Assignment-02

K. Surya Prakash
EE18BTECH11026

Theory

Q1) Degrees of freedom for homography = 8

$\Rightarrow$ We need atleast $n = \lceil \frac{d}{2} \rceil$ points to compute transformation

$$\Rightarrow \quad n = \left\lceil \frac{8}{2} \right\rceil = 4$$

$\Rightarrow$ given, $w = 0.5$

Prob. that the algo never selects n ~~per~~ inlier points for k iterations is given by

$$(1-w^n)^k = 1-0.95$$

$$\Rightarrow (1-(0.5)^4)^k = 0.05$$

$$k = \frac{\log(0.05)}{\log(1-(0.5)^4)}$$

$$k = 46.41 \simeq 47$$

$\therefore$ 47 iterations are required to have 95% chance of corrected computation

Q2) $\frac{\partial f}{\partial w_{ij}^{(1)}} = ?$

Soln.

$w_{ij}^{(1)}$: Weight of link from jth to ith node after 1st layer

$$\frac{\partial f}{\partial w_{ij}^{(1)}} = \frac{\partial f}{\partial h^2} \cdot \frac{\partial h^2}{\partial h_i^1} \cdot \frac{\partial h_i^1}{\partial w_{ij}^{(1)}}$$

* $f = \langle w^3, h^2 \rangle$

$$\therefore \frac{\partial f}{\partial h^2} = w^3 = \begin{bmatrix} w^3_1 \\ w^3_2 \end{bmatrix}$$

* $h^2 = \sigma(w^2 h^1)$

$\sigma(\cdot) \rightarrow$ Sigmoid fn.

$$\boxed{\sigma'(\cdot) = \sigma(\cdot)(1-\sigma(\cdot))}$$

$$\Rightarrow \frac{\partial h^2}{\partial h_i^1} = \sigma'(w^2 h^1) \cdot w_i^{(2)}$$

$$= \sigma(w^2 h^1)(1-\sigma(w^2 h^1)) \cdot w_i^{(2)}$$

$$\boxed{\frac{\partial h^2}{\partial h_i^1} = h^2(\cdot)(1-h^2) \cdot w_i^{(2)}}$$

Here $w_i^2 = \begin{bmatrix} w_{1i}^{(2)} \\ w_{2i}^{(2)} \end{bmatrix} \Rightarrow$ ith column in w^2

* $h^1 = \sigma(w^1 x) \Rightarrow h_i^1 = \sigma(\sum_j w_{ij}^1 x_j)$

$$\Rightarrow \frac{\partial h_i^1}{\partial w_{ij}^{(1)}} = \sigma'(\sum_j w_{ij}^{(1)} x_j) \cdot x_j$$

$$= \sigma(\sum_j w_{ij}^{(1)} x_j) \; (1 - \sigma(\sum w_{ij}^{(1)} x_j)) \cdot x_j$$

$$\boxed{\frac{\partial h_i^1}{\partial w_{ij}^{(1)}} = h_i^{(1)} (1 - h_i^{(1)}) \cdot x_j} \Big\} \text{ scalar}$$

$$\frac{\partial f}{\partial w_{ij}^{(1)}} = \langle w^3, h^2(\cdot)(1 - h^{(2)})(\cdot), w_i^{(2)} \rangle * h_i^{(1)} * (1 - h_i^{(1)}) \cdot x_j$$

$$\boxed{= \left(\sum_K w_K^3 \cdot h_K^2 \cdot (1 - h_K^2) \cdot w_{Ki}^{(2)} \right) \cdot h_i^{(1)} \cdot (1 - h_i^{(1)}) \cdot x_j}$$ //

Q3) $\Delta_{ij}^{(2)} := \Delta_{ij}^{(2)} + \delta_i^{(3)} * (a^{(2)})_j$

In <u>vector form</u>:

$$\Delta^{(2)} = \Delta^{(2)} + (\delta)^{(3)} \cdot (a^{(2)})^T$$

d inputs, M hidden units, c outputs

Q4)

a) <u>No. of weights</u> :

$M \times d + M \times c = M(d+c)$

b) <u>No. of biases</u> :

