How Self-Organizing Maps (SOMs) Learn?

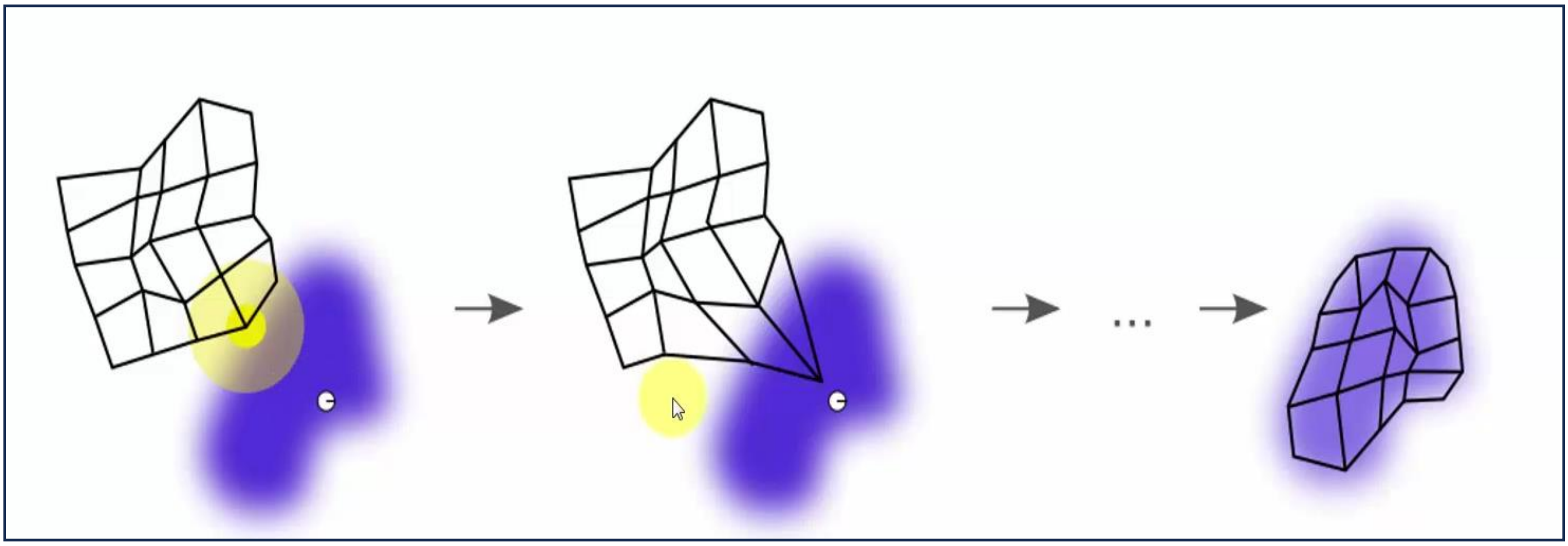


Image 1: Mesh Adaptation in Self-Organizing Maps (SOM) over many steps.

What Does This Mean?

- When you first start training a Self-Organizing Map, the grid of nodes (the mesh) has *random* weights, so it doesn't represent your data at all.
- Each time a data point is considered:
  1. The SOM finds the Best Matching Unit (BMU)—the node whose weights are closest to the data point.
  2. The BMU and its neighboring nodes are moved *slightly* closer to that data point.
  3. This process repeats for many data points and many cycles (epochs).
- At first: Only a small part of the mesh close to the BMU is noticeably pulled toward each input data point. The rest of the mesh remains mostly unchanged.
- Over time: As more data points are introduced and the process is repeated, *different parts of the mesh* are pulled toward different clusters of data. Gradually, the entire mesh (grid) stretches, bends, and moves to cover the whole data space—matching the shape and distribution of your dataset.
- Final outcome: The grid becomes a low-dimensional (usually 2D) map that preserves the relationships and clusters found in the higher-dimensional dataset. Similar data points are mapped to nearby nodes, so the mesh as a whole now "fits" and represents all the data clusters appropriately.

