Assignment 5 by nc41 & cq4

Problem 1 Deep neural networks

Why do deep networks typically outperform shallow networks?

More complex function can be combined in the deep network than in the shallow networks.

As a result, deep networks can be used to find the prediction which might be more similar to actual goal function.

What is leaky ReLU activation and why is it used?

In the normal ReLU, it kills all the nodes whose value is less than 0. It might kill too many nodes which might be useful. Therefore, leaky ReLU gives the nodes, whose value is less than 0, the second chance. These nodes multiply a very small number, such as 0.01, and these nodes can live and transport forward.

In one or more sentences, and using sketches as appropriate, contrast: AlexNet, VGGNet, GoogleNet and ResNet. What is the one defining characteristic of each network?

AlexNet: The Network had a very similar architecture to LeNet, but was deeper, bigger, and featured Convolutional Layers stacked on top of each other (previously it was common to only have a single CONV layer always immediately followed by a POOL layer).

Google Net: Their architecture consisted of a 22 layer deep CNN but reduced the number of parameters from 60 million (AlexNet) to 4 million. Additionally, this paper uses Average Pooling instead of Fully Connected layers at the top of the ConvNet, eliminating a large amount of parameters that do not seem to matter much

VggNet: VGGNet consists of 140 million parameters, which can be a bit challenging to handle.

ResNet: They were able to train a NN with 152 layers while still having lower complexity than VGGNet.

Problem 2 Decision trees, entropy and information gain

2.1

$H(P) = -\xi \log \xi - (1-\xi) \log (1-\xi)$

show $H(P/(p+n))$ $\quad H(S) \le 1 \quad H(S) = 1$ when $p = n$.

when $p = n$

$H(P/(p+n)) = H(\frac{1}{2}) = -\frac{1}{2}\log_2 \frac{1}{2} - \frac{1}{2}\log \frac{1}{2} = -\log_2 \frac{1}{2} = \log_2 2$

$= 1$ #

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

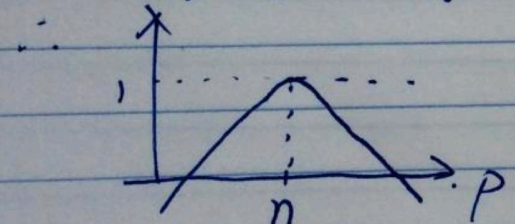
$H(q)' = -\log q - 1 + \log(1-q) + 1$

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

when $q < \frac{1}{2}$, $H(q)' > 0$ $\quad q < \frac{1}{2}$ means $p < n$

when $q > \frac{1}{2}$, $H(q)' < 0$ $\quad q > \frac{1}{2}$ means $p > n$

$\therefore$



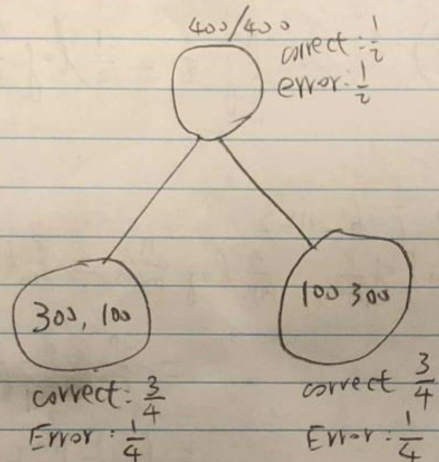
So for any given p and n, $H(S) \le 1$

2.2 & 2.3

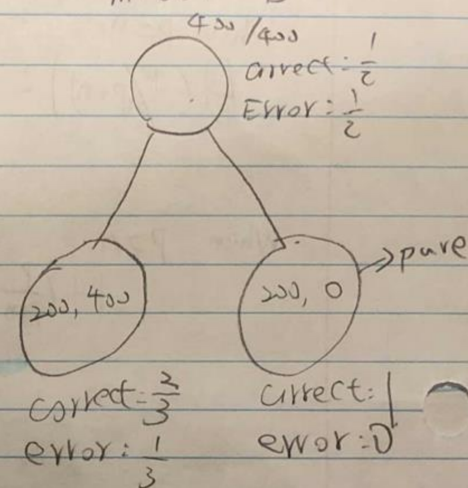
(2)

misclassification rate:

model A



model B

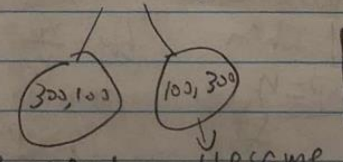


$\frac{1}{2} - \frac{1}{2} \times \frac{1}{400} \times 100 - \frac{1}{2} \times \frac{1}{400} \times 100$ $\qquad$ $\frac{1}{2} - \frac{3}{4} \times \frac{1}{600} \times 200 - \frac{1}{4} \times \frac{1}{200} \times 0$

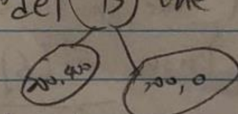
$\rightarrow = \frac{1}{2} - \frac{1}{4} = \underline{\frac{1}{4}}$ $\qquad$ $= \frac{1}{2} - \frac{1}{4} = \underline{\frac{1}{4}}$

model A and model B have the same information gain:

~~entropy~~: model A $H(S) = H(\frac{1}{2}) = 1$ $\quad$ model B the same $H(\frac{1}{2}) = 1$



$\boxed{\text{in this case model B > model A}}$



$-\frac{3}{4}\log\frac{3}{4} - \frac{1}{4}\log\frac{1}{4}$

$= -\frac{3}{4}(\log 3 + 2) + \frac{1}{2}$

$= \underline{0.81} \rightarrow 1 - \frac{1}{2} \times 0.81 - \frac{1}{2} \times 0.81$

