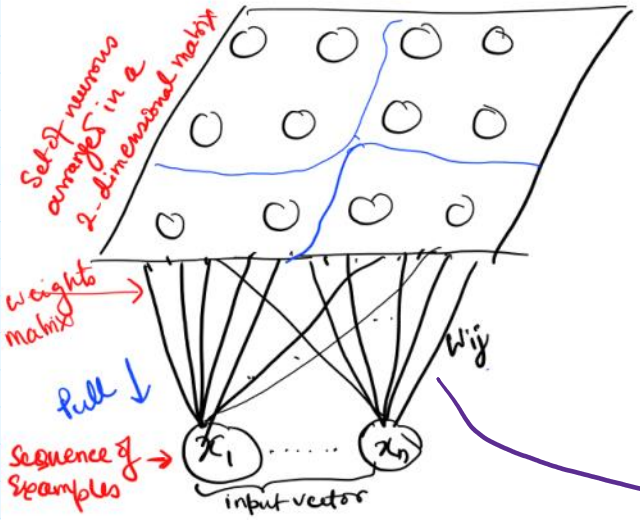
SOFM or SOM or Kohonen Map.

15 November 2024 13:21

Self Organizing Feature Map.



- Code vector, initially generated by random no. generator.
  - ↓
  - Subjected to the influence of sequence of examples (≈ attribute vectors)
    - ↓
    - Each sequence pulls vector in diff direction
      - ↓
      - Over time, vector settles in a location that represents a compromise among all conflicting forces.

Aim: By training, get a situation where neighboring neurons respond similarly to similar I/p vectors.

→ SOM is a multidimensional scaling technique that constructs an approximation of PDF of some dataset and PRESERVES TOPOLOGICAL struct.

Stochastic Learning Algo:

```
for each w_i ∈ W
        initialize w_i

while stopping condition not true:
        select random I/P vector x_i ∈ X
        find best matching neuron for x_i
        for each w_i ∈ W:
                update w_i
        end for
end.
```

Now, make this batch learning (each batch defined by neighborhood)

```
for each w_i ∈ W:
        initialize w_i;

while stopping condition not true:
        for each w_i ∈ W:
                X_i ← ϕ
                for each x_i ∈ X:
                        if x_i is in neighborhood of w_i:
                                add x_i to X_i
                        end if
                end for
        end for
        for each w_i ∈ W:
                w_i ← update mean over X_i
        end for
end.
```

Now, the winning neuron i is the neuron such that $\text{argmin}_i |x - w_i|$

has higher value for neurons closer to i on lattice ↑ neighborhood

Now, the winning neuron i is the neuron such that $\text{argmin}_i |x - w_i|$

Adapt the weights of n other neurons as:

$w_n = w_n + \eta(t) h(i) (x - w_n)$

- $h(i)$: has higher value for neurons close to i on lattice; neighborhood function
- $\eta(t)$: learning rate
  - when $t \uparrow\uparrow$, $\eta \downarrow\downarrow$, $h(i)$ also $\downarrow\downarrow$ its spread

After each training step, weight of neurons close to winning neuron are adjusted → to have stronger response to current pattern.

After training, SOM forms topological map of output.

Okay, but where's the dimensionality reduction?

Dimensions	no. of data points	Sample space
1	10	line w/ 10 pts.
2	10	10 x 10 = 100 datum
3	10	10 x 10 x 10 = 1000 datum

So, with SOM:

50, 10D vectors → [SOM] → 50, 2D vectors — dimension reduction is measured in the topological space of 2D grid.

SOM v K-Means:

- K-Means → breaks data points into some predetermined k-clusters → doesn't keep topological structure of data
- SOM → gives correct number of clusters by itself.
  - SOM is a constrained version of k-means (w/ k-means, centroids move anywhere)
  - no need to specify # of clusters.
  - SOMs keep topological structure of data

The three phases of SOM's algo:

① COMPETITION

- measure distance b/w vector of data pt and neuron in lattice
- find neuron closest to input vector = BEST MATCHING UNIT

Why competition?
Neurons compete in similarity to sample.

② COOPERATION

- find winning neuron's neighbors
- neighborhood is defined by a function h:
  - value ↓ with distance from neuron.
  - should be symmetrical about winning neuron

③ ADAPTATION

- update neurons as per h_i
- Math:

③ ADAPTATION

- update neurons as per h_i
- Math:

$w_{j(n)}$ = vector representation of neuron j @ iteration n.
$w_{i(n)}$ = winning neuron
α = learning rate
d = distance function

$$w_{j(n+1)} = w_{j(n)} + \alpha h_{ij}(d(x, w_{j(n)}))$$

h_{ij} (neighborhood fn) is a function of distance b/w i & j

$d(x, w_{j(n)})$ → distance of neuron j

Now, we can use a gaussian function to focus the influence on the neurons close to it:

$$h(d(i,j)) = e^{\frac{-d^2(i,j)}{2\sigma^2}}$$

final words: SOM

- NOT generative
- UNSUPERVISED (doesn't require labels)
- Very slow
- Best lattice dimension: $5\sqrt{N}$ → no. of datapoints.