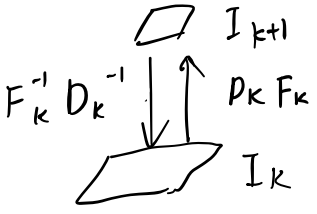
In solving the questions in the assignment, I worked together with my classmate [Shimiao Wang 1004779634]. I confirm that I have written the solutions / code / report in my own words.

CSC420 A2 Zhongliu Liu 1004706936

1. Let's assume the blurring filter as F_i,
Downsampling filter as D_i relates to I_i.
The upsampling filter as D^{-1}_i relates to I_{i+1}

From gaussian pyramid, we know:

$$I_{k+1} = D_k F_k I_k.$$



From Laplacian pyramid we know:

$$L_k = I_k - D_k^{-1} I_{k+1}$$

$$I_k = D^{-1}_k I_{k+1} + L_k$$

Assume we know I_n, we want to see what is I_0

$$I_0 = D^{-1}_0 I_1 + L_0$$

$$= D^{-1}_0 (D_1^{-1} I_2 + L_1) + L_0 \Rightarrow D^{-1}_0 D^{-1}_1 I_2 + D_0^{-1} L_1 + L_0$$

$$= D^{-1}_0 D^{-1}_1 (D_2^{-1} I_3 + L_2) + D_0^{-1} L_1 + L_0$$

$$= \prod_{i=0}^{n-1} D_i^{-1} I_n + \sum_{j=0}^{n-1} \left(\prod_{k=-1}^{j-1} D_k^{-1} \right) L_j \quad (\text{where } D_{-1}^{-1} = 1)$$

since when $j=0$. it is $D_{-1}^{-1} L_0 = L_0$.

when $j=1$ it is $D_{-1}^{-1} D_0^{-1} L_1 = D_0^{-1} L_1$.

when $j=2$ it is $D_{-1}^{-1} D_0^{-1} D_1^{-1} L_2 = D_1^{-1} D_0^{-1} L_2$

Since Laplacian pyramid is given. so we know L_i.
Therefore we still need downsampling method in Gaussian pyramid D_i.

The closed expression for I_0. is.

$$I_0 = \prod_{i=0}^{n-1} D_i^{-1} I_n + \sum_{j=0}^{n-1} \left(\prod_{k=-1}^{j-1} D_k^{-1} \right) L_j \quad (\text{where } D_{-1}^{-1} = 1)$$

2. Let X_i be the input at layer i, X_{i+1} be the output of X_i.
Let F_i be the function of $\hat{y} = w_i x_i + b_i$

$$\begin{aligned} X_{i+1} &= \sigma(W_i * X_i + b_i) \\ &= w_\sigma * (W_i * X_i) + w_\sigma * b_i + b_\sigma \\ &= W_i' X_i + b_i' \end{aligned}$$

which means the layer i is also a linear function.
$(\sigma \circ F_i)$ is linear.

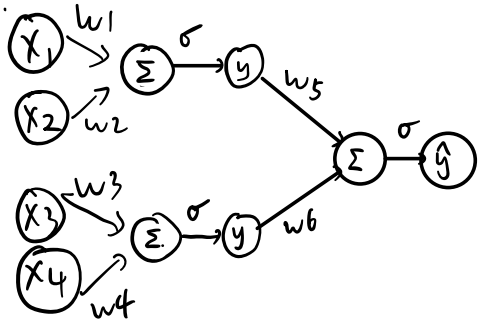
Let k be a arbitrary layer.

we have: $$\begin{aligned} X_k &= (\sigma \circ F_{i-1}) X_{i-1} \\ &= (\sigma \circ F_{i-1})(\sigma \circ F_{i-2}) X_{i-2} \\ &= (\sigma \circ F_{i-1})(\sigma \circ F_{i-2}) \cdots (\sigma \circ F_0) X_0 \\ &= W' X_0 + b_0' \end{aligned}$$

Since every $(\sigma \circ F_i)$ is linear, they can be compressed into one single linear function, and the output is still the linear combination of the input.

