Clustering, backprop

Tuesday, 23 April 2024 11:33 AM

→ <u>Learning</u>

① Supervised — Labels

② Unsupervised — Unlabelled .

→ <u>Clustering</u> :

<u>k-means clustering</u>

x_i — datapoints

TSS — Total sum of squares $= \sum_{i=1}^{N} (x_i - \overline{x})$

BSS — Between clusters sum of squares

WSS — Within clusters sum of squares

$x_i \rightarrow$ Cluster j — Average distance of x_i from the rest of centroids

centroid

$\boxed{TSS} = \underline{BSS} + \underline{WSS}$

constant   BSS ↑   WSS ↓

goal : clusters to be far away
make sure as close as possible
to assigned centroid

goal : <u>minimize WSS</u>

Algorithm :

k - hyperparameter.

① Initialise k centroids randomly $\rightarrow \{m_j\}_{i=1}^{k}$ Kmeans ++

② repeat $c(x)$ — cluster function.

→ for every Office Online Frame assign a cluster

$$c(x_i) = \underset{1 \le j \le k}{\operatorname{argmin}} \; \|x_i - m_j\|_2^2$$

→ update the clusters

for all $j \in [1, k]$

Let $S_j = \{ x_i : c(x_i) = j \}$.

$$m_j = \frac{1}{|S_j|} \sum_{x_i \in S_j} x_i$$

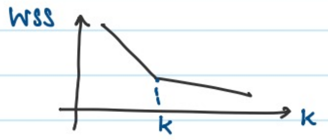
stopping condition:

① # iterations — (epochs)

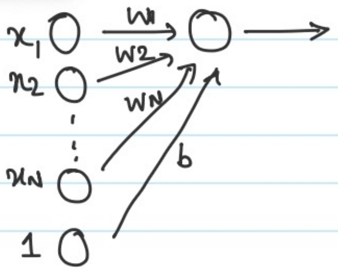
② $\frac{1}{k} \sum_{j=1}^{k} \|m_j^{(t+1)} - m_j^{(t)}\|_2^2 \le \varepsilon$.

How to choose k

Elbow method



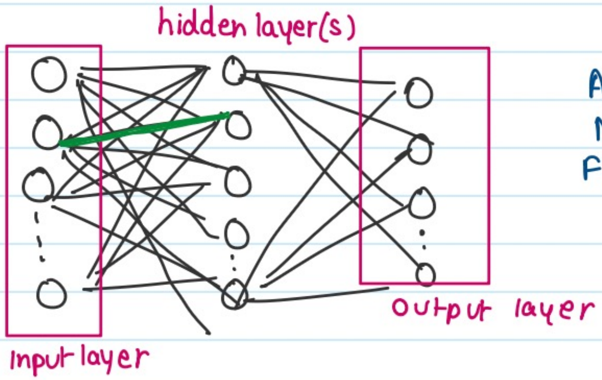
→ Neural networks



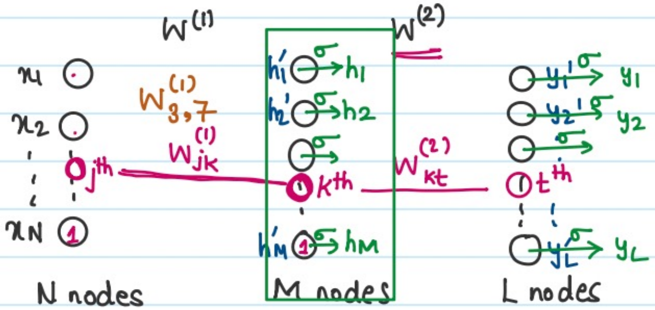
$$\longrightarrow w^T x + b \longrightarrow \theta \longrightarrow \underbrace{\theta(w^T x + b)}_{\hat{y}}$$

$$\theta(x) = \begin{cases} 1 & x \geqslant t \\ 0 & o/w \end{cases}$$

$$w_j^{(t+1)} = w_j^{(t)} + \eta \cdot [\hat{y}_i - y_i]\, x_{ij}$$



ANN – artificial NN
MLP – multi layer perceptron
FCNN – fully connected NN



datapoints – Q

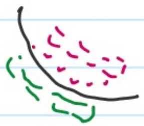
$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1N} \\ x_{21} & x_{22} & \dots & x_{2N} \\ \vdots \end{bmatrix} \quad Q \times N$$

