Part I

Q1 The minimum information requried is I_n in Gnassian pyramid.

By defination of Laplacian, F denotes upsmapling

$$L_i = I_i - F(I_{i+1}) \Rightarrow I_k = F(I_{i+1}) + L_i$$

Base case $I_0 = F(I_1) + L_0$

$$= F(F(I_2) + L_1) + L_0$$

$$\vdots$$

$$= F(F(\cdots(F(I_n) + L_{n-1}) + \cdots + L_1) + L_0$$

$$= \sum_{i=1}^{n} F(I_i + L_{i-1})$$

$\therefore$ the minimum information require is I_n.

Q2. Let x be the input, W be the layer weight, and F be the activation function matrix of function f.

$$f(Wx+b) = FWx + Fb$$

So it is still a linear relationship for the activation function layers.

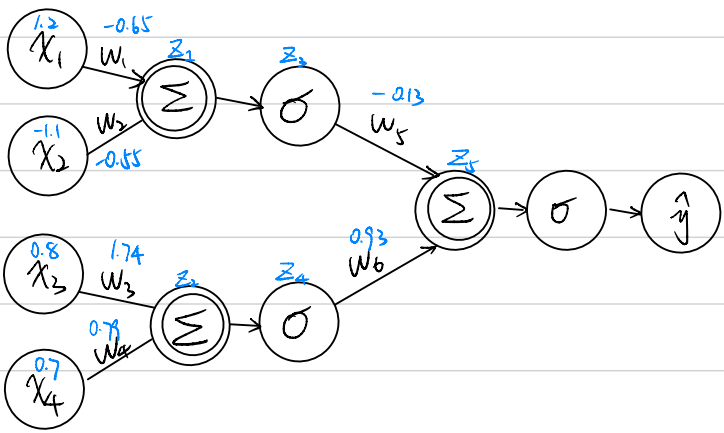
For n layers we have $\hat{y} = f(W_n f(W_{n-1} \cdots f(W_1 x))) = (f \circ W_n) \circ \cdots \circ (f \circ W_1) x$

Which is also an linear relationship can be rewritten as Wx.

∴ The number of layers has no impact on the network.

Q3.

(a)



(b) Value marked in blue are for part (b) for easy visulization.

$z_1 = w_1 x_1 + w_2 x_2 = -0.1750$ $\quad z_2 = w_3 x_3 + w_4 x_4 = 1.9450$

$z_3 = \sigma(z_1) = 0.4564$ $\quad z_4 = \sigma(z_2) = 0.8749$

$z_5 = w_5 z_3 + w_6 z_4 = 0.7543$ $\quad \hat{y} = \sigma(z_5) = 0.6801$

$\frac{\partial L}{\partial \hat{y}} = -2(y - \hat{y}) = -2(1.0 - 0.6801) = -0.6398$

$\frac{\partial L}{\partial z_5} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z_5} = \frac{\partial L}{\partial \hat{y}} (\hat{y} \cdot (1 - \hat{y})) = -0.6398 \cdot (0.6801 \cdot (1 - 0.6801)) = -0.1392$

$\frac{\partial L}{\partial z_4} = \frac{\partial L}{\partial z_5} \cdot \frac{\partial z_5}{\partial z_4} = \frac{\partial L}{\partial z_5} \cdot w_6 = -0.1392 \cdot 0.93 = -0.1295$

$\frac{\partial L}{\partial z_2} = \frac{\partial L}{\partial z_4} \cdot \frac{\partial z_4}{\partial z_2} = \frac{\partial L}{\partial z_4} \cdot (z_4(1 - z_4)) = -0.1295 \cdot (0.8749 \cdot (1 - 0.8749)) = -0.0142$

$\frac{\partial L}{\partial w_3} = \frac{\partial L}{\partial z_2} \cdot \frac{\partial z_2}{\partial w_3} = \frac{\partial L}{\partial z_2} \cdot x_3 = -0.0142 \cdot 0.8 = -0.0114$

Q4: For each filter, the size is $4 \times 4 \times 50 = 800$

The stride is 2, so there ar $\frac{12+2\cdot1-4}{2}+1 = 6$ positions and 20 filters

For each entry we have $4 \times 4 = 16$ multiplication and $4 \times 4 - 1 = 15$ additions $20 \times 6 \times 6 \times 50 \times (16+15) = 1116000$ FLOPs without bias and $20 \times 6 \times 6 \times 50 \times (16+15+1) = 1152000$ FLOPs with bias

The maxpooling for $6 \times 6 \times 20$ size require $\frac{6-3}{1}+1 = 4$, meaning max pooling size is 4×4. and each max pooling takes $3 \times 3 - 1 = 8$ FLOPs.