Therefore the number of layers has effectively no impact on the network.

3. a)



b) $w_1x_1 + w_2x_2 = -0.175$

$\sigma(w_1x_1 + w_2x_2) = 0.4564$ ①

$I_1: w_3x_3 + w_4x_4 = 1.945$

$I_2: \sigma(w_3x_3 + w_4x_4) = 0.875$ ②

$I_3: w_5① + w_6② = -0.05933 + 0.8137 = 0.7543$

$\hat{y} = \sigma(w_5① + w_6②) = 0.680$

Since $\frac{\partial J}{\partial \hat{y}} = \frac{d}{d\hat{y}}(y-\hat{y})^2 = -2(y-\hat{y})$

$$\frac{\partial J}{\partial w_3} = \frac{\partial J}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w^3}$$

$$= \frac{\partial J}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial I_3} \cdot \frac{\partial I_3}{\partial I_2} \cdot \frac{\partial I_2}{\partial I_1} \cdot \frac{\partial I_1}{\partial w_3}$$ # by chain rule.

$= -2(y-\hat{y}) \cdot \sigma(I_3)(1-\sigma(I_3)) \cdot w_6 \cdot \sigma(I_1)(1-\sigma(I_1)) \cdot x_3$

$= -2(1.0-0.68) \cdot 0.680 \cdot (1-0.680) \cdot 0.93 \cdot 0.875(1-0.875) \cdot 0.8$

$= -0.64 \times 0.2176 \times 0.93 \times 0.109375 \times 0.8$

$= -0.0113$

4. C Layer: The input size is $12 \times 12 \times 50$.

output side: $\frac{12-4+2}{2}+1 = 6$.

output size: $6 \times 6 \times 20$.

without bias: multiplication: 4×4, Addition: $4 \times 4 - 1$.

Calculation: $4 \times 4 + (4 \times 4 - 1) = 31$.

with bias: Calculation: $4 \times 4 + (4 \times 4 - 1) + 1 = 32$.

P Layer: The input size is $6 \times 6 \times 20$

output size: $\frac{6-3}{1}+1 = 4$

output size: $4 \times 4 \times 20$

with / without bias: max: $n-1 = 3 \times 3 - 1 = 8$.

Total flops with bias:

$6 \times 6 \times 20 \times 50 \times 32 + 4 \times 4 \times 20 \times 8 = 1154560$

without bias:

$6 \times 6 \times 20 \times 50 \times 31 + 4 \times 4 \times 20 \times 8 = 1118560$

5. C1: kernel size $= 32 - 28 + 1 = 5$.

trainable parameters $= 5 \times 5 \times 6 = 150$.

S2: subsampling needs no trainable parameters

C3: kernel size $= 14 - 10 + 1 = 5$.

trainable parameters $= 5 \times 5 \times 6 \times 16 = 2400$

S4: 0

C5: fully connected parameters $= 5 \times 5 \times 16 \times 120 + 120 = 48120$

C6: $120 \times 84 + 84 = 10164$

C7: $84 \times 10 + 10 = 850$

Total trainable parameter considering bias:

$150 + 2400 + 48120 + 10164 + 850 = 61684$

6. Logistic Activation function:

$$y = \frac{1}{1+e^{-x}}$$

$$\frac{dy}{dx} = -(1+e^{-x})^{-2} \cdot \frac{d}{dx}(1+e^{-x})$$

$$= -(1+e^{-x})^{-2} \left(\frac{d}{dx}1 + \frac{d}{dx}e^{-x}\right)$$

$$= -(1+e^{-x})^{-2} \left(\frac{d}{dx}e^{-x}\right)$$

$$= -(1+e^{-x})^{-2} e^{-x} \left(\frac{d}{dx}-x\right]$$

$$= -(1+e^{-x})^{-2} - e^{-x}$$

$$= \frac{e^{-x}}{(1+e^{-x})^2}$$

$$= \frac{1}{1+e^{-x}} \cdot \frac{e^{-x}}{1+e^{-x}}$$

$$= \frac{1}{1+e^{-x}} \left(\frac{1+e^{-x}}{1+e^{-x}} - \frac{1}{1+e^{-x}}\right)$$

$$= y(1-y).$$

Therefore we don't need input x for backward propagation if we have y.

