CAP 6619 Deep Learning

2024 Summer

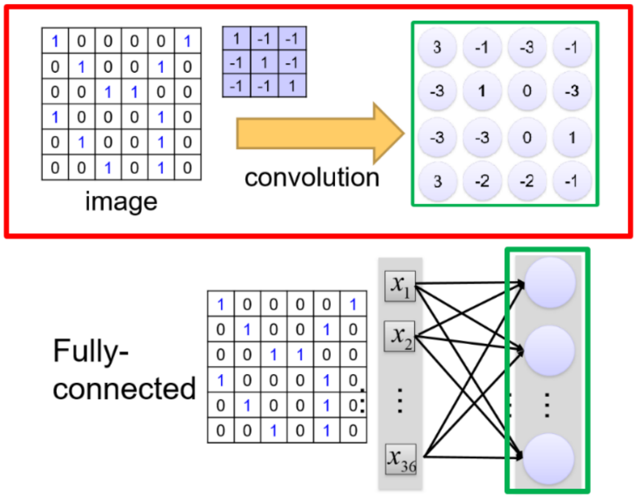
Homework 3 [18 Pts, Due: June 10 2024. Late Penalty: -2/day]

[If two homework submissions are found to be similar to each other, both submissions will receive 0 grade]

[Homework solutions must be submitted through Canvas. No email submission is accepted. If you have multiple files, please include all files as one zip file, and submit zip file online (only zip, pdf, or word files are allowed). You can always update your submissions. Only the latest version will be graded.]

Question 1 [2 pts]: In Figure 1, the upper panel shows a convolutional filter being applied to an image with 6x6 pixels to generate output, and the lower panel shows a fully connected dense network to process the same image.

- For the dense network, how many neurons are needed to produce the results which are the same as the convolutional filter?
- How many parameters are needed for the dense network?
- How many weight values does the convolutional filter have?
- Explain why the dense network is designed to find global pattern, whereas the convolutional filter is designed to find local patterns?



1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

1	-1	-1
-1	1	-1
-1	-1	1

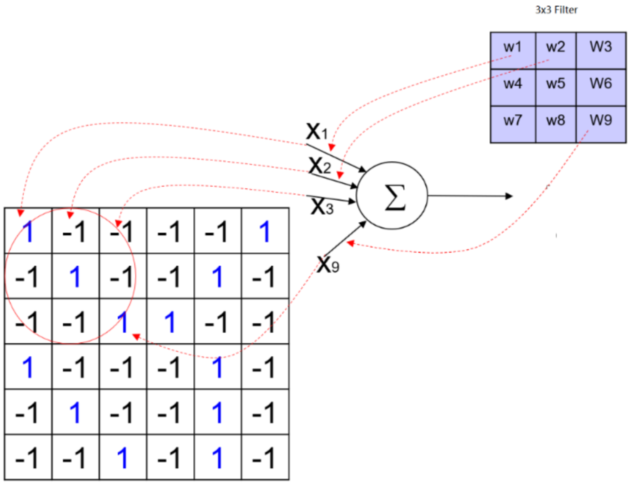
3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

Figure 1

Question 2 [2 pts]: Figure 2 shows a 6x6 image, and a 3x3 convolutional filter, and the weight symbols specified in the matrix (w1, w2, …),

- please show convolutional filter output after the filter being applied to the red colored circle (using weight values w1,…, w9. w0 denotes bias) [0.5 pt]
- what is the purpose of the weight values (w1, w2, …) of the 3x3 filter? [0.5 pt]
- When moving the filter across the image (using stride 1), what is the percentage of input shared been two consecutive points (horizontally or vertically) [0.5 pt]
- What is the size of the feature map after applying the filter to the whole image (using stride 1)? What are the factors determining the feature map size [0.5 pt]



w1	w2	W3
w4	w5	W6
w7	w8	W9

1	-1	-1	-1	-1	1
-1	1	-1	-1	1	-1
-1	-1	1	1	-1	-1
1	-1	-1	-1	1	-1
-1	1	-1	-1	1	-1
-1	-1	1	-1	1	-1

6 x 6 image

Figure 2

Question 3 [2 pts]: Table 1 shows a 3x6 synthetic image, and Table 2 shows a filter.

