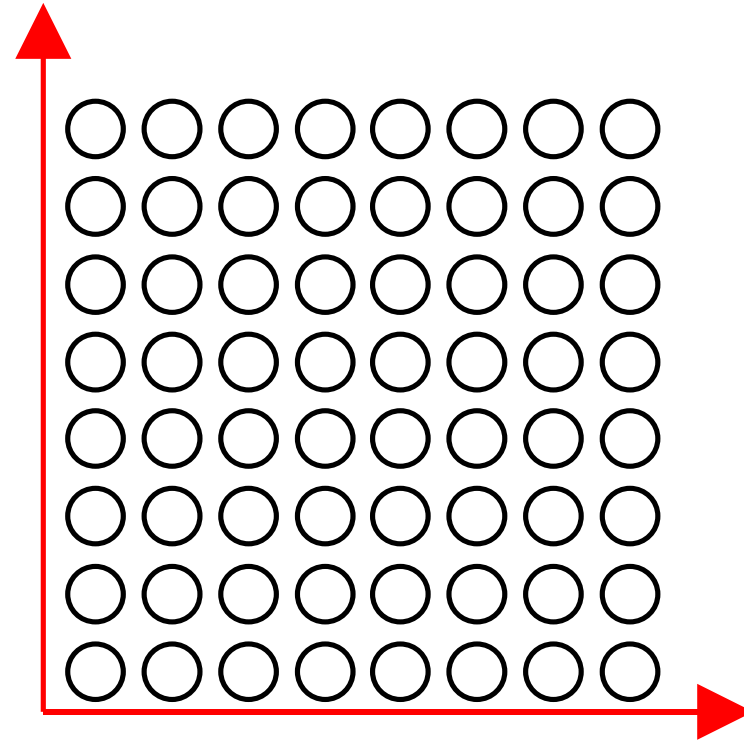


# Kohonen Maps (Self Organizing Maps)

- Retours sur les K-Means
- Idée :
  - Contraintes sur les K-Means
  - Ajouter des 'coordonnées' à chaque centroïde pour former une topologie
- Ainsi :
  - Chaque neurone  $i$  possède :
    - Coordinate Vector ( $C_i$ )
    - Feature Vector ( $W_i$ )

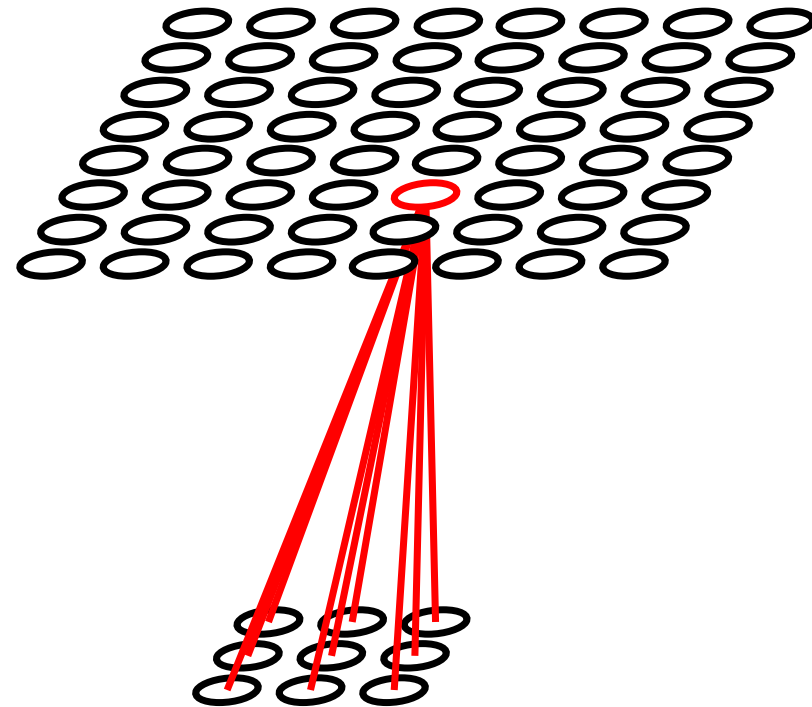
# Kohonen Maps (Self Organizing Maps)