7. a) $\tanh = \frac{1-e^{-2x}}{1+e^{-2x}} = \frac{1+e^{-2x} - 1+e^{-2x} +1-e^{-4x}}{1+e^{-2x}}$

$= 1 - \frac{2}{1+e^{-2x}}$

Since $\frac{2}{1+e^{-2x}} \in (0,1)$

$1 - \frac{2}{1+e^{-2x}} \in (-1,1)$

And sigmoid function $\frac{1}{1+e^{-x}} \in (0,1)$.

so they differ at range.

b). $\tanh(x) = \frac{1-e^{-2x}}{1+e^{-2x}} \cdot \frac{e^x}{e^x} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

$$\tanh'(x) = \frac{(e^x+e^{-x})\frac{\partial}{\partial x}(e^x-e^x) - (e^x-e^{-x})\frac{d}{dx}(e^x+e^{-x})}{(e^x+e^{-x})^2}$$

$$= \frac{(ex+e^{-x})(e^x-e^x) - (e^x-e^{-x})(e^x+e^{-x})}{(e^x+e^{-x})^2}$$

$$= 1 - \frac{(e^x-e^{-x})^2}{(e^x+e^{-x})^2} = 1 - \tanh^2(x).$$

logistic σ: $x = \frac{1}{1+e^{-x}}$

also $\tanh(x) = \frac{1-e^{-2x}}{1+e^{-2x}} = \frac{1}{1+e^{-2x}} - \frac{e^{-2x}+1-1}{1+e^{-2x}}$

$$= \frac{2}{1+e^{-2x}} - \frac{e^{-2x}+1}{1+e^{-2x}}$$

$$= 2\sigma(2x) - 1.$$

$\tan'h(x) = 1 - \tanh^2(x) = 1 - (2\sigma(2x)-1)^2 = 1 - (4\sigma^2(2x) - 4\sigma(2x) - 1)$

$= 4\sigma(2x) - 4\sigma^2(2x)$

c) tanh(x) has stronger gradients than sigmoid function. If we want big learning step. we shoud use tanh(x).

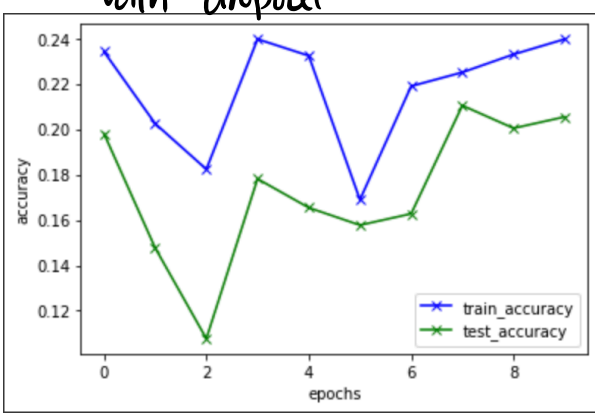
$\sigma(x)$ has an output between 0 and 1, so for probability predication. we can use $\sigma(x)$.