- Please apply the filter in Table 2 to the image in Table 1 (using convolutional filter and stride=1), and show the feature map output. [0.5 pt]
- Show perceptron architecture to implement the filter (ignore the bias) [0.5 pt]
- Apply 2x2 max pooling to the above result, and report the resulting image. [0.5 pt]
- Explain the role of the filter in Table 2, and how does the filter achieve the goal through the convolution process. [0.5 pt]

Table 1

4	6	4	0	4	5
4	-5	5	1	6	3
1	0	1	1	0	1

Table 2

-1	-1	-1
1	1	1

Question 4 [2 pts]: Figure 3 shows the structure of the LeNet5 convolutional neural network.

- What is the name of the C1 layer? What is the size of the filters of the C1 layer? How many weight values C1 layer has? [0.5 pt]
- What is the size of the filters in the C3 layer? How many weight values C3 layer has? [0.5 pt]
- How many weight values C5 and F6 layers each has, respectively [0.5 pt]

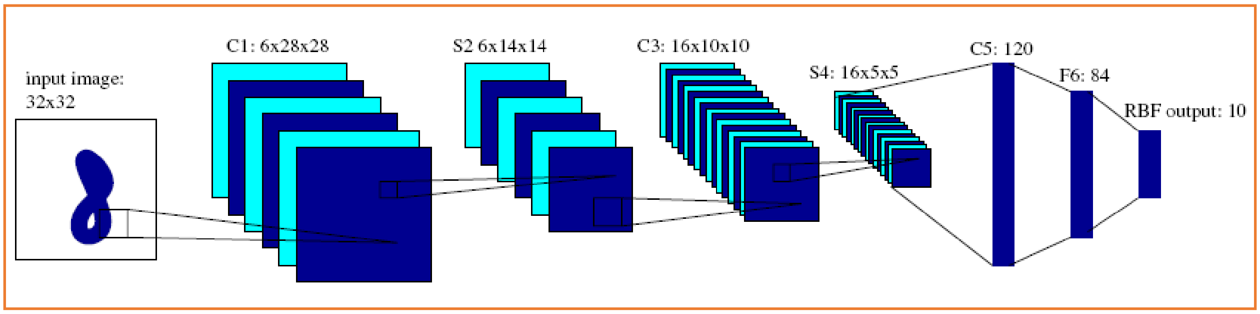
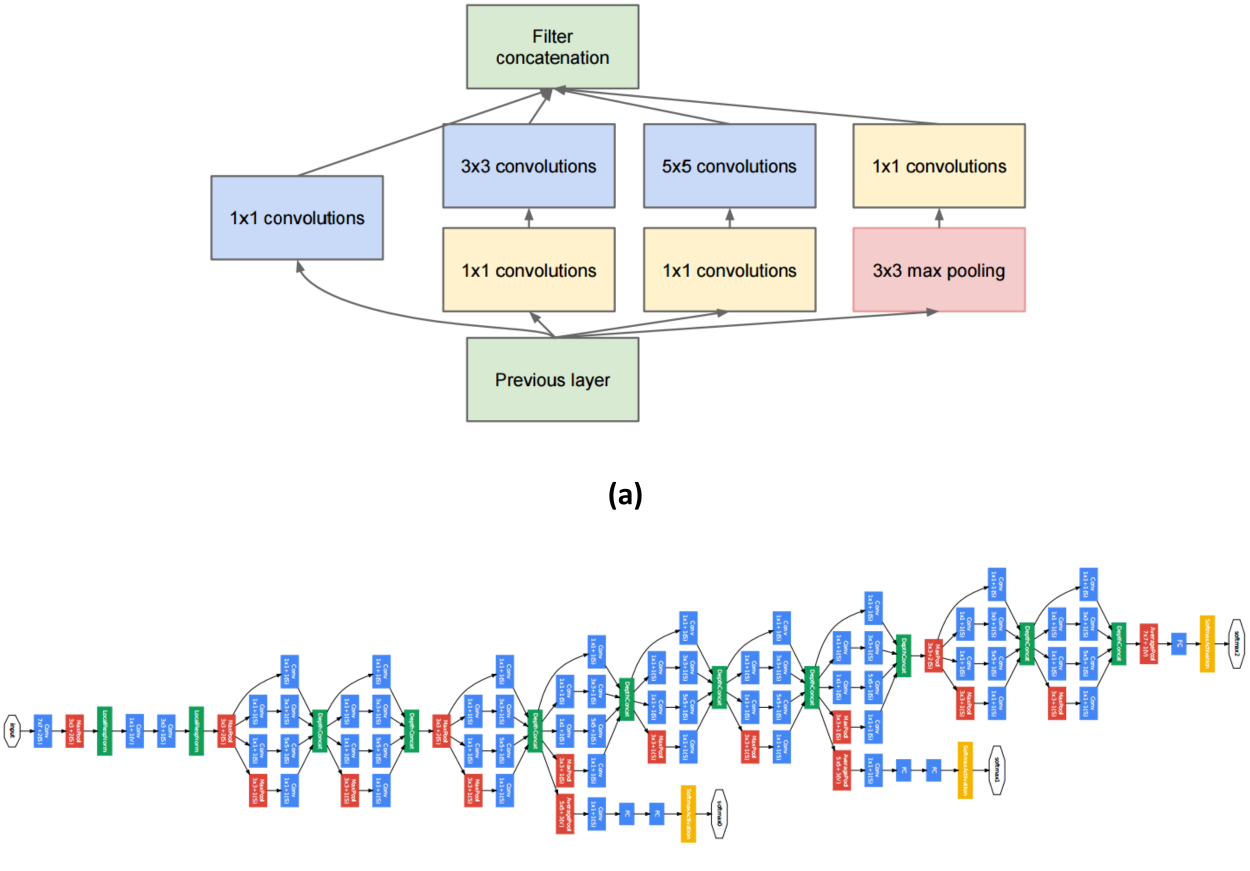