$M + C$

c) <u>No. of independent derivatives</u> :-

For 2nd hidden layer

$\frac{\partial E}{\partial W_2} = \delta^3 (a^2)^T$ } c independent derivatives

For 1st hidden layer

$\frac{\partial E}{\partial W_1} = \delta^2 (a^1)^T$ } M independent derivatives

∴ Total no. of independent derivatives

$= M + C$ //

Q5) Showing minimizing Sum of Square error is equivalent to MLE

⇒ Let x_n be the input
W: weights

$f(x_n, W)$: Estimate of nueral network

y_n : target data

Sum of Squares error is defined as for 'N' datapoints

$$\boxed{SOS = \sum_{i=1}^{N} (y_i - \tilde{y}_i)^2}$$

where $\tilde{y}_1 = f(x_n, w)$

* Considering that the target data is of the form

$$y_n = f(x_n, w) + \varepsilon_n$$

$\varepsilon_n \sim N(0, \Sigma)$: Multivariate gaussian

let y_n is a 'd' dim data point

$\varepsilon_n \to$ also is a $(d \times 1)$ vector.

$\Rightarrow$ Since for a given input x_n, the estimate is deterministic using 'w'.

$f(x_n, w)$ is not a RV.

$$\Rightarrow \boxed{\underline{y_n \mid x_n, w} \sim N(f(x_n, w), \Sigma)} \Rightarrow d \times 1$$

$\Rightarrow$ It can be seen in a way that y_n has been drawn from a normal distribution of mean : $f(x_n, w)$ } x_n, w are given
Covariance : Σ

$$P(y_n \mid x_n, \Sigma) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left\{ \frac{-1}{2} (y_n - \mu_n)^T \Sigma^{-1} (y_n - \mu_n) \right\}$$

where $\mu_n = f(x_n, w)$

* Considering

N independent data points are drawn

Likelihood of collection of N data points

$$\prod_{i=1}^{N} P(y_i / x_i, \Sigma)$$

$$= \prod_{i=1}^{N} \left(\frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left\{ -\frac{1}{2} (\cancel{x} y_i - \mu_i)^T \Sigma^{-1} (y_i - \mu_i) \right\} \right)$$

where $\mu_i = f(x_i, w)$

* Our goal is to find the parameter (W), so that the likelihood is maximum.

Assuming Σ is constant

$$\Rightarrow W^* = \arg\max_{w} \prod_{i=1}^{N} P(y_i / x_i, \Sigma)$$

log-likelihood

$$\log\left(\prod_{i=1}^{N} P\left(y_i / x_i, \Sigma\right)\right) =$$

$$\underbrace{-\frac{N}{2}\log|\Sigma| - \frac{Nd}{2}\log(2\pi)}_{\text{constant}} - \frac{1}{2}\sum_{i=1}^{N}(y_i - \mu_i)^T \Sigma^{-1}(y_i - \mu_i)$$

The objective function here is

$$\max\left\{\log\left(\prod_{i=1}^{N} P\left(y_i / x_i, \Sigma\right)\right)\right\}$$

$$= \max\left\{-\frac{1}{2}\sum_{i=1}^{N}(y_i - \mu_i)^T \Sigma^{-1}(y_i - \mu_i)\right\}$$

$$= \min\left\{\sum_{i=1}^{N}(y_i - \mu_i)^T \Sigma^{-1}(y_i - \mu_i)\right\}$$

$$\therefore \underbrace{w^*}_{\substack{\text{optimum} \\ w}} = \arg\min_{w}\left(\sum_{i=1}^{N}(y_i - \mu_i)^T \Sigma^{-1}(y_i - \mu_i)\right)$$

$$\mu_i = f(x_i, w)$$

$$\underbrace{w^* = \arg\min_{w}\left(\sum_{i=1}^{N}(y_i - f(x_i, w))^T \Sigma^{-1}(y_i - f(x_i, w))\right)}_{\text{MLE}}$$

For MLE: error function:

$$e_1(w) = \sum_{i=1}^{N} (y_i - f(x_i, w))^T \Sigma^{-1} (y_i - f(x_i, w)) \rightarrow (1)$$

For a NN using sum-of-squares error function is:

$$e_2(w) = \sum_{i=1}^{N} \| y_i - f(x_i, w) \|^2 \Big\} \text{ Sum of squares} \rightarrow (2)$$

