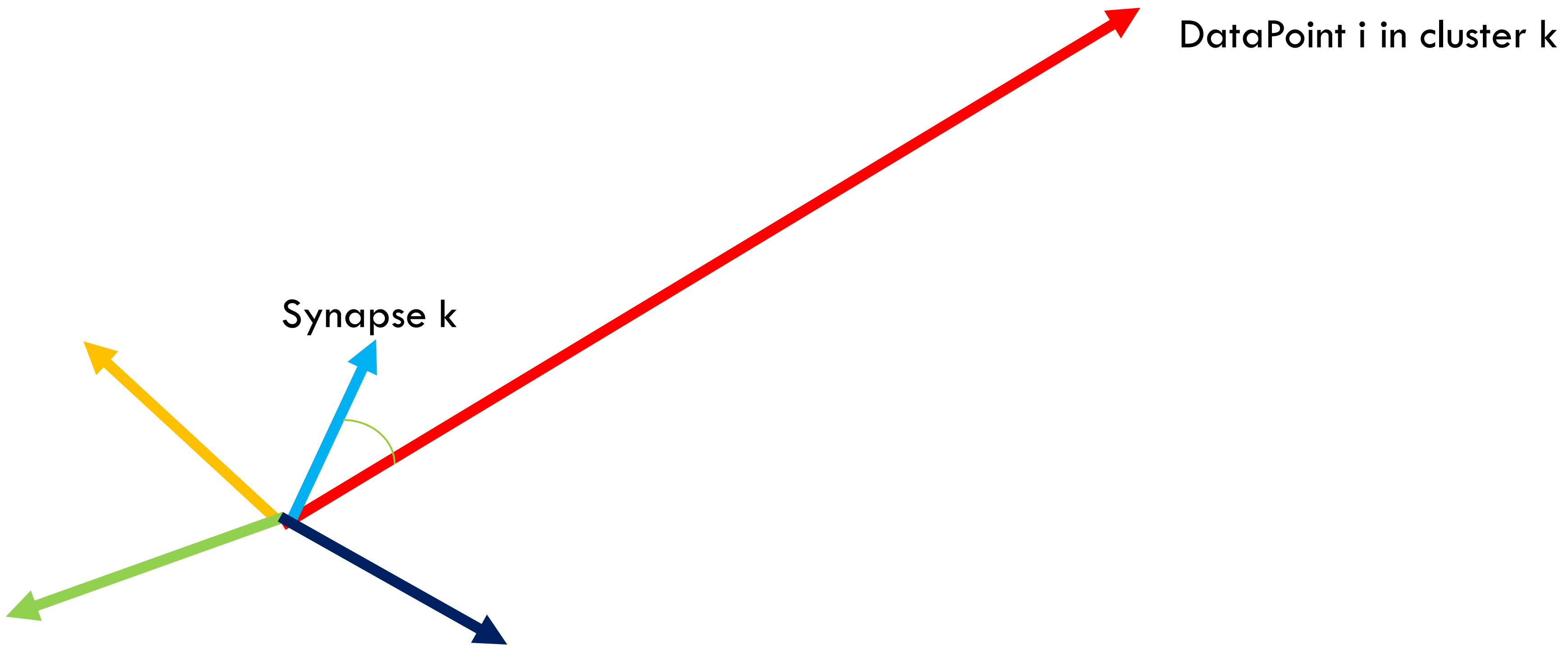
Using the learning algorithm for a different purpose

WHAT IS CLUSTERING?

- Unsupervised machine learning method
- Identifying and grouping similar data points in larger datasets without concern for the specific outcome.
- Usually used to classify data into structures that are more easily understood and manipulated