Part II

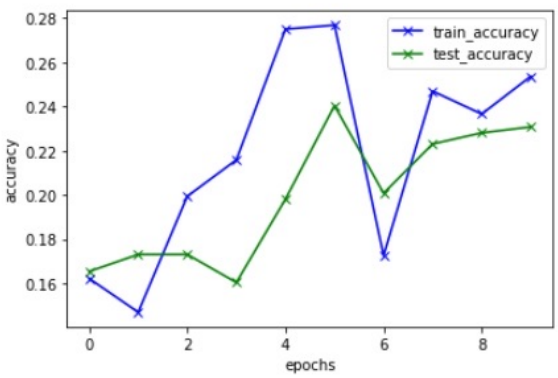
Task 1: The image in DBI shows the clear front face of the dog and the photo is taken in a medium distance. However, for SSD, the angle of photo could be very random and, some photos are taken far from dog, and dogs have various posture. some other dog or the human may also appear in this dataset.

Task 2:

with dropout



without dropout.



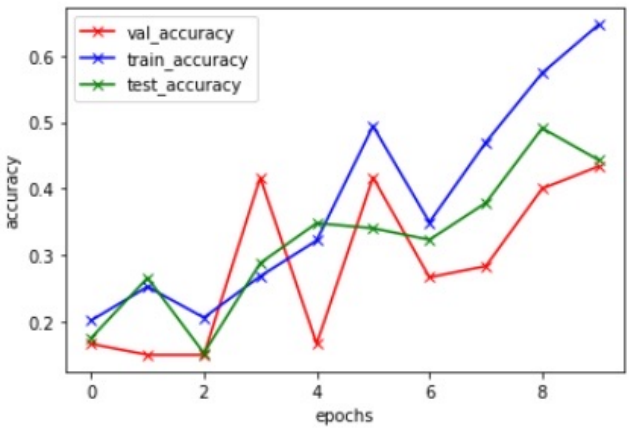
The test accuracy with dropout is around 0.2.

The test accuracy without dropout is around 0.23.

Task II:

a).

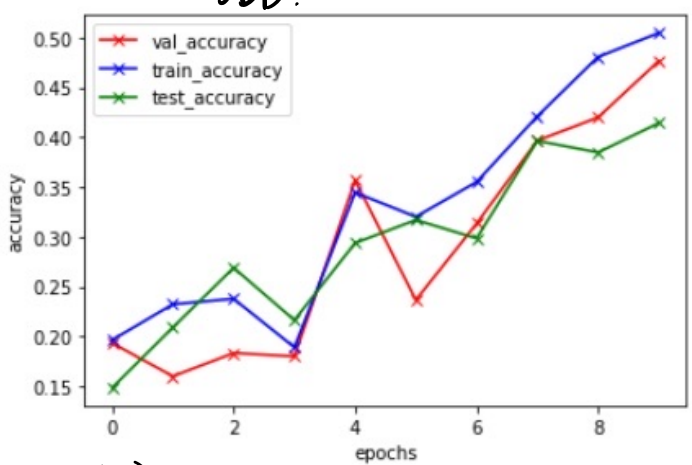
DBI



Comparing with my CNN in task I, the test accuracy improves a lot (from 0.2 to 0.4). The training accuracy improves very much, too (from 0.24 to 0.65). Additionally, not like my CNN have a high peak then drop, Resnet 18 has a increase in accuracy overall.

b). Comparing with DBI dataset. DBI has a higher test accuracy ($\approx$ 0.44), where SDD only has $\approx$0.41. test accuracy. plus, DBI has a clearly higher training accuracy 0.65 comparing to SDD (0.50).

SDD:

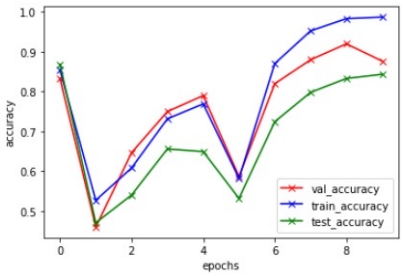


This might due to DBI has a more clearly dog front face which contain more valid details for classification of dog breed. While for SDD, the environment factor might also affect the training process.