$\therefore 4 \times 4 \times 20 \times 8 = 2560$ FLOPs

So the number of FLOPs is $1116000 + 2560 =$ 1118560 FLOPs without bias and $1152000 + 2560 =$ 1154560 FLOPs with bias

Q5

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

$(5\times5+1)\times6=156$ parameters.

S2: 0 parameters

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

$(5\times5\times6+1)\times16=2416$ parameters.

S4: 0 parameters

C5: $5\times5\times16\times120=48000$ parameters.

F6: $120\times84+84\times10=10920$ parameters.

Total: $156+0+2416+0+48000+10920=61492$ parameters.

Q6: Assume input x, weight w and bias b.

$\therefore L = \sigma(wx+b)$

$\frac{\partial L}{\partial x} = w\sigma(wx+b)(1+\sigma(wx+b))$

$= wL(1-L)$

From the result, we can see we only need L and w in order to compute the derivative of output, no need for the inputs.

Q7. (a) The output range of the function is $(-1, 1)$ because $\lim_{x \to \infty} = 1$ and $\lim_{x \to -\infty} = -1$ and the output range of logistic function is $(0, 1)$

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

$$= \frac{4e^{-2x}}{(1+e^{-2x})^2}$$

$$= 1 - \tanh^2(x)$$

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

$\therefore$ gradient of tanh can be formulated as a function of logistic function.

(C) The usage of each activation function depends on the nature of the output range. For example, the $\tanh(x)$ preserved the sign of input, it may be more useful in binary classification.

And the logistic activation can be used for probability predication, since the output range is $(0, 1)$

Part II

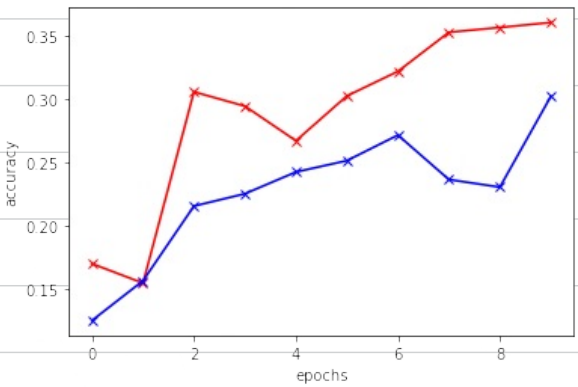
Task I:

DBI: The background is very simple or even removed, we can clearly identify the main object is the dog in the picture.

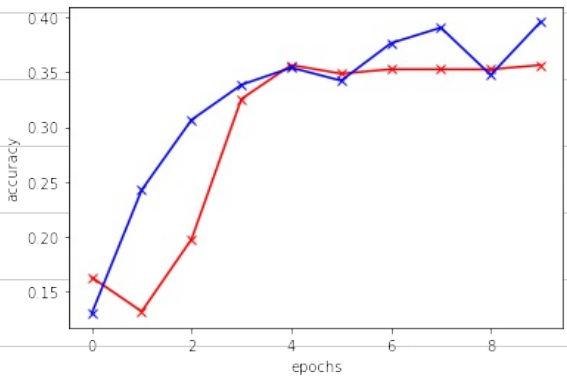
SDD: The sources of images seem to vary. The background is not so clear and may even hard to separate the dog. There are a lot of images contains non-relevance object like human, and decoration on the dog, food... And some of the images even contain water mark.

