comp540 Homework5

Xiaoye Steven Sun (xs6), Wanyi Ye (wy13)

April 2nd 2018

1 Deep neural networks

1.1

We know that a shallow network could perform as good as the deeper ones. But from the *Deep Learning* book, the number of units in a shallow network grows exponentially with task complexity. This means that a shallow network might need to be very big - bigger than the deep one - in order to achieve the same performance as a deep network. Also from the text book: *Choosing a deep model encodes a very general belief that the function we want to learn should involve composition of several simpler functions. This can be interpreted from a representation learning point of view as saying that we believe the learning problem consists of discovering a set of underlying factors of variation that can in turn be described in terms of other, simpler underlying factors of variation.* This indicates that a deep network is more beneficial in real-life.

Furthermore, in mathematical perspective, the paper *When and Why Are Deep Networks Better than Shallow Ones?* proves that the deep network does not need to have exactly the same compositional architecture as the compositional function to be approximated, which means that the complexity of a deep network (with the same compositional architecture) is smaller than that of a shallow network.

1.2

The gradient can go towards 0 because of the horizontal line in ReLU (for negative X). Instead of the function being zero when x ¡ 0, a leaky ReLU will instead have a small negative slope (of 0.01, or so).

The leaky ReLU is used for solving the "dying ReLU" problem. For activations when x ¡ 0 in ReLU, gradient will be 0 because of which the weights will not get adjusted during descent. That is, those neurons which go into that state will stop responding to variations in error/input because the gradients is 0. This problem can cause several neurons to just die. The leaky ReLU can make it a slightly inclined line rather than horizontal line and hence solve the problem.

1.3

They are all CNN architectures used in ImageNet challenge.

1. **AlexNet**: contained only eight layers; the first five were convolutional layers, and the last three were fully connected layers; uses ReLU activation function instead of Sigmoid; uses "Dropout" instead of regularisation to deal with overfitting.

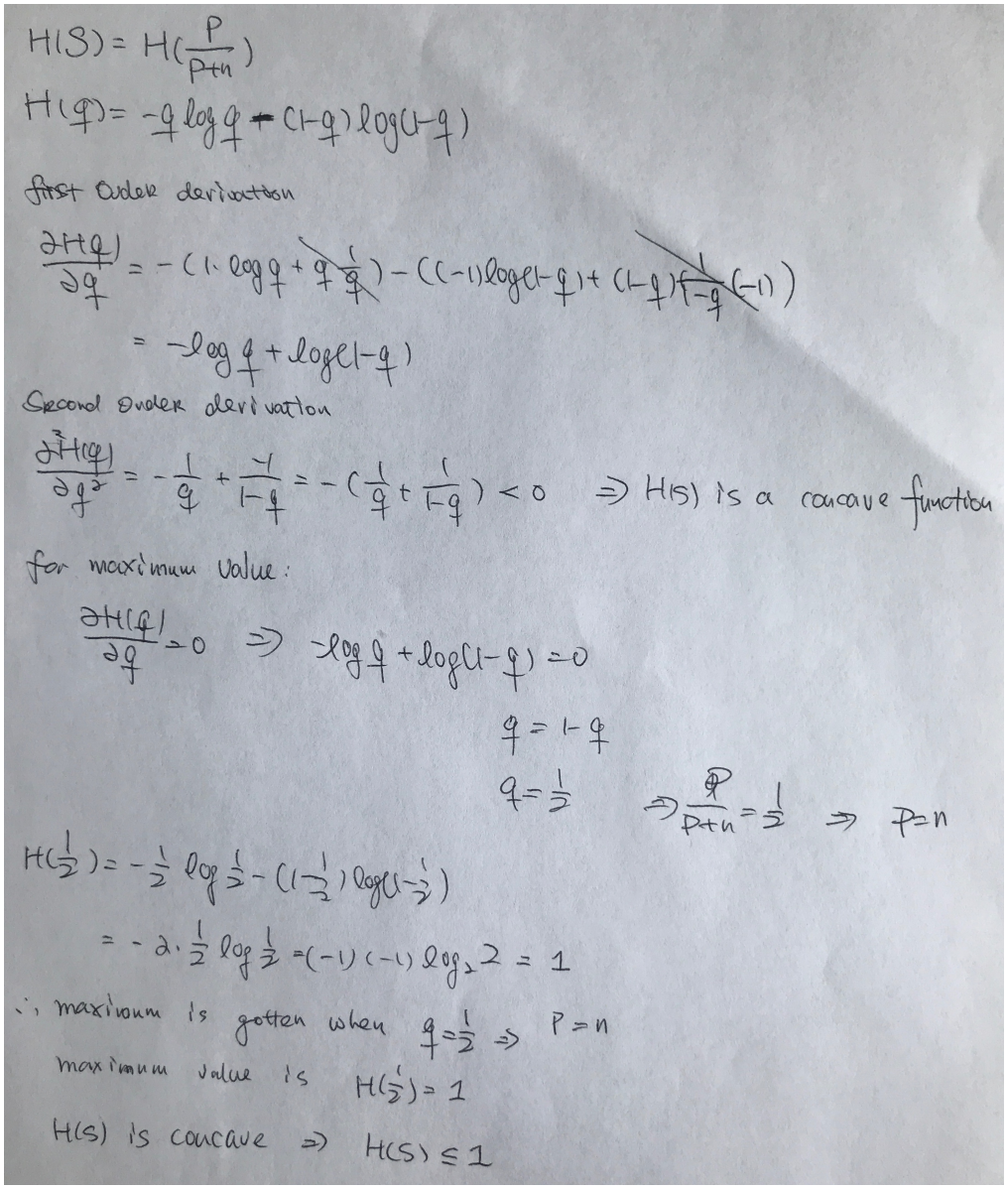
2. **VGGNet**: makes the improvement over AlexNet by replacing large kernel-sized filters with multiple 3X3 kernel-sized filters one after another. The original proposed VGG network was much deeper than the AlexNet.

