Introduction to Machine Learing: Homework V

Due on Dec 20th, 2022 at 11:59pm

Professor Ziping Zhao

Bingnan Li
2020533092

1. [Deep Learning Models]

   (a) Consider a 3D convolution layer. Suppose the input size is $32 \times 32 \times 3$ (width, height, depth) and we use ten 5×5(width, height) kernels to convolve with it. Set stride = 1 and pad = 2. What is the output size? Let the bias for each kernel be a scalar, how many parameters do we have in this layer?

   Solution:

   The input data shape is

$$[C_{in}, D_{in}, H_{in}, W_{in}] = [1, 3, 32, 32]$$

   and the kernel size is

$$[C_{out}, D_{kernel}, H_{kernel}, w_{kernel}] = [10, 3, 5, 5]$$

   Then by the formula, we have output data with a shape

$$[C_{out}, D_{out}, H_{out}, W_{out}]$$

   and

$$\begin{aligned} D_{out} &= \frac{D_{in} + 2 \times pad - (D_{kernel} - 1) - 1}{stride} + 1 = 5 \\ H_{out} &= \frac{H_{in} + 2 \times pad - (H_{kernel} - 1) - 1}{stride} + 1 = 32 \\ W_{out} &= \frac{W_{in} + 2 \times pad - (W_{kernel} - 1) - 1}{stride} + 1 = 32 \end{aligned}$$

   Thus, the output size is $[C_{out}, D_{out}, H_{out}, W_{out}] = [10, 5, 32, 32]$

   Moreover, the total number of parameters is

$$\#\{parameters\} = \#\{parameters\ in\ kernel\} + \#\{biases\} = 5 \times 5 \times 3 \times 10 + 10 = 760$$

   (b) The convolution layer is followed by a max pooling layer with 2×2 (width, height) filter and stride = 2. What is the output size of the pooling layer? How many parameters do we have in the pooling layer?

   Solution:

   The input data shape is

$$[C_{in}, D_{in}, H_{in}, W_{in}] = [10, 5, 32, 32]$$

   and the kernel shape is

$$[C_{out}, D_{kernel}, H_{kernel}, W_{kernel}] = [10, 5, 2, 2]$$

   Then by the formula, we have output data with a shape

$$[C_{out}, D_{out}, H_{out}, W_{out}]$$

   and

$$\begin{aligned} D_{out} &= \frac{D_{in} + 2 \times pad - (D_{kernel} - 1) - 1}{stride} + 1 = 1 \\ H_{out} &= \frac{H_{in} + 2 \times pad - (H_{kernel} - 1) - 1}{stride} + 1 = 16 \\ W_{out} &= \frac{W_{in} + 2 \times pad - (W_{kernel} - 1) - 1}{stride} + 1 = 16 \end{aligned}$$

Thus, the output size is $[C_{out}, D_{out}, H_{out}, W_{out}] = [10, 1, 16, 16]$.

Moreover, given that the max pooling layer only performs maximizing, there are no parameters in the pooling layer. Thus, the total number of parameters of the pooling layer is 0.

2. [Deep Learning Models]

(a) (i.) When $r = 1$, $\boldsymbol{W}$ is exactly the eigenvector of $\boldsymbol{X}\boldsymbol{X}^T$ corresponding to its largest eigenvalue.

Proof:

When $r = 1$, we know that $\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{W}$ is a scalar, which means

$$Tr(\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{W}) = \boldsymbol{W}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{W}$$

Hence, the optimization problem can be rewritten as follows:

$$\underset{\boldsymbol{W}}{\text{maximize}} \quad \boldsymbol{W}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{W} \quad s.t.\ \boldsymbol{W}^T\boldsymbol{W} = 1$$

Then, the Lagrangian is:

$$\mathcal{L}(\boldsymbol{W}, \lambda) = \boldsymbol{W}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{W} - \lambda\boldsymbol{W}^T\boldsymbol{W}$$

Taking the derivate of $\mathcal{L}$ w.r.t $\boldsymbol{W}$ and setting it to 0, we get

$$2\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{W}^\star - 2\lambda^\star\boldsymbol{W}^\star = 0$$
$$\Rightarrow$$
$$\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{W}^\star = \lambda^\star\boldsymbol{W}^\star$$

Thus, $\lambda^\star$ is eigenvalue of $\boldsymbol{X}\boldsymbol{X}^T$ and $\boldsymbol{W}^\star$ is the corresponding eigenvector. Moreover, since

$$\boldsymbol{W}^{\star T}\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{W}^\star = \lambda^\star$$

Therefore, we need to get the largest eigenvalue $\lambda^\star$ to maximize $\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{W}$, which means $\boldsymbol{W}^\star$ is the eigenvector corresponding to the largest eigenvalue.