Visualization:

Imagine the mesh as a net:
At first, only a few knots in the net move when you pull on them (single data point), but after many pulls (iterations), the entire net re-shapes to wrap around the pattern of your data.

Why do we do it over many steps?

- If you updated the mesh to fit just one data point instantly, it would lose overall structure and not learn general patterns.
- Slow, repeated, and neighborhood-based updates allow the mesh to self-organize smoothly, learning from all parts of your data.

Key Takeaway

Image 2 visually shows how, with repeated training, a SOM starts from a rough state and gradually adapts its grid, so by the end every part of the mesh fits the underlying data distribution and preserves clustering relationships

If you updated the mesh to fit just one data point instantly, it would lose overall structure and not learn general patterns:

If we made the SOM instantly change its whole structure to fit one single data point perfectly — for example, by moving the BMU *exactly* to that input point —
then:

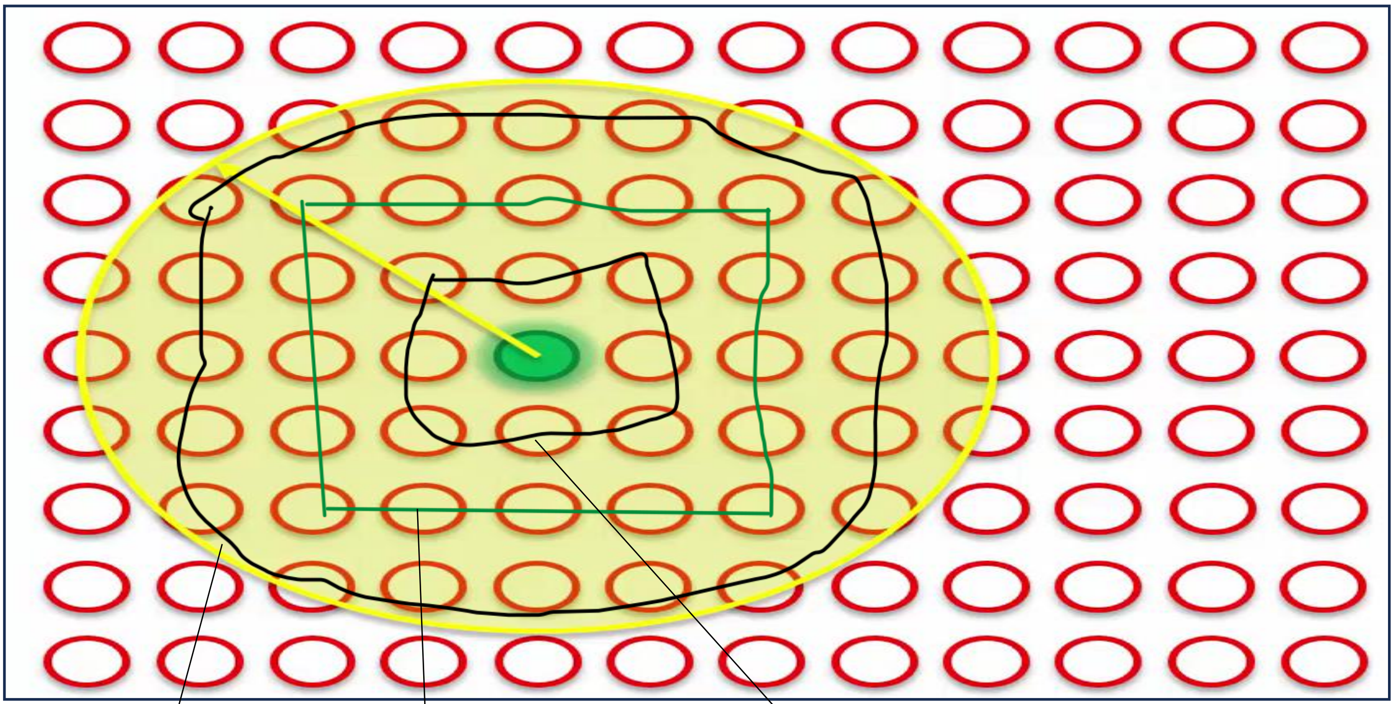
- The SOM would overfit to that one point.
- The mesh would distort, losing its smooth, organized 2D structure.
- The map would stop representing the general patterns of the entire dataset — it would just chase individual samples.