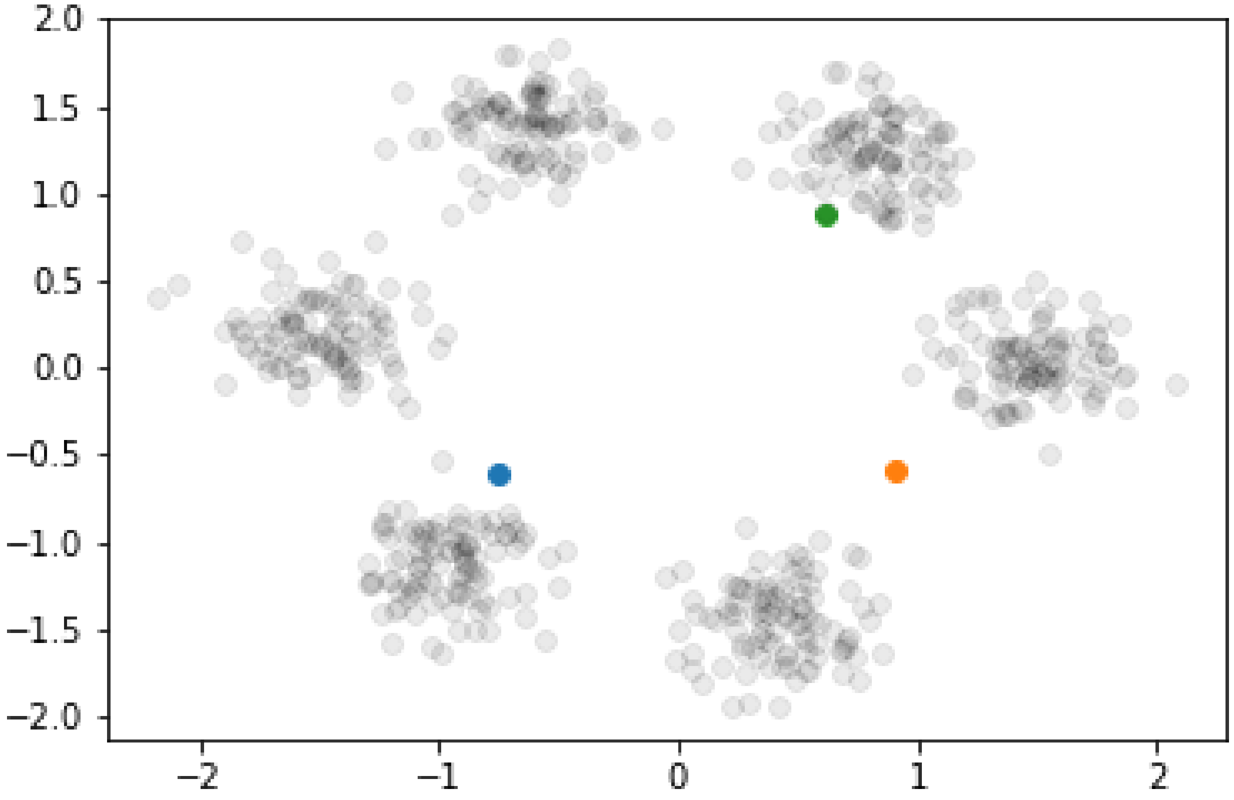
BIO CLUSTERING

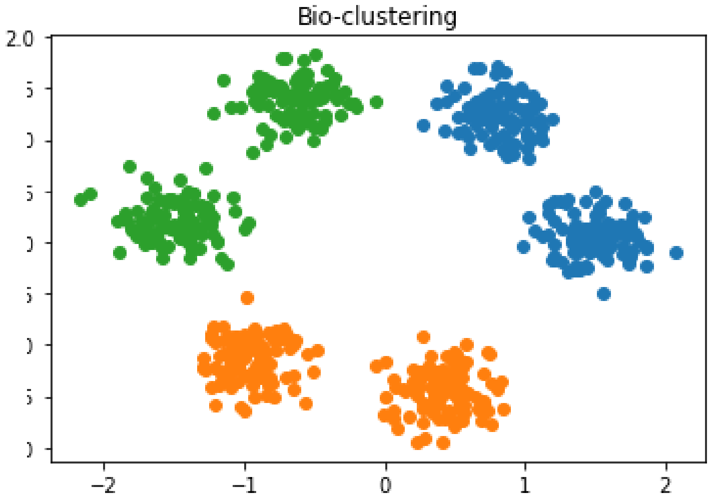


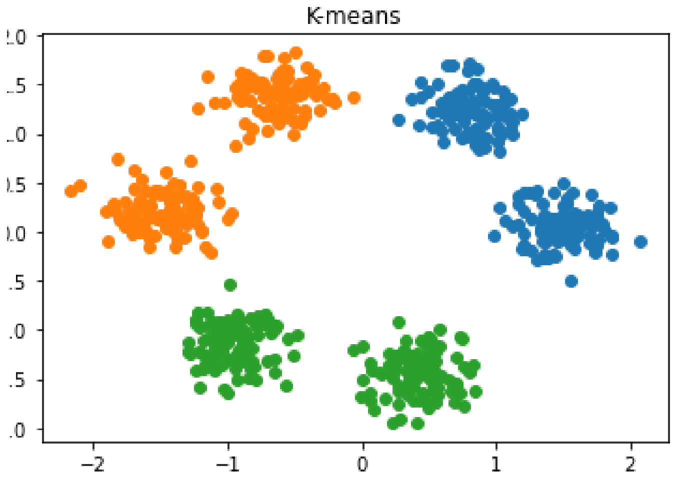
VISUALIZING BIO-CLUSTERING

Observe how the synapses behave

USING THIS FOR CLUSTERING



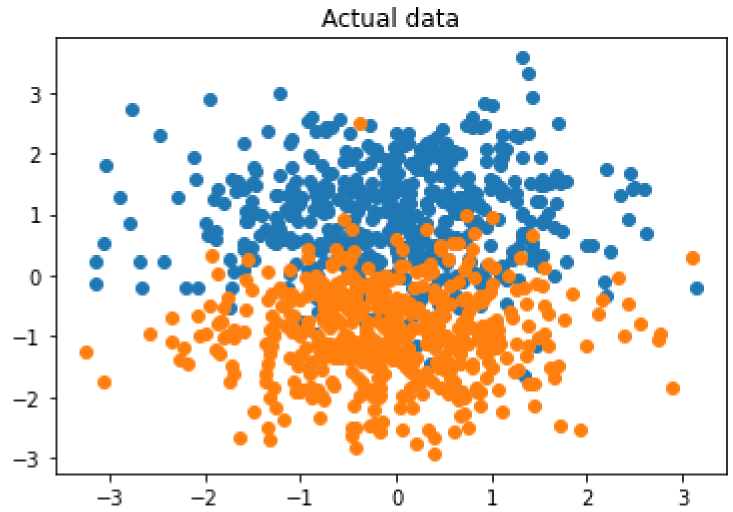




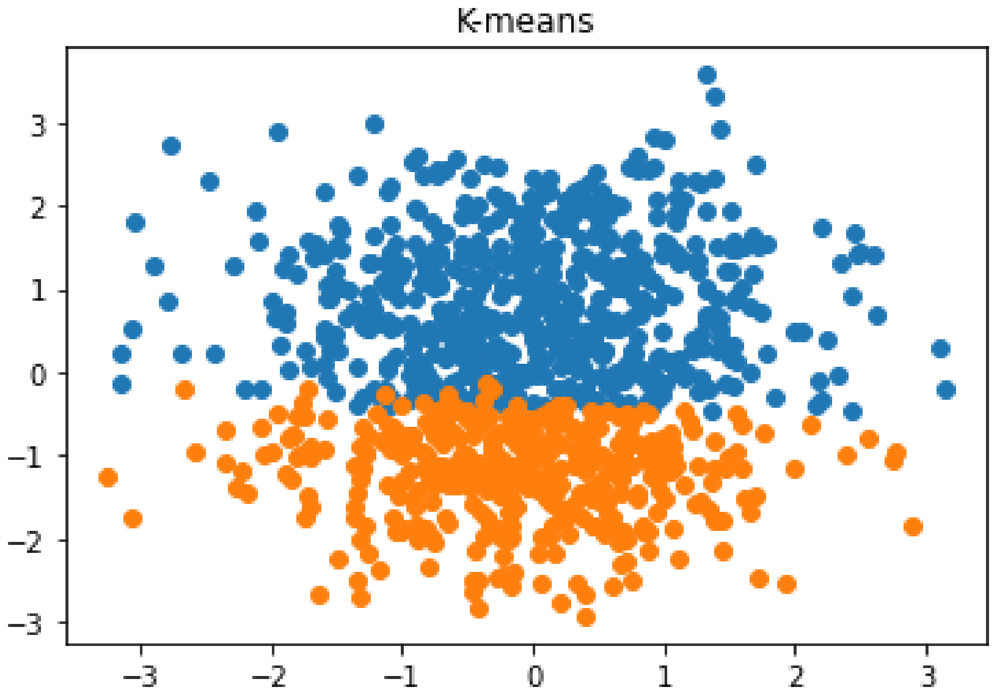
BIO-CLUSTERING VS K-MEANS CLUSTERING

For a standard to compare against

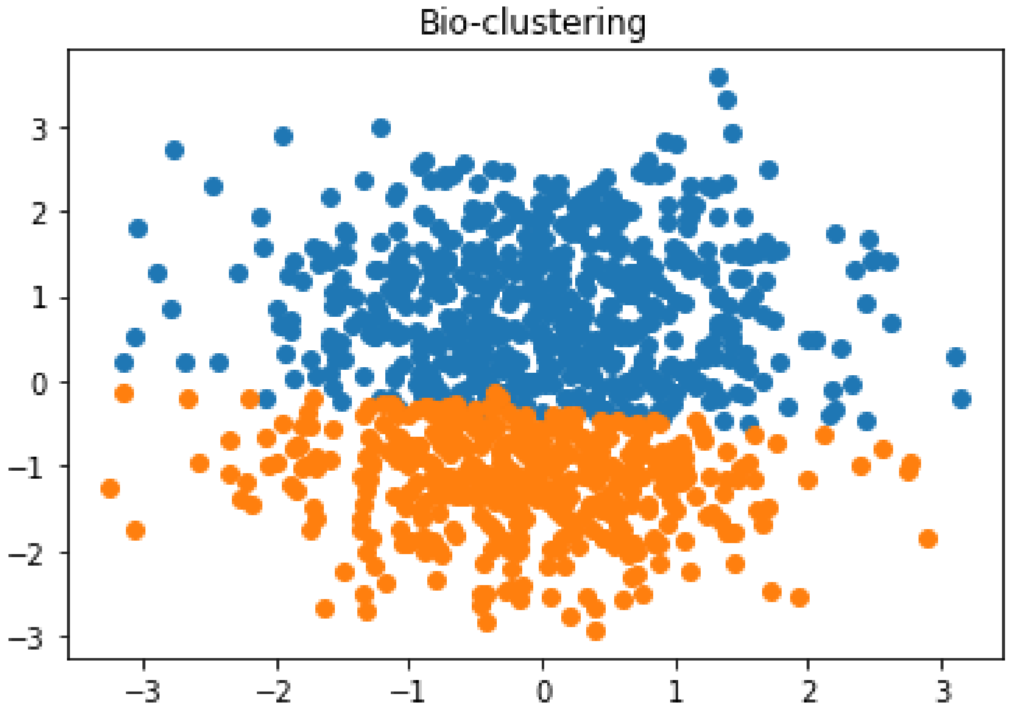
RANDOM



Actual data
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

