Homework 5

Young Won Kim (yk41) and Minh Pham (mnp7)

Spring 2017

1 Deep neural networks

· Why do deep networks typically outperform shallow networks?

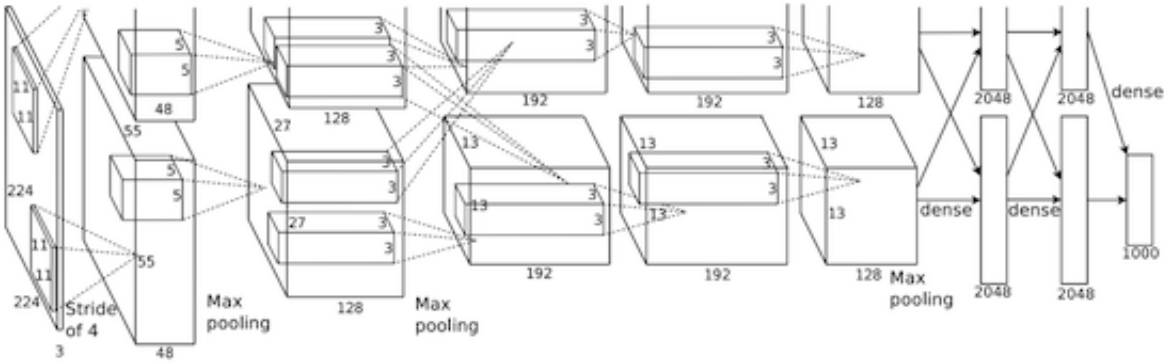
Deep networks typically outperform shallow networks because having multiple layers allow them to learn different aspects of data (e.i. edges, shapes, orientation, etc for images) at various levels. This also allows them to be better at generalizing because they can learn more of the intermediate features.

· What is leaky ReLU activation and why is it used?

Leaky ReLU is a modified verision of ReLU. It has a small negative slope (about 0.01) when $x < 0$, instead of the function being zero. This may be preferred because it attempts to avoid ReLU units "dying" during training. For example, if the learning rate is set too high during training, it can cause the gradients to be zero and forever be zero, leading to ReLU units to irreversibly die and never activate again. Leaky ReLU, by having a small gradient when $x < 0$, tries to fix such problem.

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

AlexNet



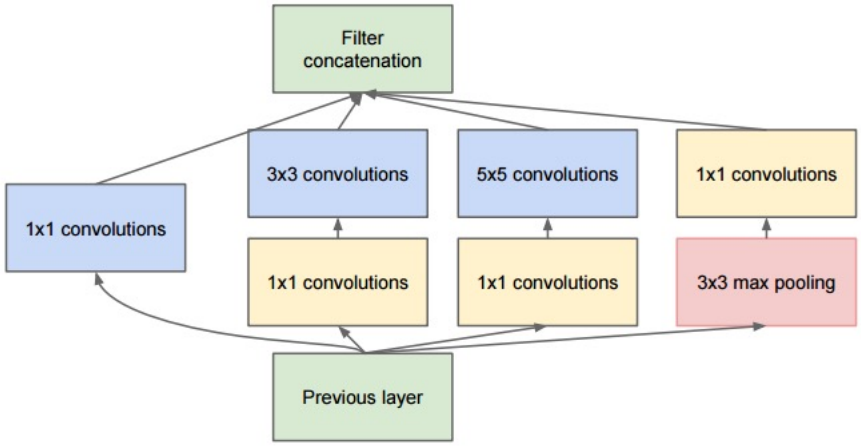
AlexNet first implemented successive use of convolutional layer rather than having a single conv layer always immediately follow by a pool layer. It also uses ReLU as non-linearities, dropout technique to avoid overfitting, and overlapping max pooling.

VGG-Net

VGG-Net is characterized by extremely homogeneous architecture that only performs 3x3 convolutions and 2x2 pooling from the beginning to the end. While it uses much smaller filter (3x3), it also uses max pooling like AlexNet. One downside of VGG-Net is that it uses large feature sizes in many layers and therefore computationally expensive.

GoogleNet

GoogleNet was the first to use the Inception Module:



The Inception Module is a combination of 1x1, 3x3, and 5x5 filters. Use of 1x1 convolutional blocks allowed it to reduce the number of features and therefore computational burden of deep networks. Unlike AlexNet or VGG-Net, it uses average pooling. Using average pooling instead of fully connected layer at the top allows further ellimination of parameters. It uses softmax classifier.

ResNet

ResNet was the first to use a network of more than hundred layers. Moreover, to prevent overfitting, it trains on residuals instead of on the raw function. While it also uses a pooling layer and softmax as final classifer, it does not have fully connected layers at the end.