In other words:

The SOM would stop learning the *overall shape* of the data distribution and instead just memorize individual points.

Image 2&3: describes how a Self-Organizing Map (SOM) learns and adapts to input data by updating the weights of its nodes.

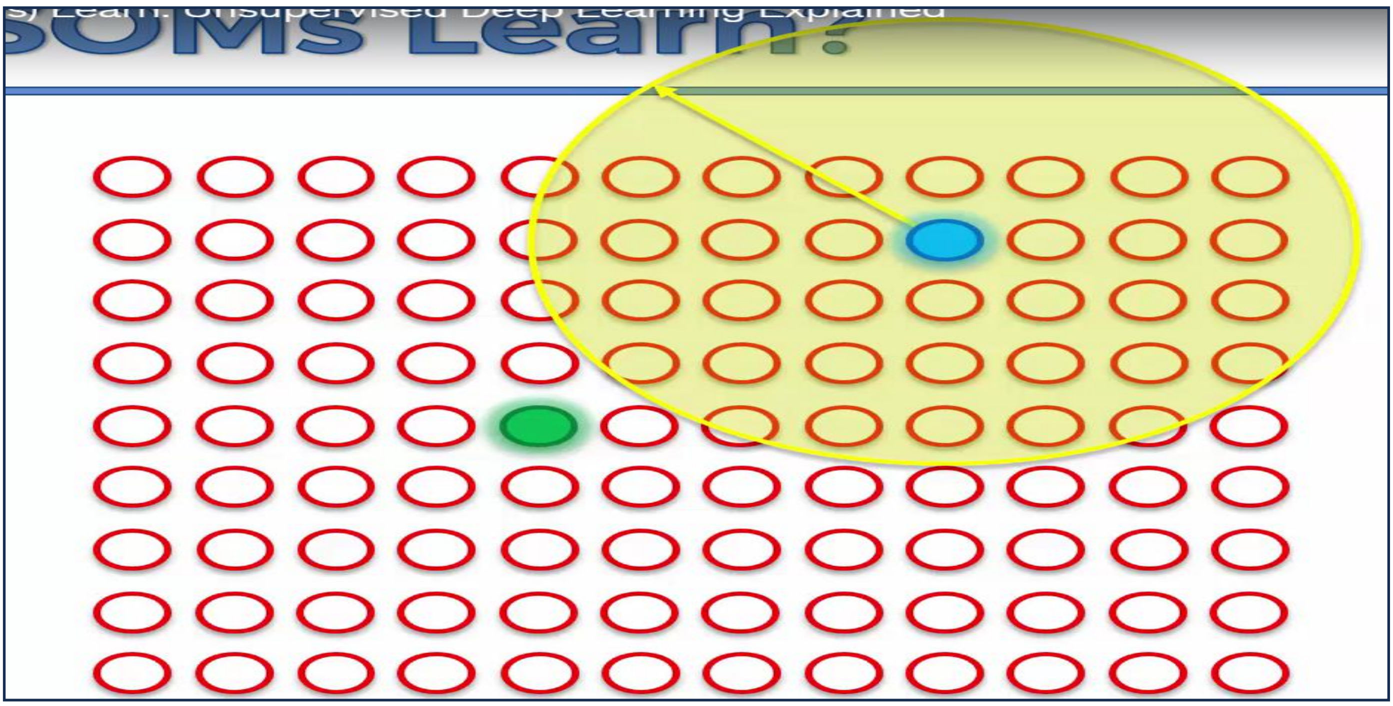
- In Self-Organizing Maps (SOMs), each red circle represents a node, and each node has its own set of weights (representing a coordinate or feature vector) that can be "moved" or updated.



these weights are going to be updated less

the closer you are to the BMU,the heavier are your weights updated.So these weights are going to be updated the most.

these weights are going to be updated even less.