Task II

with dropout



without dropout



red : test accuracy

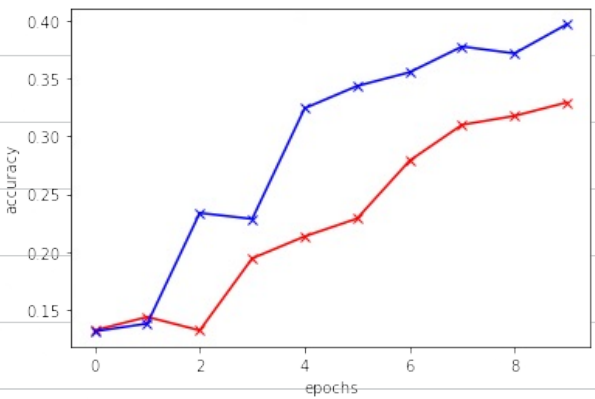
blue : train accuracy

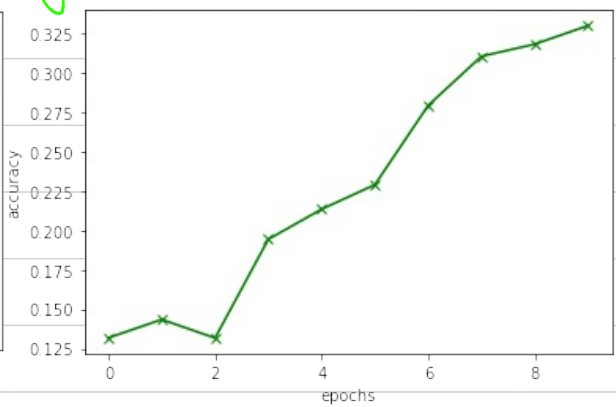
From two reports, we can see there is no obvious increase in the overall accuracy, but the fit speed seems to be faster without dropout. It may because the model we are training is relatively small and dropout in the small data made it even hard to fit.

Task III

a. ResNet-18 on DBI

red: test blue: train green: validation

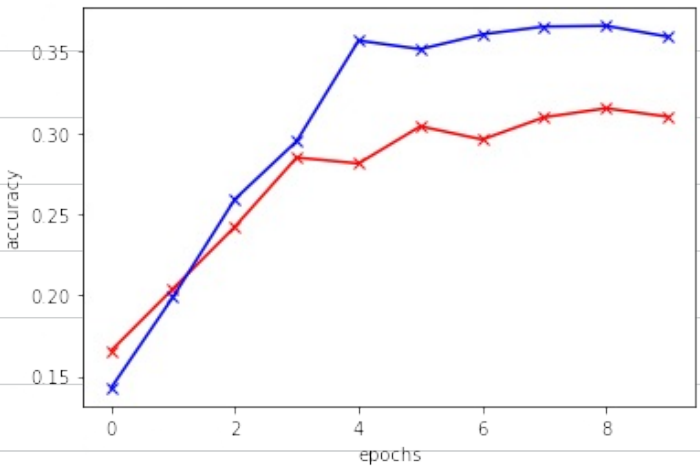




Comparing with CNN model, we can see the overall accuracy is about the same with test accuracy lower than the CNN model. But the trend seems to keep growing. So I think this model can achieve better accuracy than CNN with more training.

b. ResNet-18 on SDD

red: test blue: train



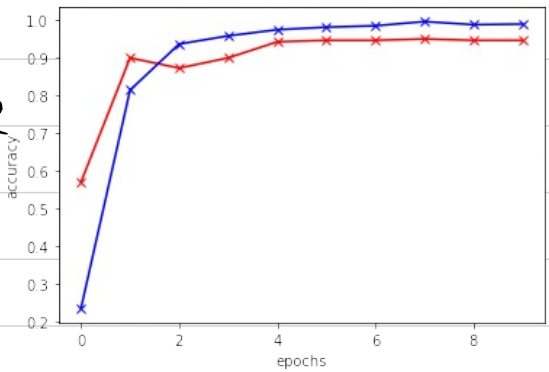
The overall accuracy is a little bit lower than the dadaset on DBI. The reason may be the background is more complex in SDD and the model may not correctly learned what is dog. It may learned the background like human -- as dog.