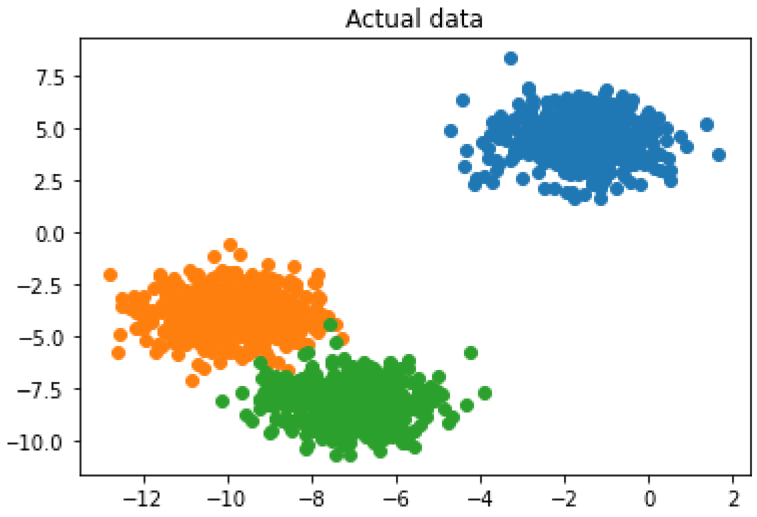


K-means
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

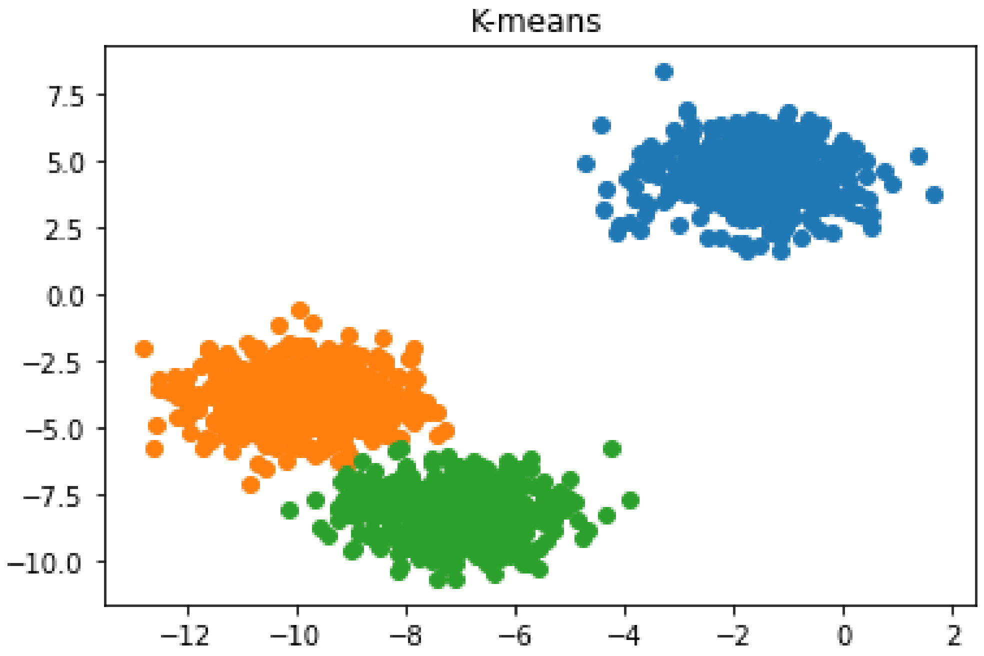


Bio-clustering
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

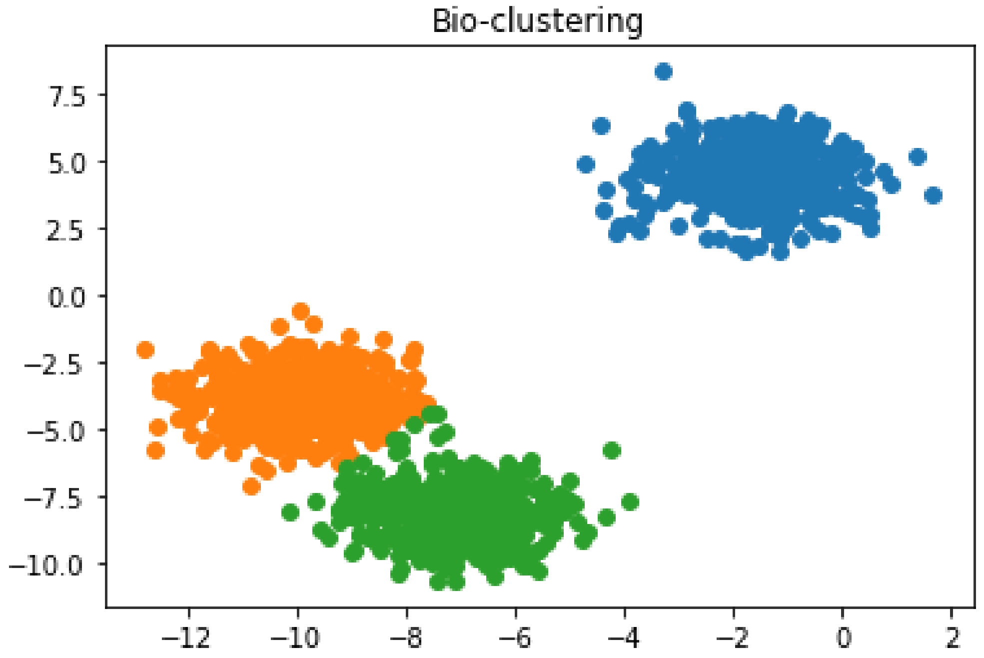
BLOBS



Actual data
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−12
−10
−8
−6
−4
−2
0
2

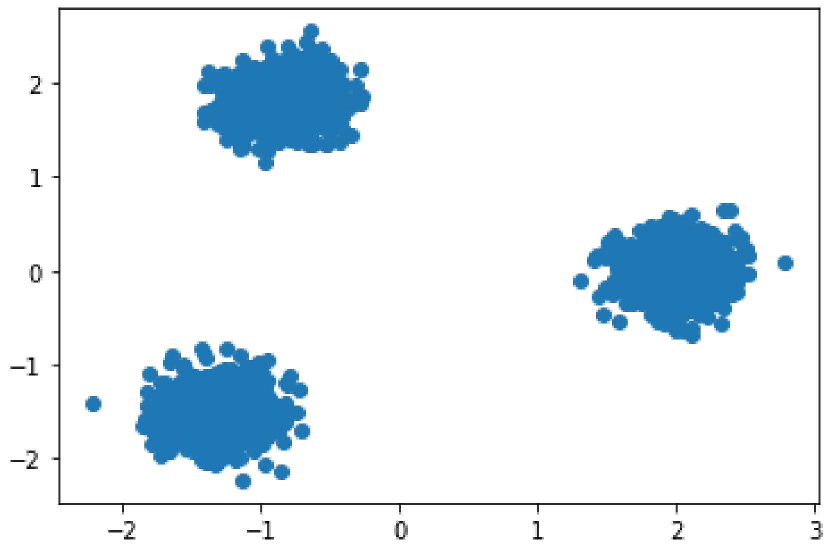


K-means
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−12
−10
−8
−6
−4
−2
0
2