When one node (the BMU) moves toward a data point, its nearby nodes also move slightly in the same direction — keeping the map smooth and preserving the topological

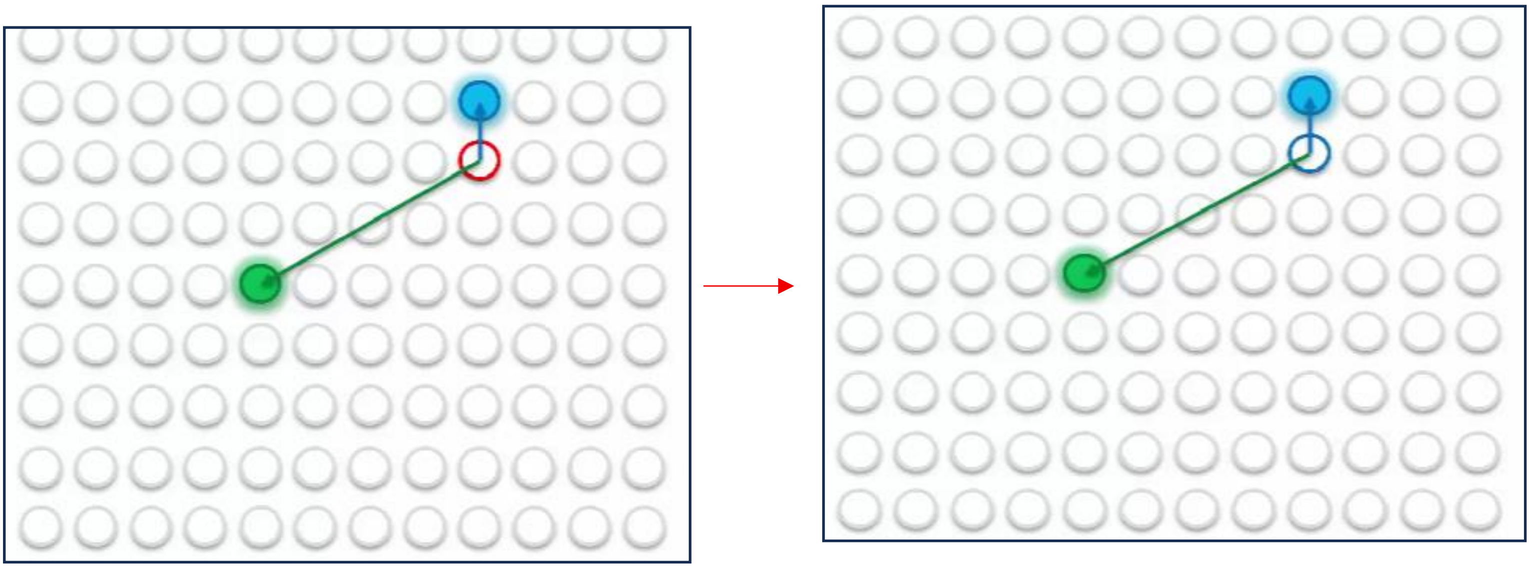
- When a data point (for example, represented by a green dot) is given to the SOM, the closest node (called the Best Matching Unit, or BMU) is identified. This node is marked and selected, as shown in the images.
- The weights of the BMU are updated to move closer to the input data point. In other words, the BMU's weights become more like the current data row. This makes the map learn and represent the data better.
- Not only the BMU but also its neighbors (within a certain radius, shown by the yellow area) are updated—the closer a node is to the BMU, the more strongly its weights are updated. This is akin to stretching or pulling a net toward the data point, so neighboring nodes are gradually organized.
- When another data point comes in, a new BMU is found (it could be in a different region of the SOM), and the process repeats: the BMU and its local neighborhood are updated, making the SOM better represent the structure of the input dataset.
- Over time, the mesh/grid of nodes adapts to the distribution of data points, creating a map that organizes itself so similar data points are located near each other on the grid, preserving the topological relationships in the data.