Task IV

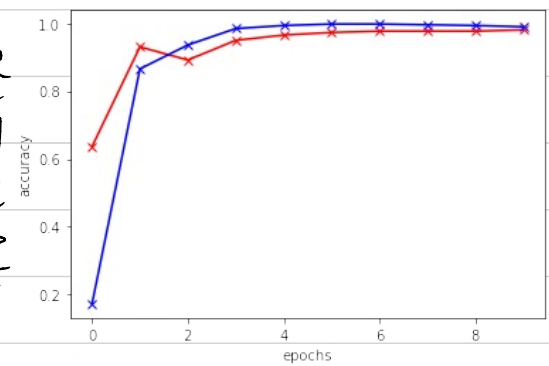
red test
blue train

DBI

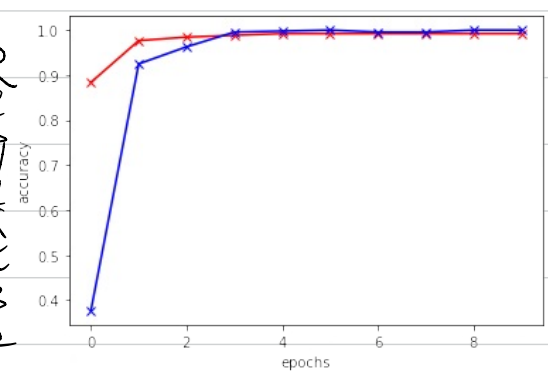
Resnet18



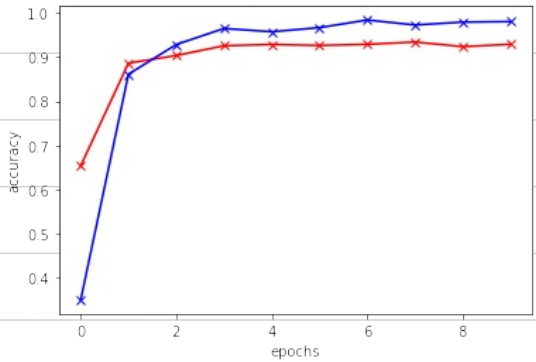
Resnet34

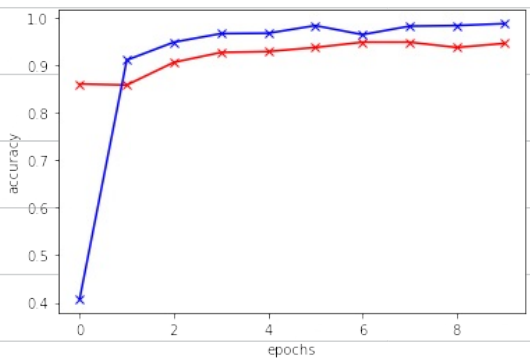


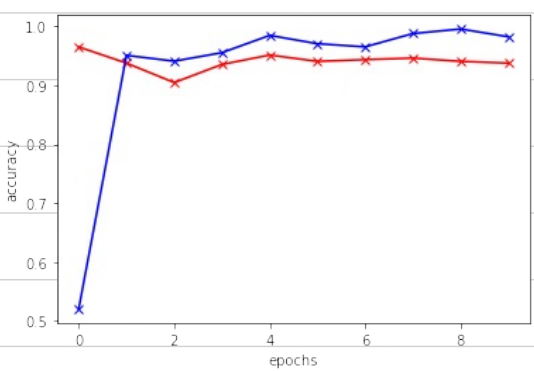
Resnext32



SDD



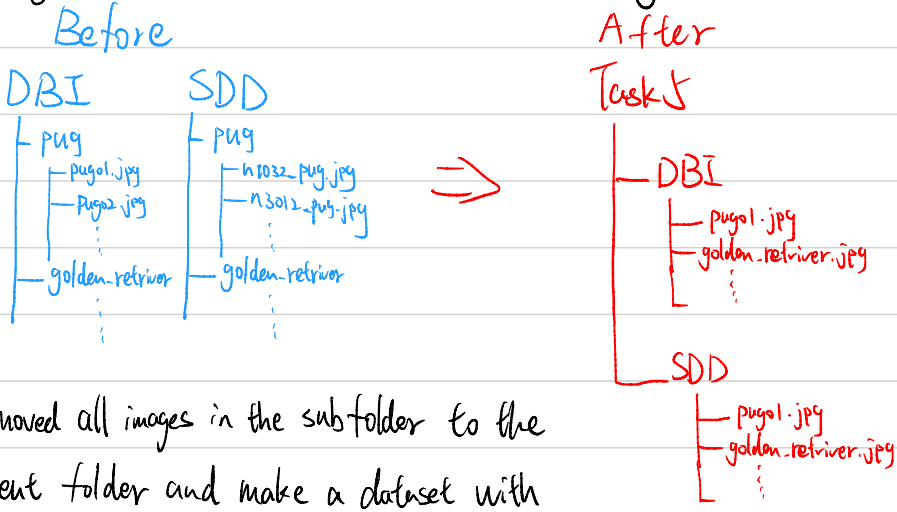




From the result, we can see these networks achieved similar results at the end with over 0.95 accuracy. In general, all networks are performing better in DBI set which is expected because the set is more clean. Among all, the ResNeXt32 seems to win from the test in the DBI set. There is no clear winner in SDD set but ResNeXt32 still has a little bit better accuracy among all.

Task V

In order to classifying the image is from SDD or DBI. I reorganized the datasets as following.



I moved all images in the subfolder to the parent folder and make a dataset with subset DBI and SDD. All images under DBI will have label 'DBI' and all images under SDD will have label 'SDD'. I modified the best resulting network from previous task – ResNeXt32 to get it learn from 2 features. The result is as following.