Figure 3

Question 5 [2 pts]: The following Keras codes show a deep learning network. Please draw the network structure from input to the output to explicitly show network components and parameters:

1. Please draw diagram of the designed network [0.5 pt]
2. Show input, output sizes, and number of weight values of each convolution layer [0.5 pt].
3. Show input and output size of each pooling layer [0.5 pt].
4. Show input, output sizes, and number of weight values of each dense layer [0.5 pt].

```
network = Sequential()
model.add(Conv2D(32,(3,3),activation="relu",input_shape=c(28,28,3))
model.add(MaxPooling2D((2,2))
model.add(Conv2D(64,(3,3),activation="relu"))
model.add(MaxPooling2D((2,2))
model.add(Flatten())
model.add(Dense(64,activation='relu'))
model.add(Dense(10,activation='softmax'))
```



(a)

(b)

Figure 4: Inception module (https://arxiv.org/pdf/1409.4842.pdf), and GoogLeNet structure

Question 6 [4 pts]: Figure 4(a) shows structure of an inception module, and Figure 4(b) shows GoogLeNet structure.

- How many layers GoogLeNet have? What are the main differences between GoogleNet vs. VGG-16 in the context of convolutional neural network? [0.5 pt]
- Explain motivation of the inception module [0.25 pt].