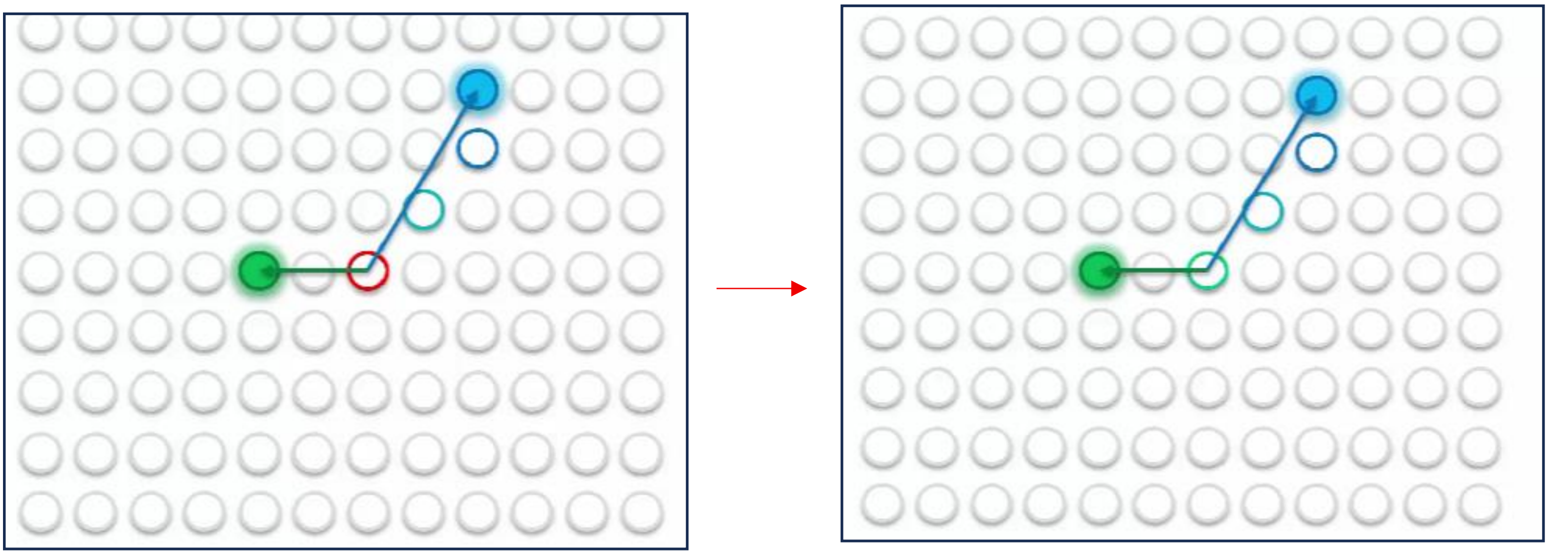
Why a Larger Map?

- The concept of using a larger map helps to better visualize and understand how the update radius affects multiple nodes and how the global structure adapts during training. A larger map visually demonstrates how far the updates spread and how neighborhoods overlap and interact as new data points are introduced.

Visual Summary for the Images

- The yellow circle shows the "radius" of influence around the BMU (blue or green filled node), representing which nodes get their weights updated.