3. **GoogleNet**: also known as Inception Module, consists of 22 layer in deep; approximates a sparse CNN with a normal dense construction since only a small number of neurons are effective; reduces the number of parameters.

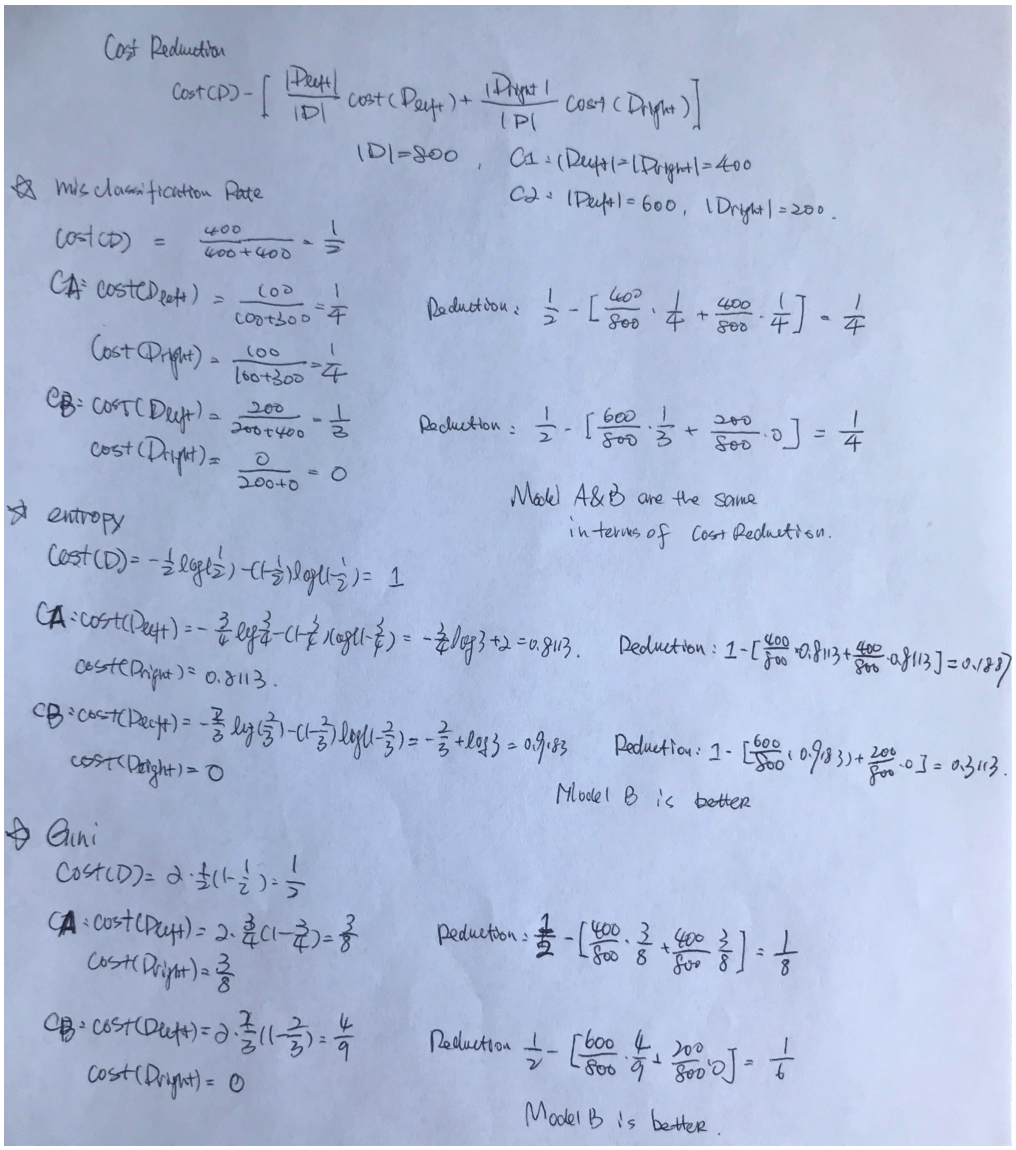
4. **ResNet**: add some additional layers learning only the residual; directly copy the input matrix to the second transformation output and sum the output in final ReLU function.