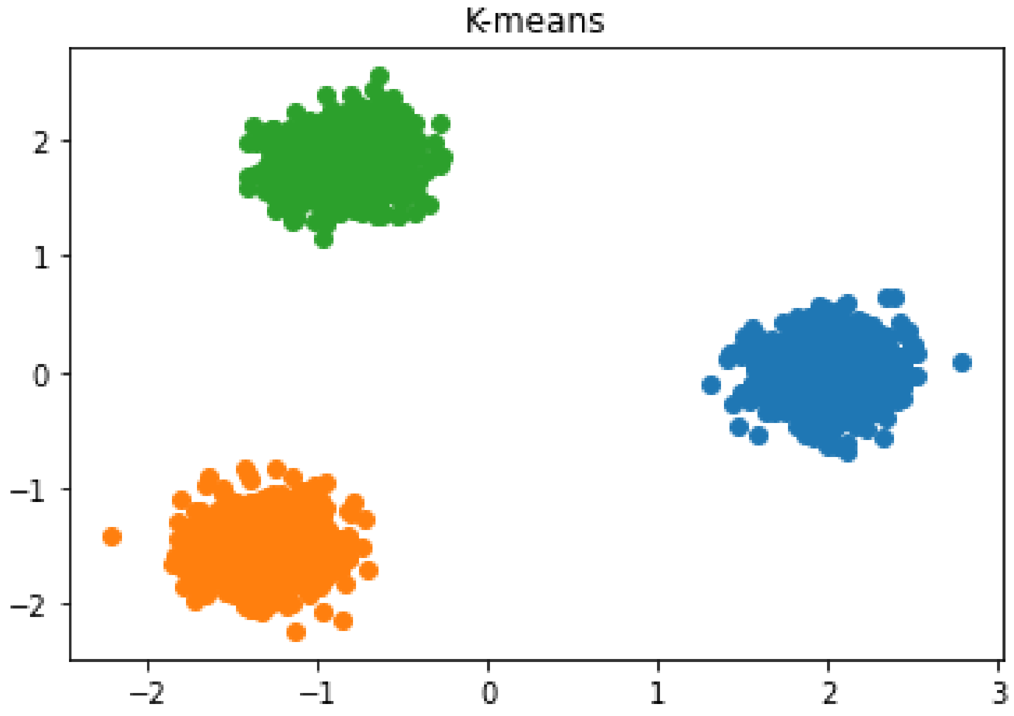


Bio-clustering
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−12
−10
−8
−6
−4
−2
0
2

