

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 $x \leq 0$, 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 $x \leq 0$ 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

$$H(S) = H\left(\frac{P}{P+n}\right)$$

$$H(q) = -q \log q + (1-q) \log(1-q)$$

first order derivation

$$\frac{\partial H(q)}{\partial q} = -\left(1 \cdot \log q + q \frac{1}{q}\right) - \left((1-q) \log(1-q) + (1-q) \cancel{(-1)}\right)$$

$$= -\log q + \log(1-q)$$

Second order derivation

$$\frac{\partial^2 H(q)}{\partial q^2} = -\frac{1}{q} + \frac{1}{1-q} = -\left(\frac{1}{q} + \frac{1}{1-q}\right) < 0 \Rightarrow H(S) \text{ is a concave function}$$

for maximum value:

$$\frac{\partial H(q)}{\partial q} = 0 \Rightarrow -\log q + \log(1-q) = 0$$

$$q = 1-q$$

$$q = \frac{1}{2} \Rightarrow \frac{P}{P+n} = \frac{1}{2} \Rightarrow P=n$$

$$H\left(\frac{1}{2}\right) = -\frac{1}{2} \log \frac{1}{2} - \left(1-\frac{1}{2}\right) \log\left(1-\frac{1}{2}\right)$$

$$= -2 \cdot \frac{1}{2} \log \frac{1}{2} = (-1)(-1) \log_2 2 = 1$$

\therefore maximum is gotten when $q=\frac{1}{2} \Rightarrow P=n$

maximum value is $H\left(\frac{1}{2}\right) = 1$

$H(S)$ is concave $\Rightarrow H(S) \leq 1$

2.2

Cost Reduction

$$\text{Cost}(D) = \left[\frac{|D_{left}|}{|D|} \text{cost}(D_{left}) + \frac{|D_{right}|}{|D|} \text{cost}(D_{right}) \right]$$

$$|D| = 800, \quad C_1 = |D_{left}| = |D_{right}| = 400$$

$$C_2 = |D_{left}| = 600, \quad |D_{right}| = 200.$$

⊗ misclassification Rate

$$\text{cost}(D) = \frac{400}{400+400} = \frac{1}{2}$$

$$CA = \text{cost}(D_{left}) = \frac{100}{100+300} = \frac{1}{4} \quad \text{Reduction: } \frac{1}{2} - \left[\frac{400}{800} \cdot \frac{1}{4} + \frac{400}{800} \cdot \frac{1}{4} \right] = \frac{1}{4}$$

$$\text{cost}(D_{right}) = \frac{100}{100+300} = \frac{1}{4}$$

$$CB = \text{cost}(D_{left}) = \frac{200}{200+400} = \frac{1}{3} \quad \text{Reduction: } \frac{1}{2} - \left[\frac{600}{800} \cdot \frac{1}{3} + \frac{200}{800} \cdot 0 \right] = \frac{1}{4}$$

$$\text{cost}(D_{right}) = \frac{0}{200+0} = 0$$

⊗ entropy

$$\text{cost}(D) = -\frac{1}{2} \log(\frac{1}{2}) - (1-\frac{1}{2}) \log(1-\frac{1}{2}) = 1$$

$$CA = \text{cost}(D_{left}) = -\frac{2}{3} \log(\frac{2}{3}) - (1-\frac{2}{3}) \log(1-\frac{2}{3}) = -\frac{2}{3} \log(\frac{2}{3}) + 2 = 0.8113. \quad \text{Reduction: } 1 - \left[\frac{400}{800} \cdot 0.8113 + \frac{400}{800} \cdot 0.8113 \right] = 0.1887$$

$$\text{cost}(D_{right}) = 0.8113.$$

$$CB = \text{cost}(D_{left}) = -\frac{2}{3} \log(\frac{2}{3}) - (1-\frac{2}{3}) \log(1-\frac{2}{3}) = -\frac{2}{3} \log(\frac{2}{3}) = 0.9183 \quad \text{Reduction: } 1 - \left[\frac{600}{800} \cdot 0.9183 + \frac{200}{800} \cdot 0 \right] = 0.3113.$$

$$\text{cost}(D_{right}) = 0$$

Model A & B are the same in terms of Cost Reduction.

⊗ Gini

$$\text{cost}(D) = 2 \cdot \frac{1}{2} \left(1 - \frac{1}{2} \right) = \frac{1}{2}$$

$$CA = \text{cost}(D_{left}) = 2 \cdot \frac{3}{4} \left(1 - \frac{3}{4} \right) = \frac{3}{8} \quad \text{Reduction: } \frac{1}{2} - \left[\frac{400}{800} \cdot \frac{3}{8} + \frac{400}{800} \cdot \frac{3}{8} \right] = \frac{1}{8}$$

$$\text{cost}(D_{right}) = \frac{3}{8}$$

$$CB = \text{cost}(D_{left}) = 2 \cdot \frac{2}{3} \left(1 - \frac{2}{3} \right) = \frac{4}{9} \quad \text{Reduction: } \frac{1}{2} - \left[\frac{600}{800} \cdot \frac{4}{9} + \frac{200}{800} \cdot 0 \right] = \frac{1}{6}$$

$$\text{cost}(D_{right}) = 0$$

Model B is better.

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 $\text{cost}(D) = (1 - \frac{A_1+A_2}{D})$.

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

So the cost reduction is:

$$\begin{aligned} \text{cost}(D) - [\frac{L}{D} \text{cost}(D_{left}) + \frac{D-L}{D} \text{cost}(D_{right})] \\ = (1 - \frac{A_1+A_2}{D}) - \frac{L}{D} (1 - \frac{A_1}{L}) - \frac{D-L}{D} \frac{A_2}{D-L} \\ = \frac{D-L-A_2-A_2}{D} \\ = \frac{B_2-A_2}{D} > 0 \end{aligned}$$

3 Bagging

3.1

Proof

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

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

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

3.2

Proof

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

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