Deep Learning Assignment 1
Theory :

Raj Patil : CS18BTECH11039

1. a) zero padding the signal such that the final activations ~~are of~~ is of the same size

$$I' = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 1 \\ 0 & 1 & -1 & 2 \end{bmatrix}$$

$$F = \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$$, flipping F & multiplying

$$F * I' = \cancel{\begin{bmatrix} -2 & 2 & -1 \\ -3 & 4 & -4 \end{bmatrix}} \begin{bmatrix} 2 & -2 & 1 \\ 3 & -4 & 4 \end{bmatrix}$$

b) $F = \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix} \begin{bmatrix} c & d \end{bmatrix}$

$\therefore ac = 1$

$ad = -1$ $\quad a = b$

$bc = 1$ $\quad c = -d$

$bd = -1$

$a = b = c = 1$

$d = -1$

$$\therefore F = \underbrace{\begin{bmatrix} 1 \\ 1 \end{bmatrix}}_{F_1} \underbrace{\begin{bmatrix} 1 & -1 \end{bmatrix}}_{F_2}$$

$\therefore$ $(F_1 * I') = \cancel{\begin{bmatrix} 0 & 2 & 0 & 1 \\ 0 & 3 & -1 & 3 \end{bmatrix}}$

flipping F_1 & ...

$$\begin{bmatrix} 0 & 2 & 0 & 1 \\ 0 & 3 & -1 & 3 \end{bmatrix}$$

$F_2 * (F_1 * I) = \cancel{\begin{bmatrix} 2 & -2 & 1 \\ 3 & -4 & 4 \end{bmatrix}}$

flipping F_2 & ..

$$\begin{bmatrix} 2 & -2 & 1 \\ 3 & -4 & 4 \end{bmatrix}$$ $\therefore$ same as $F * I$.

c) P.t. $F * I' = F_2 * (F_1 * I')$

defining row & collumn convolutions for F_1 & F_2 mathematically :-

$\cancel{(F_1 * I)_{i,j} = \sum_k I[i-k][j]}$

$$(F_1 * I)_{i,j} = \sum_k I_{i-k,j} \, F_{1k}$$

& $$(F_2 * I)_{i,j} = \sum_l I_{i,j-l} \, F_{2l}$$

~~Using these in equation (I)~~ :

Using these to obtain equation (I)

...

$$\left(F_2 * (F_1 * I)\right)_{i,j}$$

$$= \sum_{l} F_{2l} \left((F_1 * I)_{i,j-l} \right)$$

$$= \sum_{l} F_{2l} \left(\sum_{k} I_{i-k,j-l} \, F_{1k} \right)$$

$$= \sum_{l} \sum_{k} I_{i-k,j-l} \, F_{1k} \, F_{2l}$$

~~using~~ $\cancel{F = F_1 F_2 \iff F_{k,l} = F_{1k} * F_{2l}}$

now, observe that ..

$$F_{i,j} = F_{1i} \cdot F_{2j}$$

$$\therefore \left(F_2 * (F_1 * I)\right)_{i,j}$$

$$= \sum_{l} \sum_{k} I_{i-k,j-l} \, F_{k,l}$$

$$= (F * I)_{i,j}$$ (by equation (I) given in question statement)

$$\forall i,j \left(\left(F_2 * (F_1 * I)\right)_{i,j} = (F * I)_{i,j} \right)$$

$$\iff F_2 * (F_1 * I) = F * I$$ q.e.d.