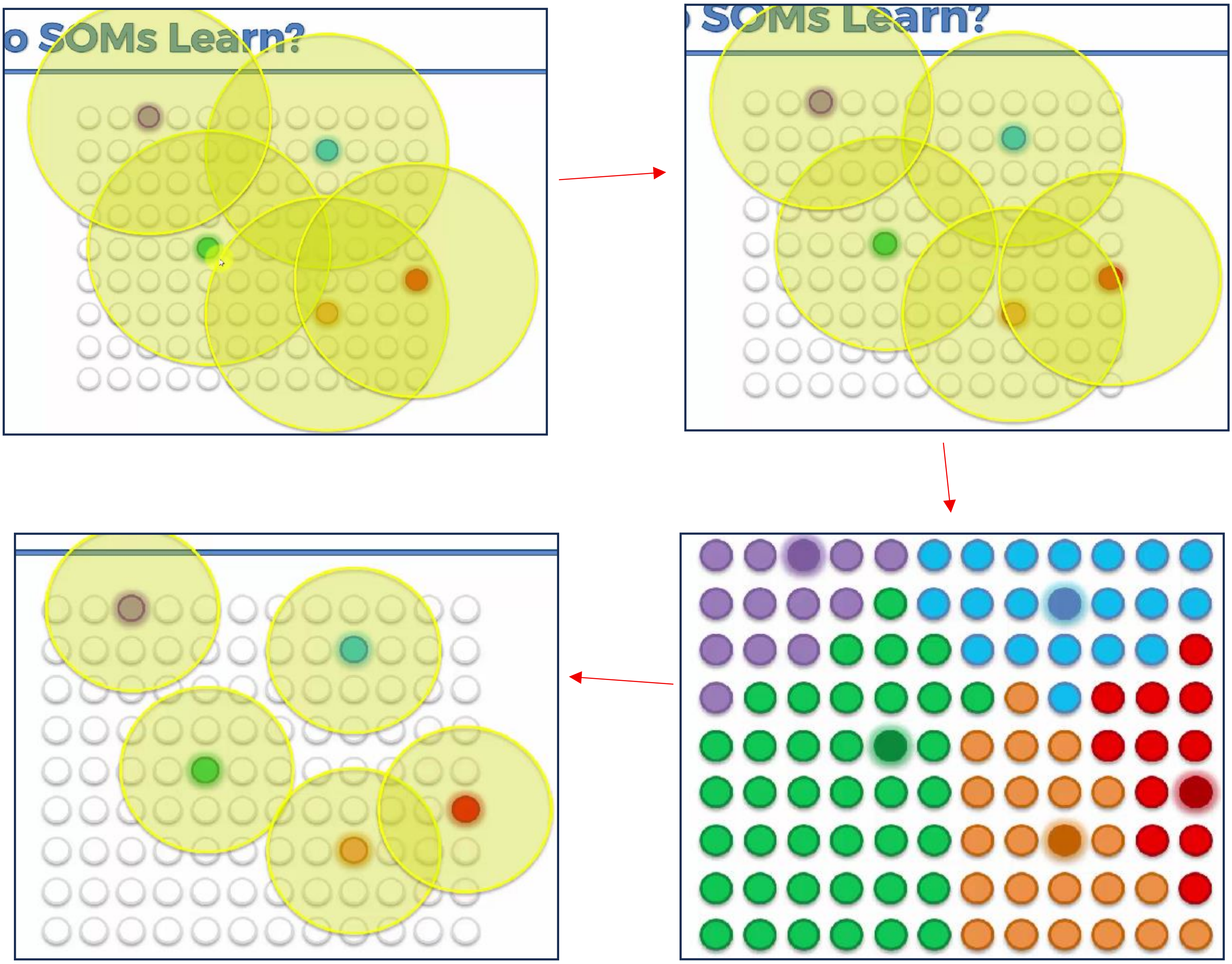


- Each data point is matched to its closest node—called the "Best Matching Unit" (BMU). In your image, the green circle is one BMU for a particular data point, and the blue circle is another BMU from a different data point.
- The red node is a specific node we are analyzing.
- The red node is closer to the blue BMU than it is to the green BMU. If the node falls outside the "influence radius" of the green BMU, it isn't pulled by that green BMU's update at all.
- During training, when the blue BMU's weights are updated, all nodes within its neighborhood are also updated. The closer a node is to the BMU, the more its weights are adjusted.
- Because the red node is so far from the green BMU, but very close to the blue BMU, it gets updated much more strongly by the blue BMU.

Over time, the red node's weights become similar to the blue BMU's weights, so its color changes to blue to indicate that it is now part of the blue BMU's group.

Summary

- SOM learning means each node is influenced most by the BMUs it's nearest to.
- Nodes closer to a BMU are more likely to resemble it after repeated updates, and their weights/colors reflect this group's characteristics.
- This is how clusters and regions gradually form on the map, letting SOMs organize similar groups together visually and quantitatively.