- Ainsi :
  - Chaque neurone  $i$  possède :
    - Coordinate Vector / Topological Vector ( $C_i$ )
    - Feature Vector ( $W_i$ )



# Kohonen Maps (Self Organizing Maps)

- Ainsi :
  - Chaque neurone  $i$  possède :
    - Coordinate Vector / Topological Vector ( $C_i$ )
    - Feature Vector ( $W_i$ )

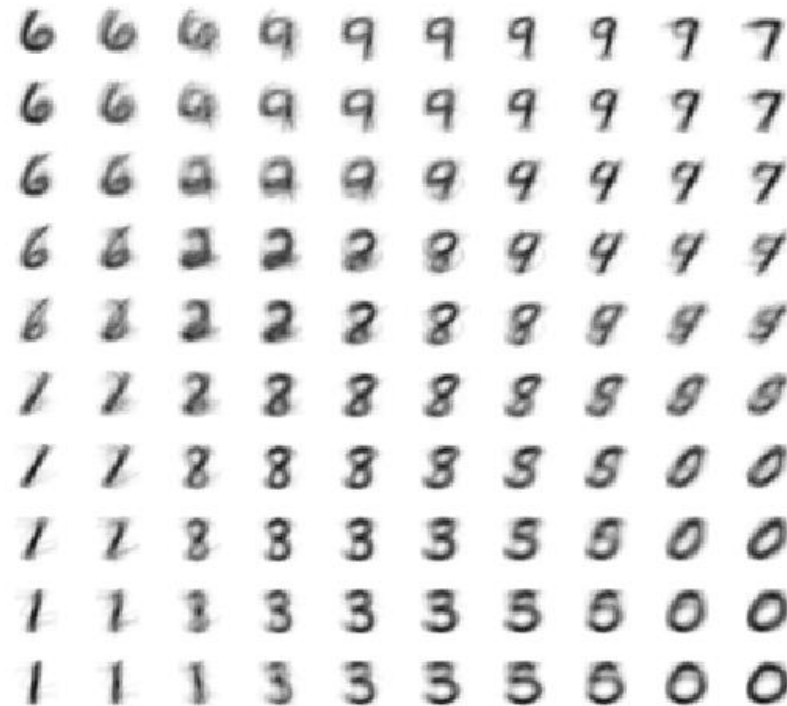


# Kohonen Maps (Self Organizing Maps)

- Algorithme :
  - Initialiser :
    - Choisir les  $W_i$  complètement au hasard (ou parmi les  $S_j$  existants au hasard)
  - Répéter :
    - Choisir un exemple au hasard :  $S_j$
    - Trouver le 'neurone'  $k$  dont le feature vector est le plus proche de  $S_j$
    - Modifier le feature vector du neurone élu  $W_k$  pour qu'il ressemble un peu plus à l'exemple choisi
    - Modifier les feature vectors des neurones voisins  $W_i$  pour qu'ils ressemblent à l'exemple choisi, mais de manière proportionnellement atténuée en fonction de la distance au neurone choisi dans l'espace topologique
- Formule générique classique de mise à jour pour tous les feature vectors :
  - $W_i = W_i + \alpha \times e^{\frac{-||C_i - C_k||^2}{2\gamma}} \times (S_j - W_i)$
- $\alpha$  est le pas d'apprentissage
- $\gamma$  permet de faire varier l'impact de la correction selon la distance

# Kohonen Maps (Self Organizing Maps)

- Exemples de résultats sur le MNIST Digits dataset :



# Kohonen Maps (Self Organizing Maps)

- Faites varier :
  - $\alpha$
  - $\gamma$
  - Le nombre de neurones
- Quels impacts ?