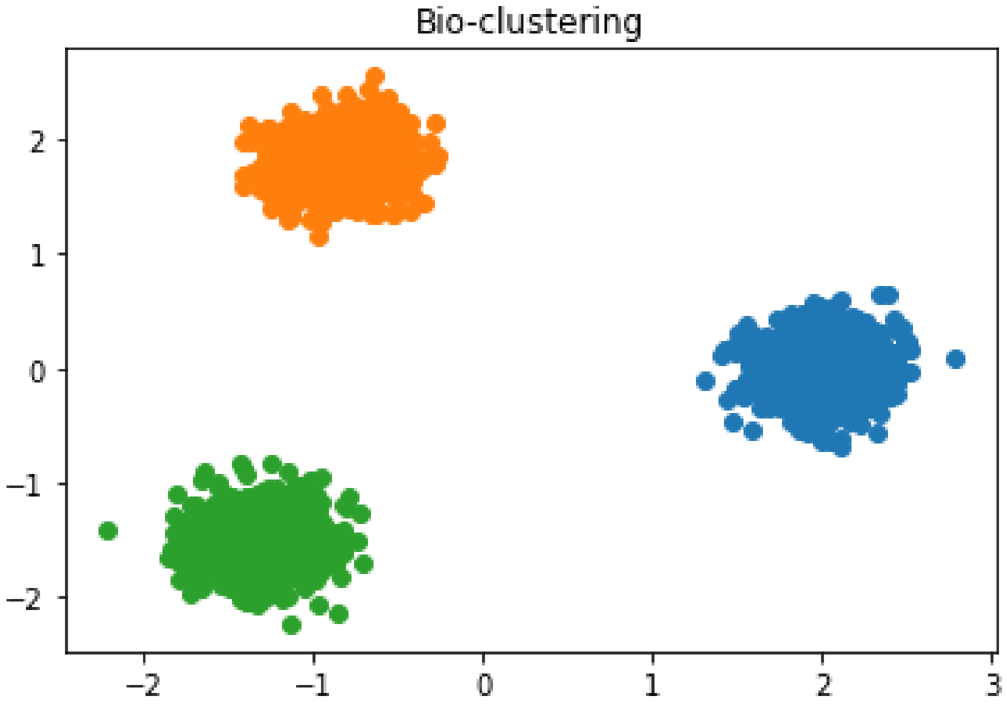
BLOBS



2
1
0
−1
−2
−2 −1 0 1 2 3

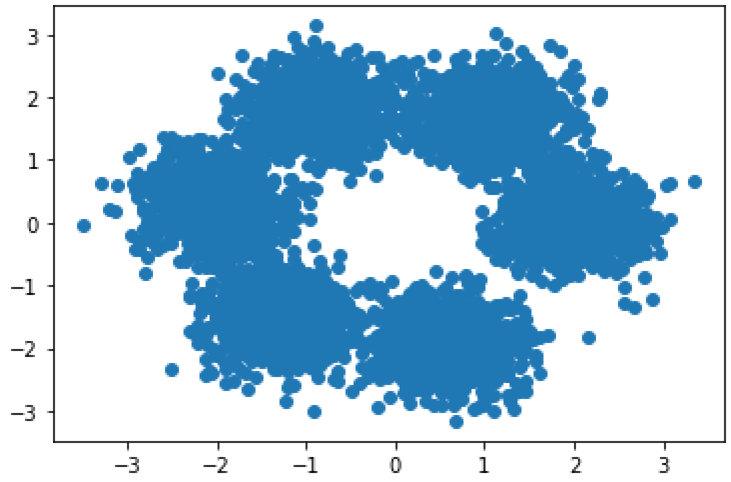


K-means
2
1
0
−1
−2
−2 −1 0 1 2 3

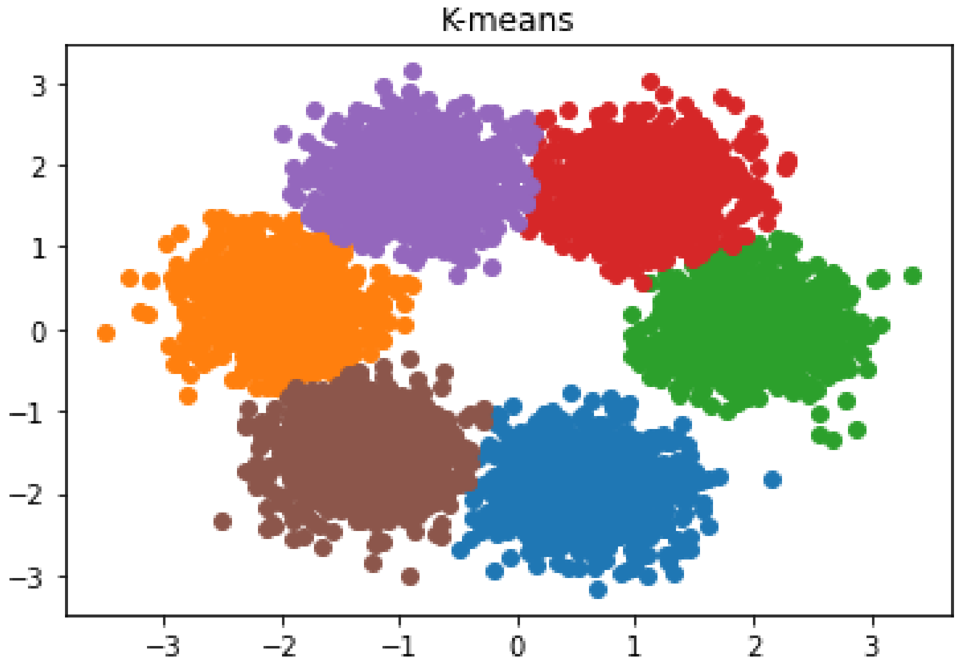


Bio-clustering
2
1
0
−1
−2
−2 −1 0 1 2 3

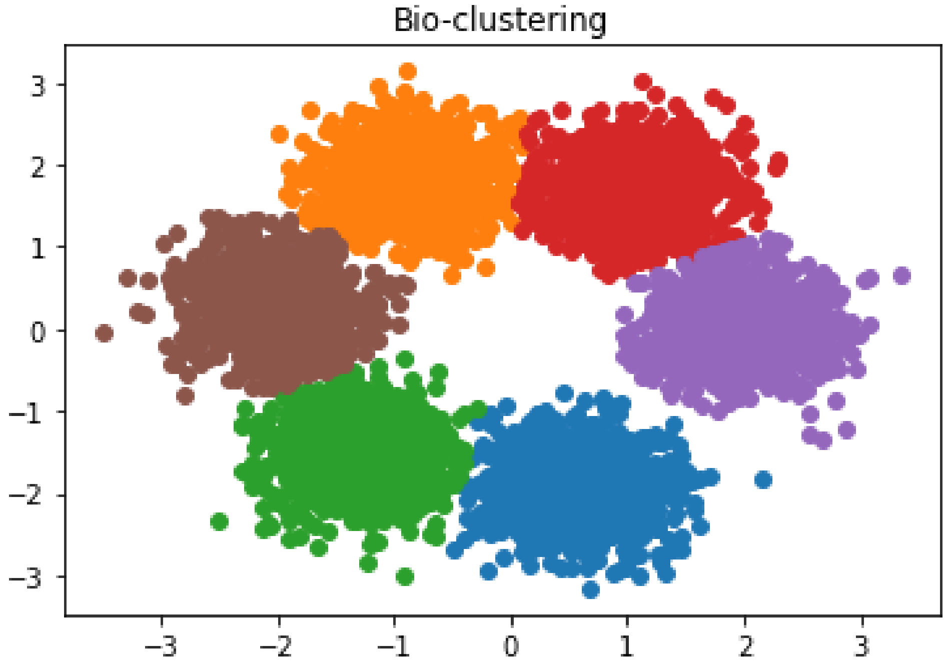
BLOBS



3
2
1
0
−1
−2
−3
−3 −2 −1 0 1 2 3



K-means
3
2
1
0
−1
−2
−3
−3 −2 −1 0 1 2 3



Bio-clustering
3
2
1
0
−1
−2
−3
−3 −2 −1 0 1 2 3

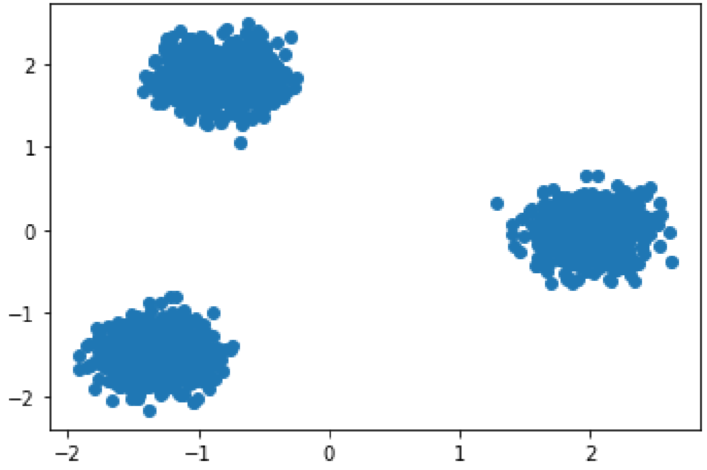
FIND OPTIMAL NUMBER OF CLUSTERS

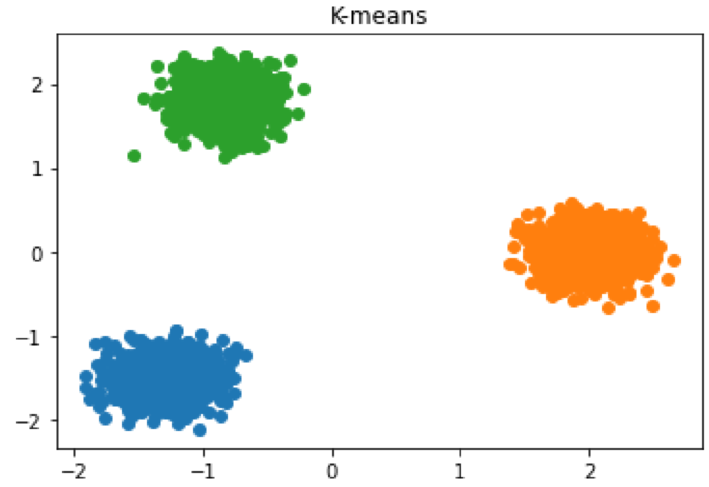
A metric to run elbow method over

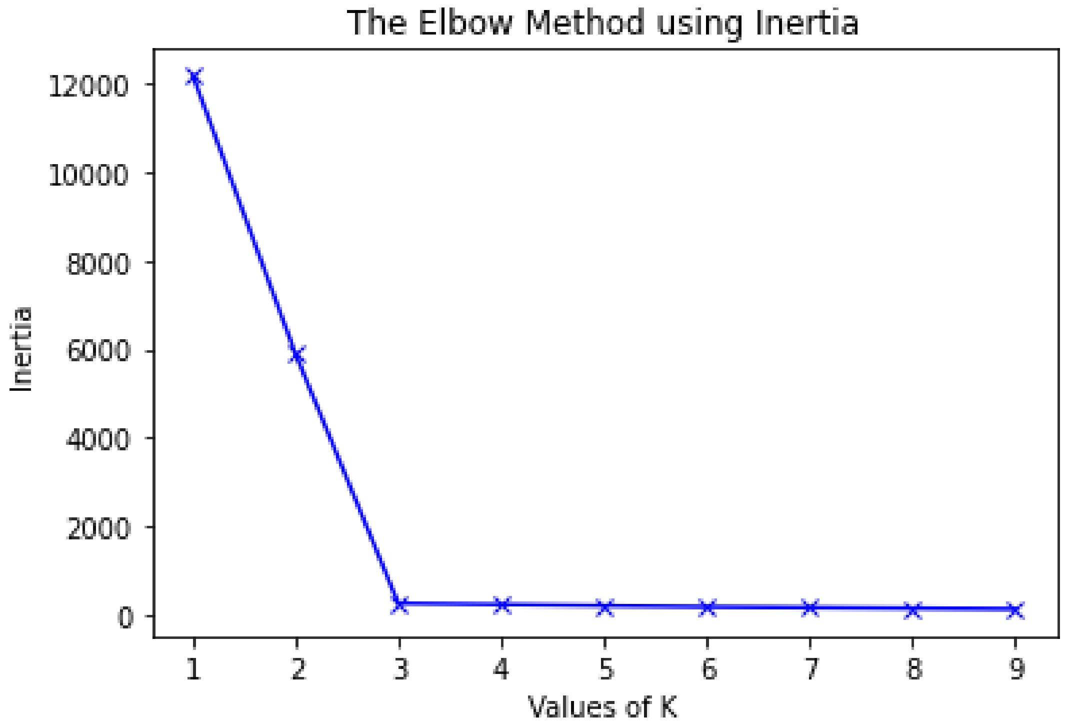
ELBOW METHOD IN K MEANS:

METRICS:

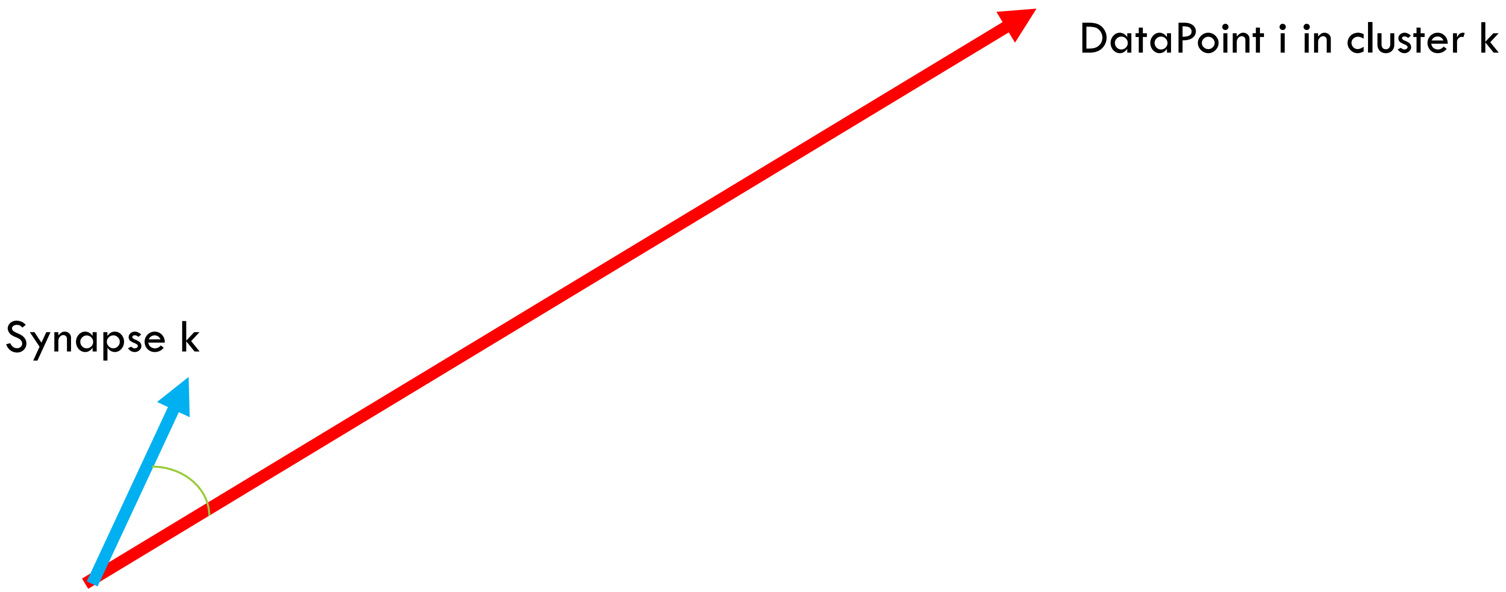
- Distortion
- Inertia



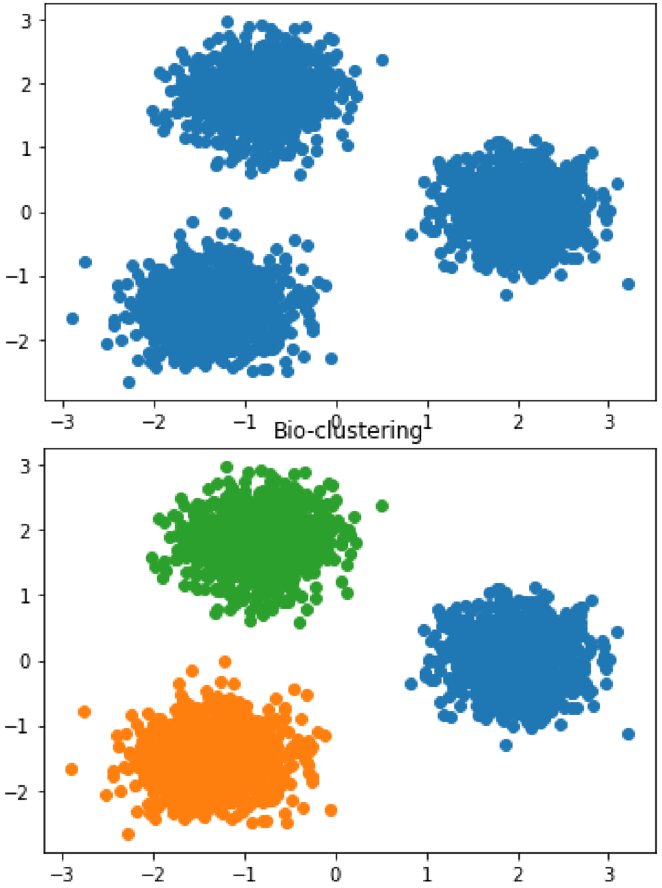


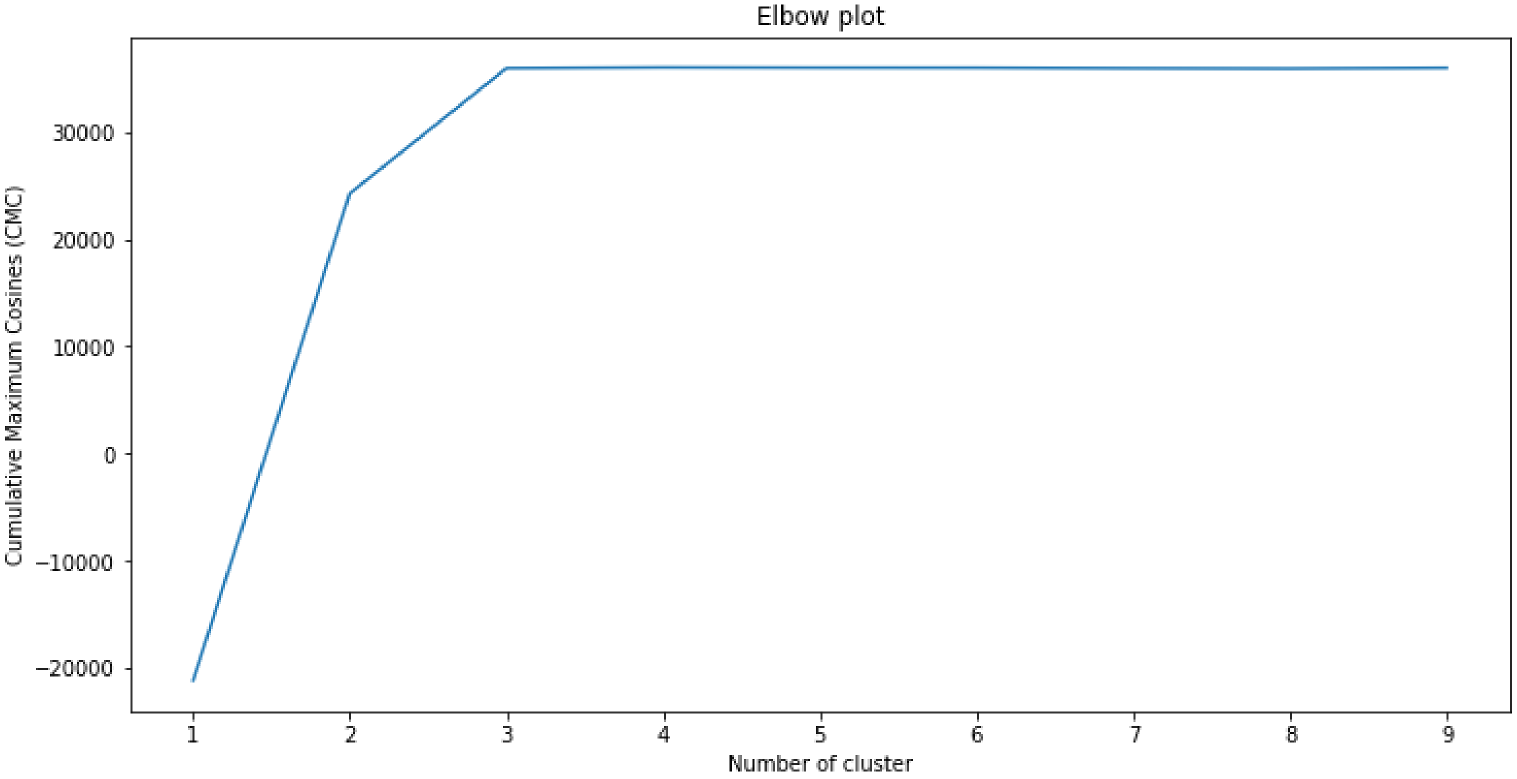


CULUMATIVE SUM OF COSINES

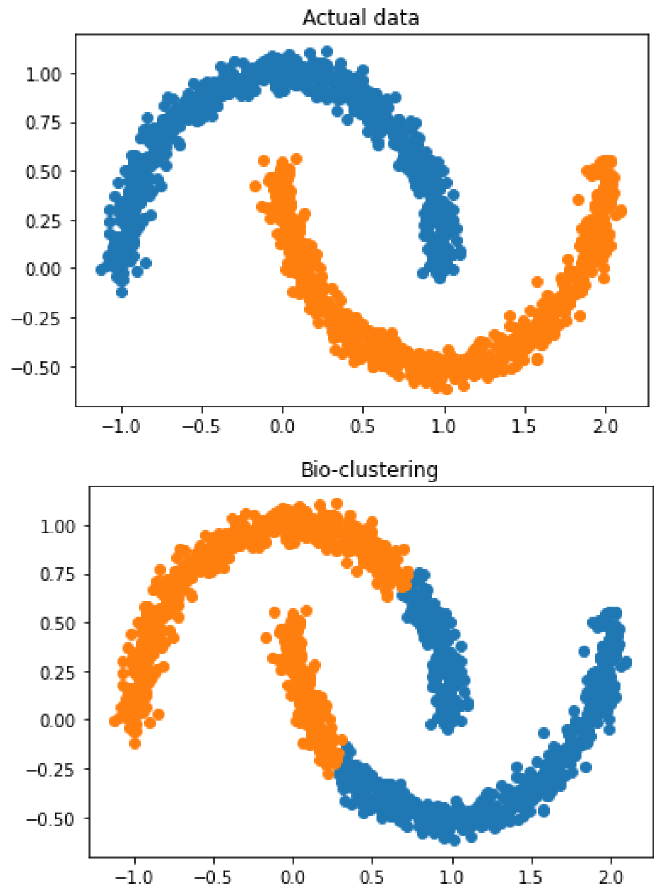


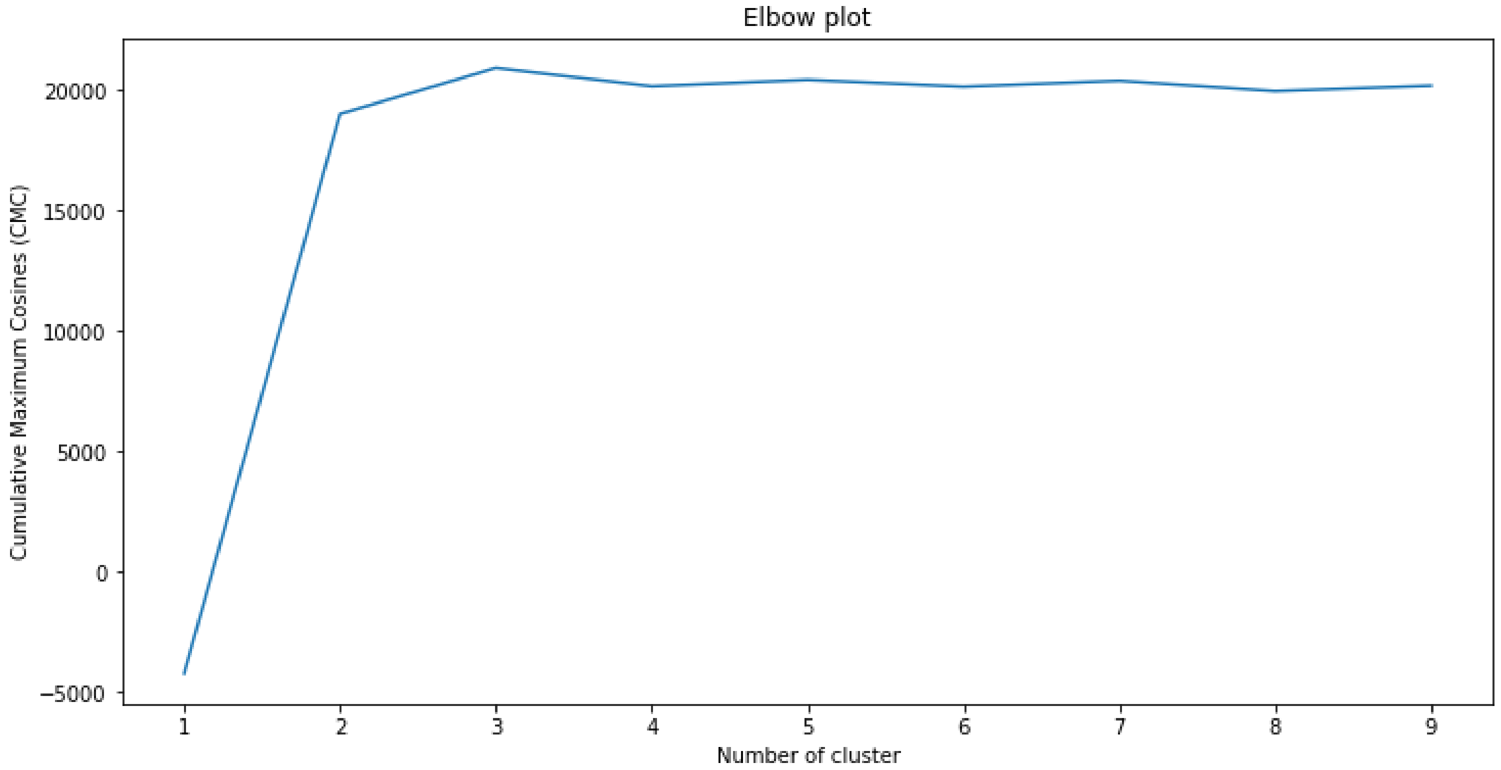
EXAMPLE


Bio-clustering


Elbow plot
Cumulative Maximum Cosines (CMC)
Number of cluster

ANOTHER EXAMPLE





APPLICATIONS OF CLUSTERING

- Pattern recognition
- Marketing and Sales
- Outlier detection applications
  - Identifying fraudulent or criminal activity
  - Classifying network traffic
- **Fake news detection**

APPLICATION: FAKE NEWS DETECTION

Comparing with kmeans results.

Reference:- kaggle.com/nasirkhalid24/unsupervised-k-means-clustering-fake-news-87

PROCEDURE

Step 1

Importing and cleaning the data

Step 2

Converting the articles into Sentence vectors

Step 3

Run Bio-clustering

Step 4

Mapping the predicted labels to clusters

GETTING SENTENCE VECTORS

- **Word Embedding**
  - Language modeling technique used for mapping words to vectors of real numbers
- Need for cleaning of data
- **Word2Vec**
  - Consists of models for generating word embedding.
  - Shallow two layer neural networks

MAPPING CLUSTER LABEL TO RIGHT CLASS

1
- Take a fraction of data points from each class

2
- Find the most labelled cluster for each class fraction

3
- Map this modal cluster label to this class

RESULTS

Comparing with kmeans results.

