Homework #2
Practical Issue

Computer Vision Practice with Deep Learning
NTU, Spring 23

Outline

- Problem
- Submission & Rules
- Helps

Problem 1 - Faster Convolution

- Convolution is a common operation in CNN, and the following equation is a general form, where **S(., .)** is a pre-defined similarity measure, **Y** is output features, **X** is input features and **W** is a convolution kernel. In normal convolution, the S(., .) is multiplication.

$$Y \in \mathbb{R}^{H \times W \times C_{out}},\ X \in \mathbb{R}^{H \times W \times C_{in}}$$

$$W \in \mathbb{R}^{k \times k \times C_{in} \times C_{out}}$$

$$Y[x, y, s] = \sum_{i=1}^{k} \sum_{j=1}^{k} \sum_{c=1}^{c_{in}} S(X[x+i, y+j, c], W[i, j, c, s])$$

Problem 1 - Faster Convolution

- Nevertheless, normal convolution requires numerous multiplication operations, which have a much higher computational complexity than addition operations.

$$Y \in \mathbb{R}^{H \times W \times C_{out}},\ X \in \mathbb{R}^{H \times W \times C_{in}}$$

$$W \in \mathbb{R}^{k \times k \times C_{in} \times C_{out}}$$

$$Y[x, y, s] = \sum_{i=1}^{k} \sum_{j=1}^{k} \sum_{c=1}^{C_{in}} S(X[x+i, y+j, c], W[i, j, c, s])$$

Problem 1 - Faster Convolution

1-1 (7%)

- **To create the "faster convolution", we adopt L1 distance as our similarity measure. Please provide the equation for the "faster convolution", which avoids the use of multiplication operations.**
- Note
  - The output obtained from the new equation must still adhere to the definition of similarity, where a higher value indicates a greater degree of similarity.
  - Subtraction can be viewed as a form of addition operation, while division can be seen as a type of multiplication operation.

Problem 1 - Faster Convolution

1-2 (7%)

- **Using the same notation as on the page 3, please calculate the number of addition and multiplication operations involved in both normal convolution and faster convolutio, and fill in the results in the following table.**

Convolution type	# of addition operation	# of multiplication operation
normal		
faster		

Problem 1 - Faster Convolution

- In deep learning, we use back-propagation to compute the gradients and stochastic gradient descent to update the network's parameters.
- The partial derivative of output feature **Y** with respective the kernel **W** is as followed.

$$\frac{\partial Y[x,y,s]}{\partial W[i,j,c,s]} = X[x+i, y+j, c]$$

Problem 1 - Faster Convolution

1-3 (7%)

- Please derive the partial derivative for faster convolution proposed in problem 1-1.

Problem 1 - Faster Convolution

1-4 (8%)

- **What issues could arise during model optimization when using the partial derivative proposed in problem 1-3? Due to these issues, we can modified the original partial derivative to the following equation:**

$$\frac{\partial Y[x,y,s]}{\partial W[i,j,c,s]} = X[x+i, y+j, c] - W[i, j, c, s]$$

Problem 1 - Faster Convolution

1-5 (9%)

- Current research suggests that when the variance of input features and output features are equal, it can have a positive impact on the network's performance.
- **Given the suggestion above, what is the value that the variance of w (normal convolution) should be initialized to in order to ensure that the variance of the output features matches that of the input features?**

Problem 1 - Faster Convolution

1-5 **(9%)**

- Note:
  - Assuming that the input features and kernel weights are independent and identically distributed following a normal distribution.
  - Using the same notation as on the page 3
- Hint: Given two random variable U, V, and T = U x V. If U and V are independent. The Var(T) = Var(U) x Var(V).

Problem 1 - Faster Convolution

- According to the table below, the L2-norm of the weight gradient in faster convolution is smaller than in normal convolution. This is because the variance of the output features obtained from faster convolution is much larger than that of normal convolution.