Page No.:
Date: / /

d) part a) for each activation, we needed 4 multiplications & ~~there are~~ 6 activations

∴ total multiplications for part a = $6 \times 4 = \boxed{24}$

part b) for each intermediate activation, we needed 2 multiplications

there $\overset{F_1 * I}{\cancel{8}\,5}$ + $\overset{\nearrow F_2 * F_1 * I}{6}$ such activations

∴ total multiplications needed:

$14 \times 2 = \boxed{28}$

~~#~~ the first ~~per~~ way requires lesser multiplications but that is only because the problem size is small yet ~~to~~ this will change for a larger computation.

e) for simple convolution:

final output size (no zero padding) is

$$(M_1 - M_2 + 1) \times (N_1 - N_2 \times 1)$$

for each activation, we need $M_2 \times N_2$ multiplications

∴ total multiplications =

$$M_2 \times N_2 \times (M_1 - M_2 + 1) \times (N_1 - M_2 + 1)$$

$$\in O(M_2 N_2 M_1 N_1)$$

---

for consecutive 1d convolutions, the output size transitions as follows

$$M_1 \times N_1 \xrightarrow{F_1 * I} (M_1 - M_2 + 1) \times N_1 \xrightarrow{F_2 * (F_1 * I)} (M_1 - M_2 + 1) \times (N_1 - N_2 + 1)$$

M_2 multiplications for each output.

N_2 multiplications for each output.

Page No.
Date / /

∴ total multiplications =

$M_2 \times (M_1 - M_2+1) \times N_1$

$+ \; N_2 \times (M_1 - M_2 +1) \times (N_1 - N_2 +))$

$\in \; O(M_1 N_1 (M_2 + N_2).)$

Answers summarized.

i) $M_2 N_2 (M_1 - M_2 +1)(N_1 - N_2 + 1)$

ii) $M_2 (M_1 - M_2 + 1) N_1 + N_2 (M_1 - M_2 + 1)(N_1 - N_2 + 1)$

iii) 2 successive 1D convolutions are faster than a simple 2D convolution asymptotically as

$$\boxed{O(M_1 N_1 (M_2 + M_2)) \subset O(M_1 N_1 M_2 N_2)}$$

Page No.:
Date: / /

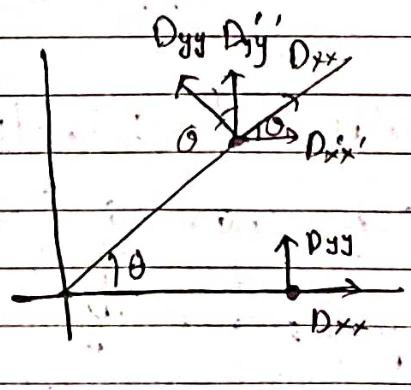
2-> a) for points on x axis, under the transformation application of

$$\begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \end{pmatrix} \begin{pmatrix} x \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} x\cos\theta \\ x\sin\theta \end{pmatrix}$$

Comparing Initial and final gradient magnitudes.

initially, as the edge is horizontal ~~Dxx is lesser than Dyy so we only look at Dyy for the final gradient magnitude.~~

~~Dyy~~ new gradients can be found as



$$D_{y'y'} = D_{yy}\cos\theta + D_{xx}\sin\theta$$

$$D_{x'x'} = D_{xx}\cos\theta - D_{yy}\sin\theta$$

Page No.
Date. / /

$\therefore$ new gradients magnitude =

$$\sqrt{(D_{y'y'})^2 + (D_{x'x'})^2}$$

$$= \sqrt{\left(D_{yy}^2 \cos^2\theta + D_{xx}^2 \sin^2\theta + \cancel{2 D_{xx} D_{yy} \sin\theta\cos\theta} \right.}$$

$$\overline{\left. + D_{yy}^2 \sin^2\theta + D_{xx}^2 \cos^2\theta - \cancel{2 D_{xx} D_{yy} \sin\theta\cos\theta} \right)}$$

$$= \sqrt{(D_{xx}^2 + D_{yy}^2)} = \text{old gradient magnitude.}$$

$\therefore$ The edge will still be detected by Canny edge detector.

b) Because the prominent edges are detected, the higher threshold (h) is low enough. To deal with the disconnected long edges, one needs decrease the lower threshold (l) for more neighboring pixels of the long edges to occur in the middle band. For the numerous spurious edges, increase h slightly so as to not make the long edges dissappear but only push the noisy gradient magnitudes into the middle band where they don't have neighboring strong edges.