x_{ij} – j^{th} feature of i^{th} datapoint

H – $Q \times M$

$$h_{ij} = \sum_{k=1}^{N} x_{ik} W_{kj}$$

$$X W^{(1)} = H$$
$N \times M$



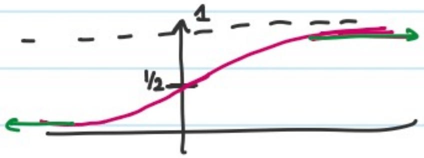
$$h_1' = \sum_{j=1}^{N} W^{(1)}_{j1} x_j \qquad y_t' = \sum_{j=1}^{M} W^{(2)}_{jt} h_j'$$

$$h_k' = \sum_{j=1}^{N} W^{(1)}_{jk} x_j$$

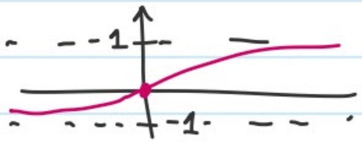
$$Y = X W^{(1)} W^{(2)} W^{(3)}$$
$$\Downarrow$$
output is a linear function of input

Activation functions — Non-linearities

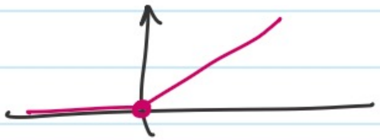
① sigmoid : $f(x) = \frac{1}{1+e^{-x}}$



② tanh : $f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$



③ ReLU : $f(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{o/w} \end{cases}$



$f'(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{o/w} \end{cases}$

④ LReLU : $f(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0.01x & \text{if } x < 0 \end{cases}$

→ Loss function / Error function $L(y, \hat{y})$ — $L(y, W^{(1)}, W^{(2)})$

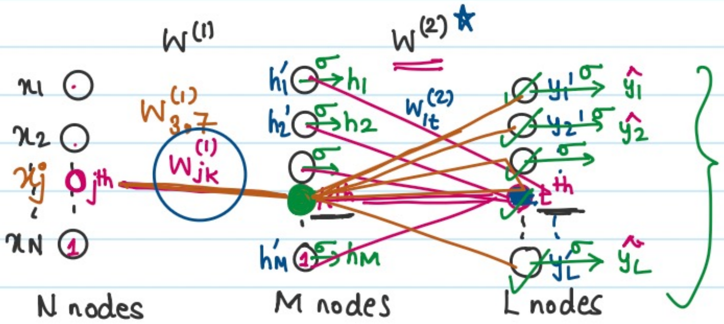
① Mean Squared error $= \frac{1}{N} \sum_{i=1}^{N} \| y_i - \hat{y}_i \|_2^2$

② Cross-entropy loss $= -\sum_{i=1}^{N} y_i \log \hat{y}_i$

→ Backpropagation

→ Gradient descent

$$W^{(t+1)} = W^{(t)} - \eta \cdot \frac{\partial L}{\partial W}$$



$$\hat{y}_j = \sum_{i=1}^{M} h_i w_{ij}^{(2)} \qquad (j = t)$$

gradient – direction of steepest ascent

$$\frac{\partial L}{\partial W^{(2)}_{kt}} = \frac{\partial L}{\partial \hat{y}_t} \underbrace{\boxed{\frac{\partial \hat{y}_t}{\partial W^{(2)}_{kt}}}}_{= h_k}$$

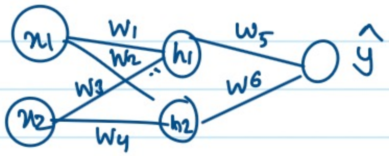
$$\hat{y}_t = \sum_{k=1}^{M} h_k W^{(2)}_{kt}$$

$$= h_1 W^{(2)}_{1t} + h_2 W^{(2)}_{2t} - - - h_k W^{(2)}_{kt} + h_{k+1} W^{(2)}_{k+1t} + \cdots h_M W^{(2)}_{Mt}$$

$$\frac{\partial h_k W^{(2)}_{kt}}{\partial W^{(2)}_{kt}} = h_k$$

$$\frac{\partial L}{\partial W^{(2)}_{kt}} = \frac{\partial L}{\partial \hat{y}_t} \cdot h_k .$$

$$\frac{\partial L}{\partial W^{(1)}_{jk}} = \sum_{t=1}^{L} \frac{\partial L}{\partial \hat{y}_t} \cdot \frac{\partial \hat{y}_t}{\partial h_k} \cdot \frac{\partial h_k}{\partial W^{(1)}_{jk}}$$



$$\hat{y} = w_5 h_1 + w_6 h_2$$

$$= w_5 (x_1 w_1 + x_2 w_2) + w_6 (w_3 x_2 + w_4 x_2)$$