2 Decision trees, entropy, and information gain

Part 1

· Show that $H(S) \leq 1$.

$$\begin{aligned} H(S) =& H(\frac{p}{p+n}) = -SlogS - (1-S)log(1-S) \\ \frac{dH(S)}{dS} =& (-logS + (-1)) - (-log(1-S) + 1) = 0 \\ =& -logS - 1 + log(1-S) + 1 = log\frac{1-S}{S} = 0 \\ 2^0 =& 1 = \frac{1-S}{S} \\ \therefore S =& 0.5 \end{aligned}$$

$H(S)$ reaches maximum when $S = 0.5$, in which case $H(S) = 1$. $\therefore H(S) \leq 1$.

· Show that $H(S) = 1$ when $p = n$.

When $p = n$, $\frac{p}{p+n} = \frac{1}{2}$.
$\therefore -\frac{1}{2}log\frac{1}{2} - \frac{1}{2}log\frac{1}{2} = 1$

Part 2

- Misclassfication rates
  A : (100 + 100) / 800 = $\frac{1}{4}$
  B : 200/800 = $\frac{1}{4}$

- Entropy gain model A

$$\begin{aligned} costD_{left} =& -\frac{3}{4}log\frac{3}{4} - \frac{1}{4}log\frac{1}{4} = 0.811 \\ costD_{right} =& -\frac{3}{4}log\frac{3}{4} - \frac{1}{4}log\frac{1}{4} = 0.811 \\ & \frac{1}{2} * 0.811 + \frac{1}{2} * 0.811 = 0.811 \end{aligned}$$

- Entropy gain model B

$$\begin{aligned} costD_{left} =& -\frac{2}{3}log\frac{2}{3} - \frac{1}{3}log\frac{1}{3} = 0.92 \\ costD_{right} =& 1 \\ & \frac{3}{4} * 0.92 = 0.69 \end{aligned}$$

The entropy after split is lower for the model B.

- Gini index model A

$$\begin{aligned} costD_{left} =& 2 * \frac{3}{4} * \frac{1}{4} = \frac{3}{8} \\ costD_{right} =& 2 * \frac{3}{4} * \frac{1}{4} = \frac{3}{8} \\ & \frac{1}{2} * \frac{3}{8} + \frac{1}{2} * \frac{3}{8} = \frac{3}{8} \end{aligned}$$

- Gini index model A

$$\begin{aligned} costD_{left} =& 2 * \frac{2}{3} * \frac{1}{3} = \frac{4}{9} \\ costD_{right} =& 0 \\ & \frac{3}{4} * \frac{4}{9} = \frac{1}{3} \end{aligned}$$

The Gini index of the splits is lower for the model B. A model with lower entropy gain and Gini index means that it's a better classifier.

3 Bagging

Part 1

Show that $E_{bag} = \frac{1}{L} E_{av}$.

$$\begin{aligned} E_{bag} =& E_X[\epsilon_{bag}(x)^2] \\ \epsilon_{bag}(x) =& \frac{1}{L} \sum l = 1L\epsilon_l(x) \\ \epsilon_{bag}(x)^2 =& (\frac{1}{L} \sum_{l=1}^{L} \epsilon_l(x))^2 \\ =& \frac{1}{L^2} \sum_{l=1}^{L} \epsilon_l(x)^2 + \sum_{l=1}^{L} \sum_{m=1}^{L} \epsilon_l(x)\epsilon_m(x) \qquad (l \neq m) \\ E_X[\epsilon_{bag}(x)^2] =& E_X[\frac{1}{L^2} \sum_{l=1}^{L} \epsilon_l(x)^2] + 0 \\ =& \frac{1}{L} E_{av} \end{aligned}$$

Part 2

Show that $E_{bag} \leq E_{av}$ without any assumptions.

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

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

Using Jansen's equality, if we set $\lambda = \frac{1}{L}$ and $x_l = \epsilon_l(x)$,

$$f(\sum_{l=1}^{L} \lambda_l x_l) = (\sum_{l=1}^{L} \frac{1}{L}\epsilon_l(x))^2$$

$$\leq \sum_{l=1}^{L} \lambda_l f(x_l) = \sum_{l=1}^{L} \frac{1}{L}\epsilon_l(x)^2$$

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

$$\therefore E_{bag} \leq E_{av}$$

References:

http://cs231n.github.io/neural-networks-1/
https://culurciello.github.io/tech/2016/06/04/nets.html
http://cs231n.github.io/convolutional-networks/case