Convolution type	Layer1	Layer2	Layer3
Faster	0.0010	0.0013	0.0150
Normal	0.2271	0.3002	0.4660

Problem 1 - Faster Convolution

1-6 (9%)

- **Following the initialization procedure outlined in problem 1-5, why is the variance of the output features obtained from faster convolution much larger than those obtained from normal convolution at the first iteration?**

Problem 1 - Faster Convolution

1-7 (8%)

- **Please describe how to solve the aforementioned issue by adaptively adjusting the learning rate for each layer. Additionally, please provide a mathematical explanation for why this approach is effective.**
- Hint: Typically in a neural network, the number of channels increases as the depth of the layer increases.

Problem 2 - Pruning

1. (7%) Briefly describe the main differences and advantages among weight pruning , neuron pruning, and filter pruning.

2. (9%) How to perform filter pruning? Please **find an article or paper** that talks about a certain approach to perform filter pruning and describe its **methodology and how it compares to other methods**.

3. (7%) Prune the weight of nn.Linear(1024, 512) by 30% randomly, and compare the MACs(multiply–accumulate) and number of parameters before and after pruning.

Problem 3 - Before FasterNet

3-1 (8%)

- There are various metrics that can be used to measure the efficiency of a model, including MACs (multiply-accumulate), the number of parameters, and the number of memory accesses (read and write each count once).
- **Please use these three metrics to evaluate the performance of regular convolution, depthwise convolution, and pointwise convolution by filling out the table**.
- You may assume the shapes of input and output are both H*W*C, and the kernel size can be denoted by k if needed.

	MACs	# of parameters	# of memory accesses
regular conv			
DW conv			
PW conv			

Problem 3 - Before FasterNet

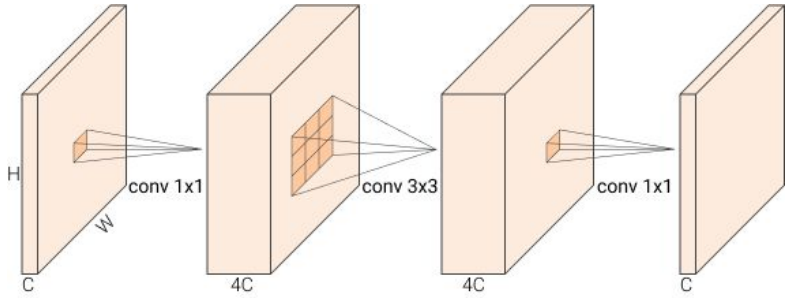
3-2 (7%)

- To improve the efficiency of regular convolution, a combination of depthwise and pointwise convolutions is sometimes used, known as depthwise separable convolution.
- Please **compare regular and depthwise separable convolutions based on the metrics** computed in the previous question. Additionally, please **explain the advantages** of using depthwise separable convolution.
- It is worth noting that for the purpose of this comparison, depthwise separable convolution can be treated as a fused operation, thus eliminating the redundant memory access between the depthwise and pointwise convolutions.

Problem 3 - Before FasterNet

3-3 (7%)

- Following the previous question, it has been shown that directly combining depthwise and pointwise convolutions without modifying the channel number of the architecture can lead to significant performance drops. In order to address this issue, the inverted bottleneck was proposed.
- Please compute the **three metrics** (MACs, number of parameters, and memory access) **based on the given architecture** using the inverted bottleneck approach. Then, please **briefly describe any potential drawbacks** that may arise from using this design.



Submission & Rules

- **Late policy:**

late (hour)	(0, 24]	(24, 48]	>48
deduction	60%	30%	0

- **Plagiarism**
  - Please note that the use of ChatGPT or any other AI composition software is strictly prohibited when completing handwritten homework assignments.
  - Plagiarism is a serious offense and will not be treated lightly.
- **Submission**
  - Deadline: **2023/4/18 (Tue.) 23:59 (GMT+8)**
  - Submit all results to **Gradescope** (Refer to the announcement in NTU COOL)

Helps

- Mail
  - If you have any questions, contact TAs via this email
    - ntu.cvpdl.ta@gmail.com
  - Please note that emails sent to TAs' personal email addresses will not receive a response.