Homework 3

(2D conv in matrix form) In the class, we expressed discrete 1D convolution as matrix-vector multiplication, and the matrix is Toeplitz. Here let us consider 2D convolution. We have a 3×3 image

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix},$$

and a 2×2 convolutional kernel/filter,

$$\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}.$$

Suppose we use "valid" padding, and stride equals 1, without any dialation.

1. (5') What is the 2D convolution's output?
2. (10') Express the 2D convolution as matrix-vector multiplication.

   (Hint: you are allowed to use "vectorize" and "reshape" operation. In specific, vectorizing a matrix simply stacking its columns to form a long vector. Reshaping can convert the long vector back to its matrix format.)
3. (5') Discuss the form of the matrix multiplied to the vector.

(Transfer learning) In the class, we talked about how to reuse models trained on big datasets, especially when your task does not have many training samples. Here we excercise on a task, categorizing 100 classes of sports images [1]. The dataset consists of 13493 training, 500 validation and 500 test images of size $224 \times 224 \times 3$.

1. (20') train your own AlexNet from scratch, without referring to any weights published. Report test accuracy.
2. (20') Reuse the weights from AlexNet [2], design a transfer learning strategy, and report test accuracy. Compare and discuss with the result you get for question 1.
3. (40') If you are allowed to use multiple existing models, e.g., vgg19 [3], resnet50 [4], and Inception[5]. Design a transfer learning strategy and report accuracy.

[1] data downloadable from kaggle page
[2] downloadable from pytorch link
[3] downloadable from tensorflow link, pytorch link
[4] downloadable from tensorflow link, pytorch link
[5] downloadable from tensorflow link, pytorch link