$= 1 - 0.405 - 0.405 = \underline{0.19}$

$-\frac{2}{3}\log\frac{2}{3} - \frac{1}{3}\log\frac{1}{3}$ $\qquad$ $\frac{1 \cdot \log 1 + 0 \cdot \log 0}{1} \Rightarrow \underline{0}$

$= -\frac{2}{3}(\log 2 - \log 3) + \frac{1}{3}\log 3$

$= \log 3 - \frac{2}{3} \doteq 1.58 - 0.67$

$= \underline{0.91}$

$1 - \frac{3}{4} \times 0.91 - 0 \doteq \underline{0.32}$

Gini Index:

model A $\Rightarrow 2\times\frac{1}{2}(1-\frac{1}{2}) = \frac{1}{2}$

300, 100 $\rightarrow \frac{3}{4}$ $\quad$ 100, 300 $\rightarrow \frac{3}{4}$

$2\times\frac{3}{4}(1-\frac{3}{4}) = \frac{3}{8}$ $\quad$ $2\times\frac{3}{4}\times(1-\frac{3}{4}) = \frac{3}{8}$

$1-\frac{1}{2}\times\frac{3}{8}-\frac{1}{2}\times\frac{3}{8} = \frac{5}{8} = 0.625$

model B $\Rightarrow 2\times\frac{1}{2}(1-\frac{1}{2})$

200, 400 $\rightarrow \frac{2}{3}$ $\quad$ 200, 0 $\rightarrow 1 \rightarrow 0$

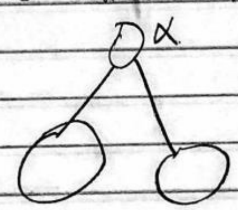
$2\times\frac{2}{3}\times\frac{1}{3} = \frac{4}{9}$ $\quad$ 0

$1-\frac{3}{4}\times\frac{4}{9}-0 = \frac{2}{3} \doteq 0.667$

In Gini Index information gain model B > model A

so we choose model B #

(3) If root misclassification rate is α



β error rate $\quad$ γ error rate

when spliting, the information gain > 0 means we can have progress when spliting

$\beta, \gamma \le \frac{1}{2}$ because error rate $= 1-\max(p, 1-p)$

$\alpha - \frac{|D_l|}{|D|}\cdot\beta - \frac{|D_r|}{|D|}\cdot\gamma$

assume $\Rightarrow \alpha - \frac{1}{100}\beta - \frac{99}{100}\gamma \Rightarrow$ the information gain still can be more than 0

if $\beta > \alpha \quad \gamma < \alpha$

So we still split, although β error rate will be more than α if the data in that branch is not larger enough to make information gain less than 0 the tree will still split.

Problem 3 Bagging

3.1

hw5 bagging

(1) $E_x[\{f(x)+\epsilon_l(x)-f(x)\}^2] = E_x[\epsilon_l(x)^2]$

$(h_l(x) = f(x)+\epsilon_l(x))$

$\boxed{E_{av}} = \frac{1}{L}\sum_{l=1}^{L} E_x[\epsilon_l(x)^2]$

$\boxed{h_{bag}(x)} = \frac{1}{L}\sum_{l=1}^{L} h_l(x) \rightarrow$ vote

$\epsilon_{bag}(x) = h_{bag}(x) - f(x) = \frac{1}{L}\sum_{l=1}^{L} h_l(x) - f(x) = \frac{1}{L}\sum_{l=1}^{L} f(x)+\epsilon_l(x) - f(x)$

$= \frac{1}{L}\sum_{l=1}^{L}\epsilon_l(x)$

$E_{bag} = E[\epsilon_{bag}(x)^2]$

if $E_x[\epsilon_l(x)] = 0$ $E_x[\epsilon_m(x)\epsilon_l(x)] = 0$ when $m \neq l$

$E_{bag} = E[\epsilon_{bag}(x)^2] = E[\frac{1}{L^2}(\sum_{l=1}^{L}\epsilon_l(x))^2]$

$= \frac{1}{L^2}E[\sum_{l=1}^{L}\epsilon_l(x)^2 + \sum_{l\neq m}\underbrace{\epsilon_l(x)\epsilon_m(x)}_{0}]$

$= \frac{1}{L^2}E[\sum_{l=1}^{L}\epsilon_l^2(x)]$

$= \frac{1}{L}\cdot\frac{1}{L}\sum_{l=1}^{L}E[\epsilon_l^2(x)]$

$= \frac{1}{L}E_{av}$

3.2

$f(x) = x^2$ here, $\lambda_l = \frac{1}{L}$

$$f\left(\sum_{1}^{L} \frac{1}{L} \epsilon_l\right) \leqslant \sum_{1}^{L} \frac{1}{L} f(\epsilon_l)$$

$$\Rightarrow \left(\sum_{1}^{L} \frac{1}{L} \epsilon_l\right)^2 \leqslant \sum_{1}^{L} \frac{1}{L} \epsilon_l^2$$

$$\Rightarrow E_{bag} \leqslant E_{av}$$

Extra

In this section, we tried several different artechitures, firstly using a small 6 layer net. The performance is bad. And since training on CPU is really slow. It is difficult to tune hyperparameters. We then implement it using Keras, and ran it on colab. However, later Devika said we can not use framework but just the API. We have to re-implement it using API.

So below is the result we achieved. And you can see more details from the two htmls file.

	Train Accuracy	Validation Accuracy	Test Accuracy
Keras(extra.html)	71.17%	73.4%	73.7%
Api(Conv....html)	94.3%	74.6 %	72.9 %