(ii.) $\boldsymbol{W}$ is also the solution to

$$\underset{\boldsymbol{W}}{\text{minimize}} \quad ||\boldsymbol{X} - \boldsymbol{W}\boldsymbol{W}^T\boldsymbol{X}||_F^2 \quad s.t. \quad \boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}_r$$

Proof:

Given that $|\boldsymbol{P}|_F = \sqrt{Tr(\boldsymbol{P}^T\boldsymbol{P})}$, we get

$$\begin{aligned}
||\boldsymbol{X} - \boldsymbol{W}\boldsymbol{W}^T\boldsymbol{X}||_F^2 &= Tr[(\boldsymbol{X} - \boldsymbol{W}\boldsymbol{W}^T\boldsymbol{X})^T(\boldsymbol{X} - \boldsymbol{W}\boldsymbol{W}^T\boldsymbol{X})] \\
&= Tr[\boldsymbol{X}^T\boldsymbol{X} - \boldsymbol{X}^T\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{X} - \boldsymbol{X}^T\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{X} + \boldsymbol{X}^T\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{X}] \\
&= Tr[\boldsymbol{X}^T\boldsymbol{X} - \boldsymbol{X}^T\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{X}] \\
&= Tr(\boldsymbol{X}^T\boldsymbol{X}) - Tr(\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{X}) \\
&= Tr(\boldsymbol{X}^T\boldsymbol{X}) - Tr(\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{W})
\end{aligned}$$

Since the first term of the equation has nothing to do with $\boldsymbol{W}$, the optimization problem can be rewritten as follows:

$$\underset{\boldsymbol{W}}{\text{minimize}} \quad -Tr(\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{W}) \quad s.t.\ \boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}_r$$

which is equivalent to the PCA problem.

(b) (i.) $\boldsymbol{A}_2^\star$ can be solved from:

$$\underset{\boldsymbol{A}_2}{\text{minimize}} \quad ||\boldsymbol{X} - \boldsymbol{A}_2\boldsymbol{A}_2^\dagger\boldsymbol{X}||_F^2$$

(ii.) The solution $\boldsymbol{W}$ from (a) can be taken as the same as $\boldsymbol{A}_2^\star$.

3. [Ensemble Learning] Suppose there are L independent two-class classifiers used for simple voting and the output of classifier j ($j = 1 \cdots L$) is denoted as d_j. From the point of view that the mean squared error of an estimator can be decomposed into the bias part and the variance part, explain why increasing L can lead to an increase in classification accuracy.

Proof:

By the definition of MSE, we get

$$\begin{aligned}
MSE(\hat{y}) &= \mathbb{E}[(\hat{y}-y)^2] = \mathbb{E}[(\hat{y}-\mathbb{E}[\hat{y}]+\mathbb{E}[\hat{y}]-y)^2] \\
&= \mathbb{E}[(\hat{y}-\mathbb{E}[\hat{y}])^2 + 2(\hat{y}-\mathbb{E}[\hat{y}])(\mathbb{E}[\hat{y}]-y) + (\mathbb{E}[\hat{y}]-y)^2] \\
&= \mathbb{E}[(\hat{y}-\mathbb{E}[\hat{y}])^2] + 2\mathbb{E}[(\hat{y}-\mathbb{E}[\hat{y}])]\mathbb{E}[(\mathbb{E}[\hat{y}]-y)] + \mathbb{E}[(\mathbb{E}[\hat{y}]-y)^2] \\
&= \mathbb{E}[(\hat{y}-\mathbb{E}[\hat{y}])^2] + 2(\mathbb{E}[\hat{y}]-\mathbb{E}[\hat{y}])\mathbb{E}[(\mathbb{E}[\hat{y}]-y)] + (\mathbb{E}[\hat{y}]-y)^2 \\
&= \mathbb{E}[(\hat{y}-\mathbb{E}[\hat{y}])^2] + (\mathbb{E}[\hat{y}]-y)^2 \\
&= \text{Var}(\hat{y}) + \text{Bias}^2(\hat{y},y)
\end{aligned}$$

Then, for the second term of the equation above, we get

$$\begin{aligned}
\text{Bias}^2(\hat{y},y) &\propto \text{Bias}(\hat{y},y) \\
&= \mathbb{E}[\hat{y}] - y \\
&\propto \mathbb{E}[\hat{y}] \\
&= \mathbb{E}\left[\frac{1}{L}\sum_j d_j\right] \\
&\geq \frac{1}{L} \times L \min_j\{\mathbb{E}[d_j]\} \\
&= \min_j\{\mathbb{E}[d_j]\}
\end{aligned}$$

which means that the Bias term won't change as L gets larger.

As for the first term, we get

$$\begin{aligned}
\text{Var}(\hat{y}) &= \text{Var}\left(\frac{1}{L}\sum_j d_j\right) \\
&= \frac{1}{L^2}\text{Var}\left(\sum_j d_j\right) \\
&\leq \frac{1}{L^2} \times L \max_j\left(\text{Var}(d_j)\right) \\
&= \frac{1}{L}\max_j\{\text{Var}(d_j)\}
\end{aligned}$$

which means that the Variance term will get smaller when L gets larger.

In conclusion, $MSE(\hat{y})$ will get smaller as L gets larger, so the classification will be more accurate.

4. [Model Assessment and Selection] Suppose we carry out a K-fold cross-validation on a dataset and obtain the classification error rates $\{p_i\}_{i=1}^K$, describe the steps of a one-sided t test on testing the null hypothesis H_0 that the classifier has error percentage p_0 or less at a significance level α.