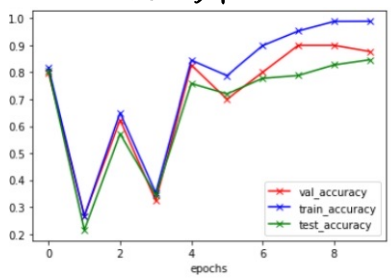
Task IV.

DBI:

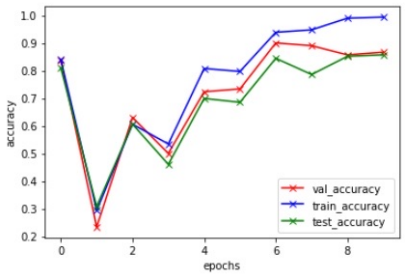
ResNet 18



ResNet 34

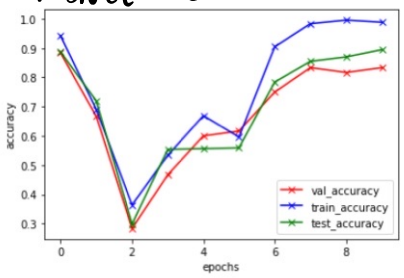


ResNetXt 32.

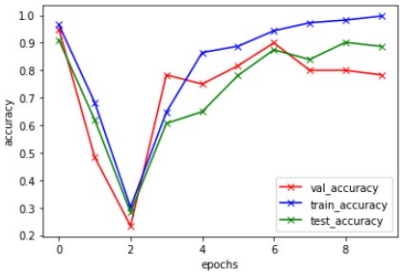


SDD.

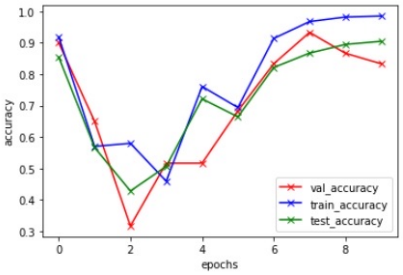
ResNet 18



ResNet 34



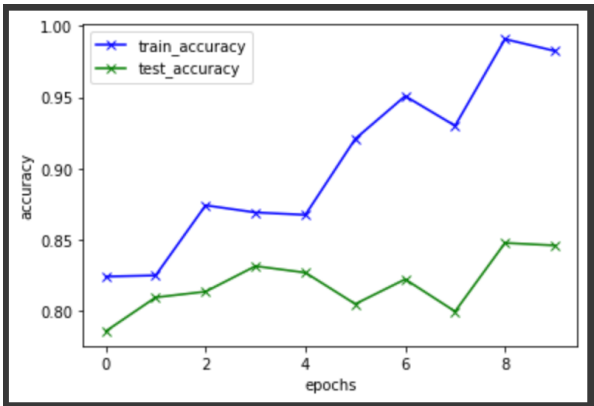
ResNetXt 32



- From the above graphs, we can see that ResNet18 in DBI and SDD have little difference. They both have a drop down of accuracy at the beginning then increase. and have another drop down at epoch 5, and increase sharply after, and finally gradually increase to a high peak.
- In the mean time, we can see that ResNet 34 and ResNetXt 32 shares almost the same structure in DBI. however they performed differently in SDD. In DBI, they both experienced two drop of accuracy at epoch 1 and epoch 3 and increase gradually after epoch 5, However in SDD, Resnet 34 only experienced one cleary drop at the beginning, and in ResNet Xt 32. It has a second drop on epoch 5 but the extent is small. Overall, ResNetXt32 having highest performance, test accuracy reached 0.85 in DBI dataset and 0.91 in SSD Dataset.

Task V

From task IV, we can tell ResNet32 has the best test accuracy on both DBI and SSD dataset. Therefore I choose it as the model for task V. Since the data set now is larger for one class. So I increase the batch size from 10 to 16. And the final accuracy for this process is 0.846




```
The final test accuracy for ResNet32, of DBI/SDI is:  0.8460343480110168
```