2 Decision trees, entropy and information gain

2.1



2.2



2.3

NO.

Assume that:
The set at the root node has D samples. The split creates set D_{left} and D_{right}. set D_{left} has L samples, including A_1 samples in Class A. set D_{right} has $D-L$ samples, including A_2 samples in Class A and including B_2 samples in Class B. So, in set D there are A_1+A_2 samples in Class A.

The question is actually asking, if the cost reeducation can be less than zero when the cost is the mis-classification rate. Without loss of generality, we assume that at root node, the majority class is A. So we have $cost(D) = (1-\frac{A_1+A_2}{D})$.

Without loss of generality, we assume that at the left node, the majority class is A as well. So we have $cost(D_{left}) = (1-\frac{A_1}{L})$. Hence we have $cost(D_{right}) = \frac{A_2}{D-L}$ (class B is the majority in D_{right}).

So the cost reduction is:
$cost(D) - [\frac{L}{D}cost(D_{left}) + \frac{D-L}{D}cost(D_{right})]$
$= (1-\frac{A_1+A_2}{D}) - \frac{L}{D}(1-\frac{A_1}{L}) - \frac{D-L}{D}\frac{A_2}{D-L}$
$= \frac{D-L-A_2-A_2}{D}$
$= \frac{B_2-A_2}{D} > 0$

3 Bagging

3.1

Proof

$$\begin{aligned}
E_{bag} &= E_X\left[\epsilon_{bag}(x)^2\right] \\
&= E_X\left[\left(\frac{1}{L}\sum_{l=1}^{L}(f(x)+\epsilon_l(x))-f(x)\right)^2\right] \\
&= \frac{1}{L^2}E_X\left[\left(\sum_{l=1}^{L}\epsilon_l(x)\right)^2\right] \\
&= \frac{1}{L^2}E_X\left[\sum_{l=1}^{L}\epsilon_l(x)^2+\sum_{\substack{1\le i,j\le L\\ i\neq j}}\epsilon_i(x)\epsilon_j(x)\right] \\
&= \frac{1}{L^2}E_X\left[\sum_{l=1}^{L}\epsilon_l(x)^2\right]+\frac{1}{L^2}E_X\left[\sum_{\substack{1\le i,j\le L\\ i\neq j}}\epsilon_i(x)\epsilon_j(x)\right] \\
&= \frac{1}{L^2}E_X\left[\sum_{l=1}^{L}\epsilon_l(x)^2\right]+\frac{1}{L^2}\sum_{\substack{1\le i,j\le L\\ i\neq j}}E_X\left[\epsilon_i(x)\epsilon_j(x)\right] \\
&= \frac{1}{L^2}E_X\left[\sum_{l=1}^{L}\epsilon_l(x)^2\right]
\end{aligned}$$

since $E_X\left[\epsilon_i(x)\epsilon_j(x)\right]=0$ for $i\neq j$

$$\begin{aligned}
E_{bag} &= \frac{1}{L^2}\sum_{l=1}^{L}E_X\left[\epsilon_l(x)^2\right] \\
&= \frac{1}{L}E_{avg}
\end{aligned}$$

3.2

Proof

$$\begin{aligned}
E_{bag} &= E_X\left[\epsilon_{bag}(x)^2\right] \\
&= E_X\left[\left(\sum_{l=1}^{L}\frac{\epsilon_l(x)}{L}\right)^2\right] \\
&\le E_X\left[\sum_{l=1}^{L}\frac{\epsilon_l^2(x)}{L}\right] \\
&= \frac{1}{L}\sum_{l=1}^{L}E_X\left[\epsilon_l^2(x)\right] \\
&= E_{avg}
\end{aligned}$$

where using Jensen's inequality with $\lambda_l=\frac{1}{L}$.