- At the start, each Best Matching Unit (BMU) in the SOM finds a data point that it matches with.
  Then, that BMU moves closer to the data point to become more similar to it.
- But it’s not just the BMU that moves — the neighboring nodes around it also move a bit.
  These neighbors are inside a certain radius (shown as yellow circles).
  Nodes close to the BMU move a lot, while nodes farther away move a little.
- In the beginning of training, this radius is large, so almost the whole map changes a little each time.
  This helps the SOM quickly get the rough shape of the data.
- As training continues, the radius becomes smaller.
  Now, only nearby nodes around the BMU move.
  This allows the SOM to fine-tune details and form clear borders between different data groups.
- During this process, there is a push and pull between nodes — neighboring BMUs compete for nearby data points.
  Gradually, they settle into distinct regions or clusters.
- After many rounds, the SOM forms clear clusters on the map.
  Each region represents a group of similar data points,
  and the colors show which group each area belongs to (like the final colorful grid).

Summary:

- The SOM first makes big, general changes to match the overall data pattern,then slowly makes smaller, precise adjustments so that each node represents its closest group of data.
- In the end, the SOM becomes an organized map where similar data points are grouped together,just like the colored blocks in the final image.

Stage	Radius size	Effect on nodes	What happens	Purpose
Stage 1: Rough Training	🔵 Very large	Almost all nodes move slightly	The SOM makes big, global adjustments to roughly match data patterns	To get a basic structure of the data
Stage 2: Middle Phase	🟡 Medium	Only nearby nodes around each BMU move	The SOM starts to fine-tune clusters and separate groups	To create clearer boundaries between groups
Stage 3: Fine Tuning	🔴 Small	Only the BMU and its closest neighbors move	The SOM polishes the map, aligning each node with similar data	To reach a stable, detailed map where similar data points are grouped

Note (About Outside Radius):

Nodes outside the current radius are not updated — they stay where they are.
Only the BMU and its nearby neighbors inside the radius move.

Important points:

1. SOMs retain the topology of the input set

Meaning:
SOMs keep the **spatial relationships** (structure or pattern) of your data intact.
If two data points are **close to each other** in the original dataset,
they will also appear **close to each other** on the SOM map.

Example:
If countries with similar living standards (like Sweden and Norway) are close in data values,
they'll also appear **close together** on the SOM grid.

✅ **In short:** SOMs preserve the **shape and structure** of your data when reducing dimensions.

🔍 2. SOMs reveal correlations that are not easily identified

Meaning:
SOMs can **find hidden patterns or relationships** in complex data that humans can't easily see.
Even if your dataset has **many columns (features)**,
the SOM can **group and show** similar data points together visually.

Example:
If 50 features describe countries (health, education, GDP, etc.),
SOM can show which countries are **similar overall**, even if you couldn't tell just by numbers.

✅ **In short:** SOMs help you **discover hidden similarities** and **relationships** in data.

🤖 3. SOMs classify data without supervision

Meaning:
SOMs are **unsupervised algorithms**, so they **don't need labeled data** (no "cat" or "dog" tags).
They **learn by themselves** from raw data, finding natural groupings or clusters.

Example:
If you give SOM 1,000 customer profiles, it will **group similar customers** together
(e.g., high spenders, frequent buyers) — without you telling it who's who.

✅ **In short:** SOMs **self-learn and group** data on their own — no human labeling needed.

🎯 4. No target vector → No backpropagation

Meaning:
Since SOMs don't have **target outputs** (like "this is a cat"),
there's **no error to calculate** and **no need for backpropagation**.
Weights are updated based on **distance to the input data**, not on an error signal.

Example:
In normal neural networks, you compare predicted vs. actual output → calculate error → adjust weights.
In SOMs, you just find which node (BMU) is closest and **move it closer** — no error term used.

✅ **In short:** SOMs learn by **proximity**, not by **error correction**.

🔗 5. No lateral connections between output nodes

Meaning:
Output nodes (neurons on the SOM grid) are **not directly connected** like in regular neural networks.
There's **no activation function or formula** passing signals between them.
Their interaction happens **only through the neighborhood radius** —
when one node (BMU) moves, its nearby nodes move slightly too, but not through any link.

Example:
If you see a SOM grid with lines connecting nodes — those lines are **just for visual layout**,
not real data connections.

✅ **In short:** SOM nodes **don’t pass signals** to each other; they **only influence neighbors indirectly** during updates.