PCA VISUALISATION

'0' represents Fake news, '1' represents True news

CONCLUSION

Both K-Means and Bio-clustering show around 87% accuracy in classification

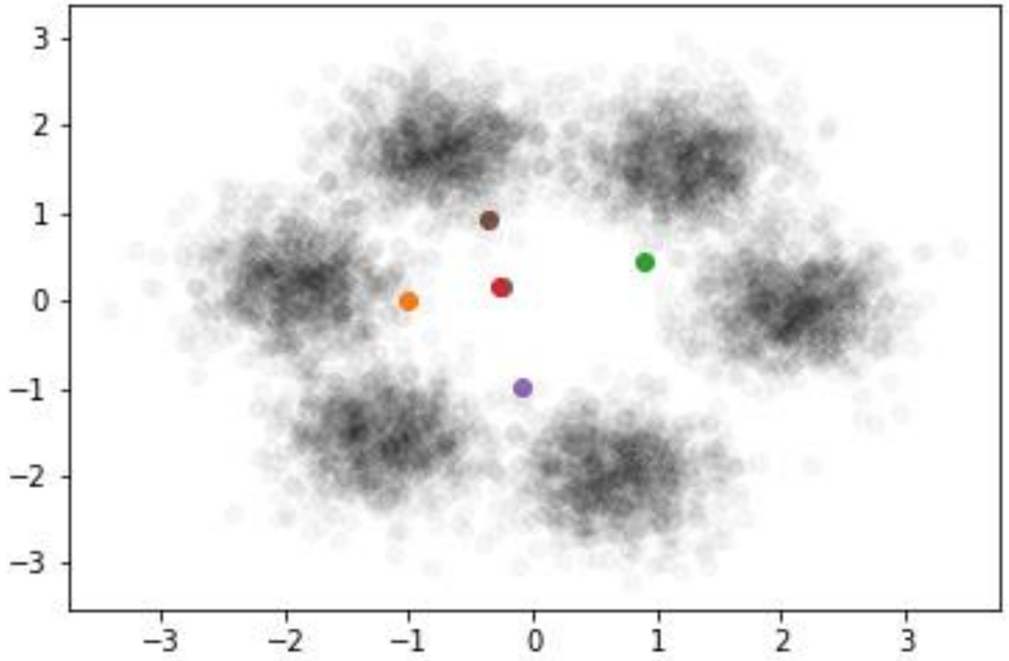
We can conclude that bio-clustering is a viable partitioning clustering algorithm

OUR INNOVATION

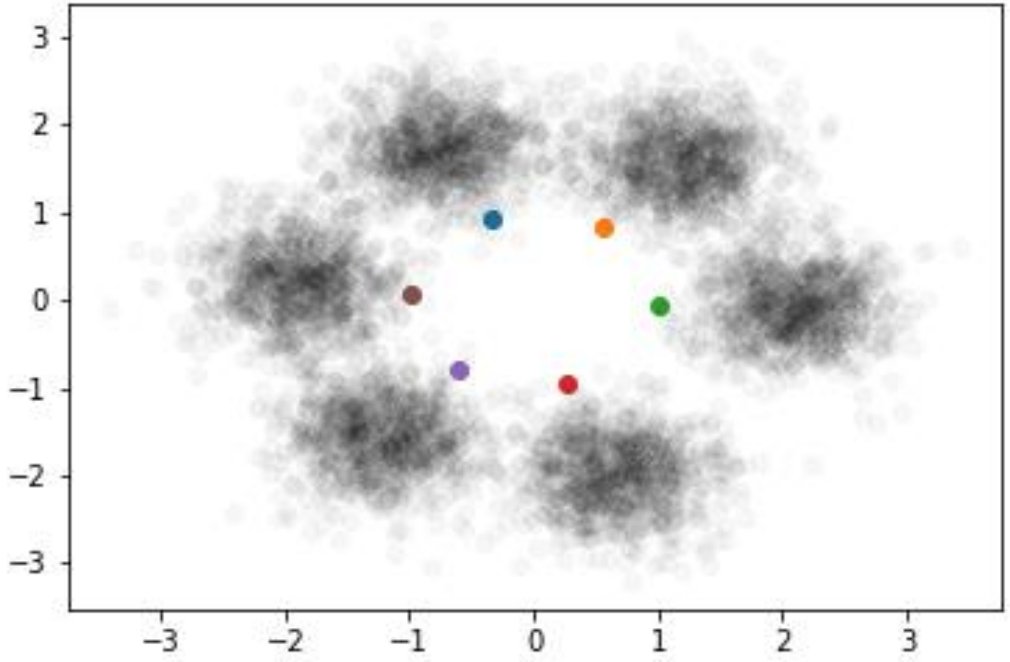
- Mean shifted Bio clustering
- Data based random synapse initialization
- Divisive normalization (L2)
- Using Cumulative Maximum Cosine in elbow method for finding optimal number of cluster

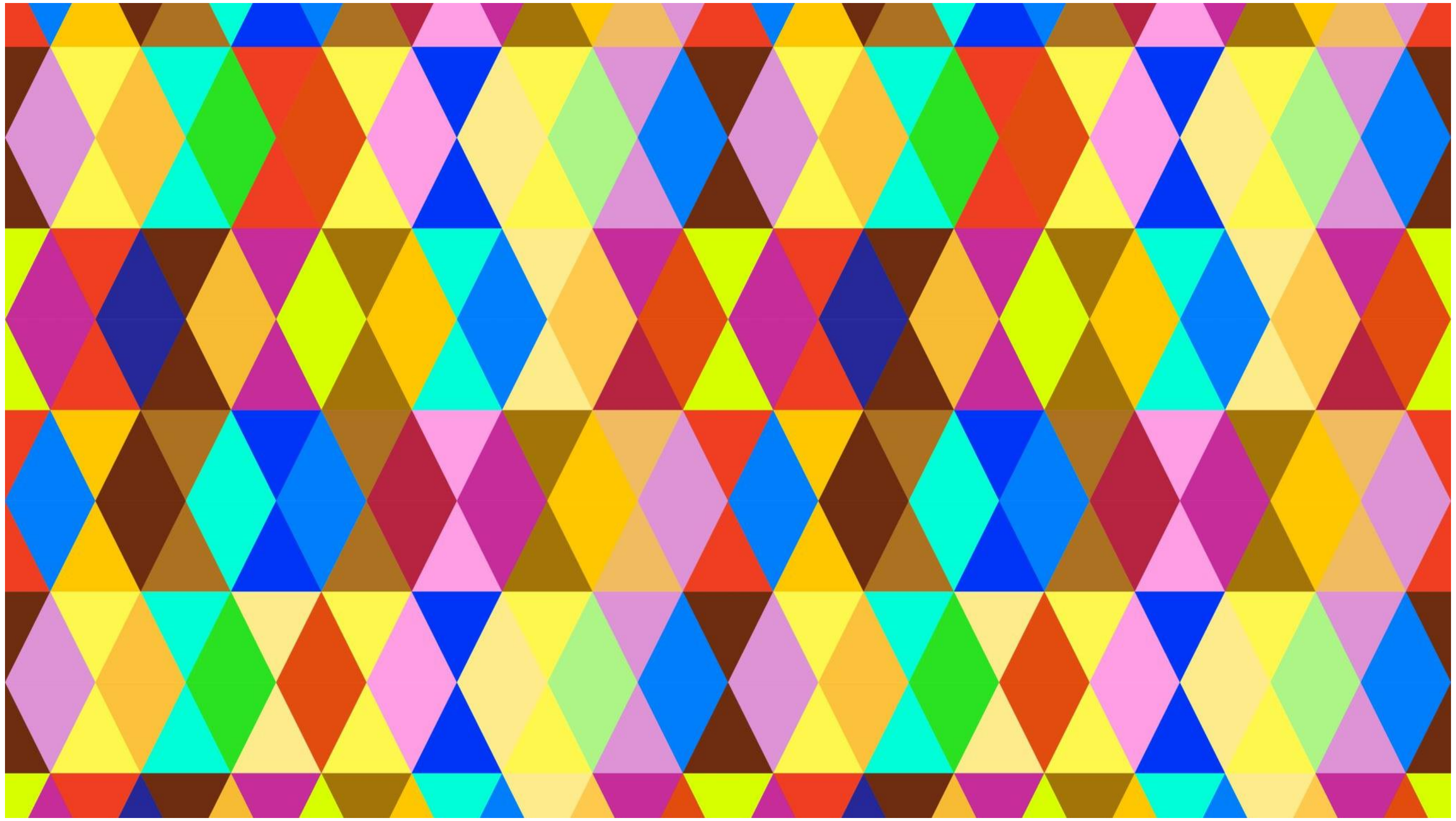
EFFECT OF OUR ADDITION

Algorithm as it is



With normalization and data driven synapse initialization





THANK YOU !!