$\therefore$ For $\Sigma = \sigma^2 I$: I: identity matrix

$\Rightarrow$ eq (1)

$$e_1(w) = \frac{1}{\sigma^2} \sum_{i=1}^{N} (y_i - f(x_i, w))^T I (y_i - f(x_i, w))$$

$$= \frac{1}{\sigma^2} \sum_{i=1}^{N} \| y_i - f(x_i, w) \|^2$$

$\therefore$ we can see that:

$$\arg\min_w (e_1(w)) = \arg\min_w (e_2(w))$$

Both are equivalent if $\boxed{\Sigma = \sigma^2 I}$ //

Q6) Scale-Space Symmetry:

1)

Problem: Vanishing gradient problem

* ~~Let~~ ~~left layers~~ are ~~scaled~~

Let incoming layers are scaled by 'γ'

& outgoing by '$\frac{1}{\gamma}$'

→ During Accumulation of gradients in back-propagation:

→ Let (δ_l, δ_l') are the gradients before and after scaling:

→ For before layer: $(\delta_{l-1}, \delta_{l-1}')$

After " $(\delta_{l+1}, \delta_{l+1}')$

Gradient of Weights

$$\Delta_l' = \delta_{l+1}' \cdot a_l'$$

$$= \delta_{l+1} * \gamma * a_l$$

$$\boxed{\Delta_l' = \gamma \Delta_l}$$

Since, l is changed

$\delta_{l+1} = \delta_{l+1}'$

$\delta_l = \frac{1}{\gamma} \delta_l'$

$\delta_{l-1} = \delta_{l-1}'$

Hence, γ is scaled,

if γ is very small, while computing initial layer's gradient → 0

Q.6)

b) The permutation-symmetry in weight space.

Since every node is connected to all the nodes in the previous layer. This makes the weights ~~&~~ nodes / the neural network invariant to permutation of nodes.

⇒ This leads to permutation-symmetry in weight space of each layer.

Consequences:

1) Gives rise to multiple-equal global minima of the loss function.

2) Also creates saddle points on the path between these minima.

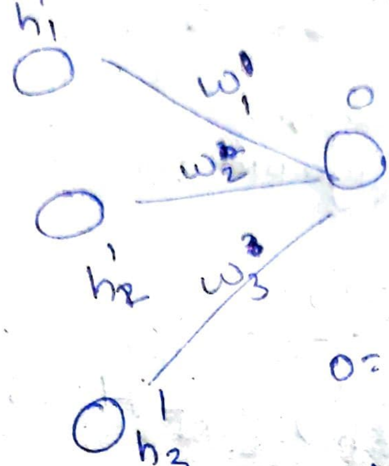
---

* For a multilayer network with d-1 hidden layers → each with 'n' neurons.

n! : No. of possible permutation per layer

∴ Total no. of equivalent configuration which yield the same ~~loss~~ error = $(n!)^{d-1}$

Eg:

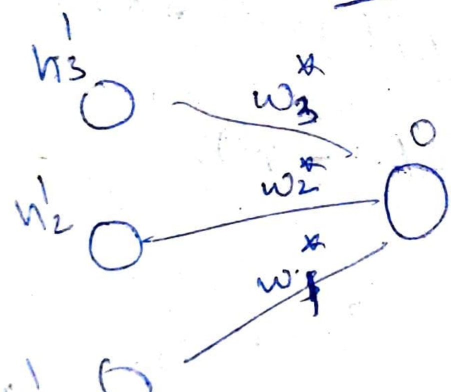


$\sigma = \sigma(\Sigma w'$

$o = \sigma(w_1 h_1' + w_2 h_2' + w_3 h_3')$

$w_1 = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$

⇒ Permutated : the nodes



$w_2 = \begin{bmatrix} w_3^* \\ w_2^* \\ w_1^* \end{bmatrix}$

$o = \sigma(w_3 h_3' + w_2 h_2' + w_1 h_1')$

for $\bar{w}_1 = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$; $\bar{w}_2 = \begin{bmatrix} w_3 \\ w_2 \\ w_1 \end{bmatrix}$

→ Yield the same activation

→ Thus will yield same loss

- Hence they are permutation symmetric