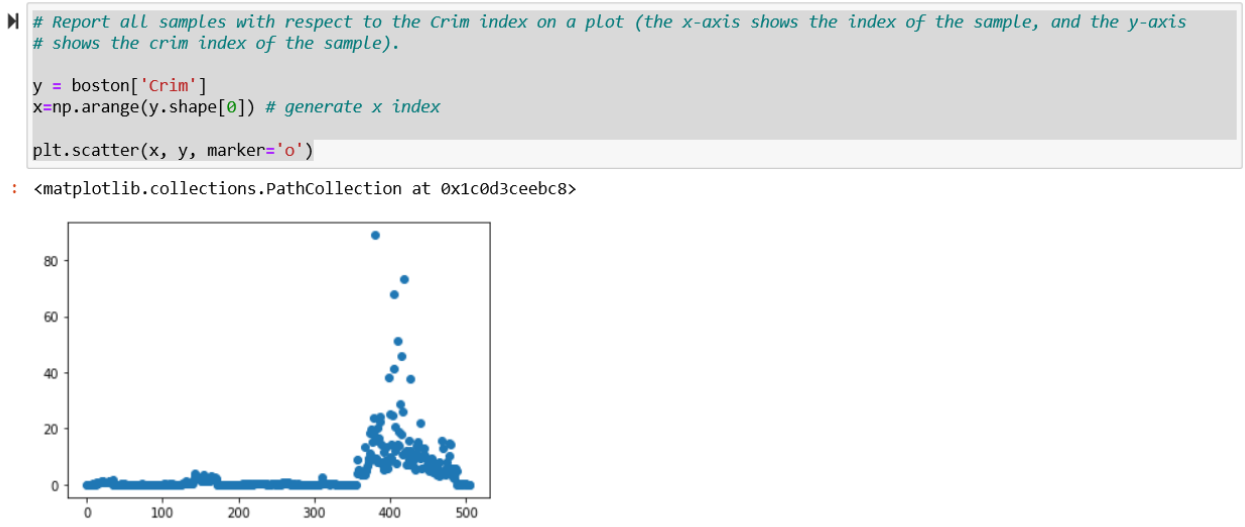
- Explain purpose and functionality of the 1x1 convolutions of the inception module [0.25 pt].
- The input image sizes to 224x224x3 (three color challenges). The first convolution layer of the network has 64 filters (each of which is 7x7 in size), the second volution layer has 64 1x1 filters, followed by 192 convolution filters (each with size 3x3). What are the number of tunable parameters at second and third convolution layer, respectively? [0.5 pt]
- Figure 4(a) shows inception module layer 3(a), the number of feature maps from previously layer is 28x28x192 (i.e., 192 feature maps), calculate number tunable parameters for each path of the inception module [0.5x4=2pts]
- Explain purpose and functionality of the 1x1 convolutions of the inception module [0.25 pt].
- The network has three softwmax (and output) at different stage of the network. What are the purposes of these outputs [0.5 pt].

Questions 7 is a programming task

For all programming tasks, solutions must be submitted as notebook (html or pdf) files for grading (your submission must include scrips/code and the running results of the script).

If you are not familiar with Python programming (and want to use Python for the coding tasks), please check Python Plotting notebook and Python Simple Analysis notebook posted in the Canvas, before working the coding tasks.

For each subtask, please use task description (requirement) as comments, and report your coding and results in following format:



```
# Report all samples with respect to the Crim index on a plot (the x-axis shows the index of the sample, and the y-axis
# shows the crim index of the sample).

y = boston['Crim']
x=np.arange(y.shape[0]) # generate x index

plt.scatter(x, y, marker='o')
```

```
: <matplotlib.collections.PathCollection at 0x1c0d3ceebc8>
```

Question 7 [4 pts]: Please follow "**CNN Image Classification [html, Notebook]** " to create a convolutional neural network (CNN) image classifiers and validate its performances.

- Download the cat vs. dog images (556MB zip file) from the following URL

https://www.cse.fau.edu/~xqzhu/courses/cap6619/dataset/train.zip
You can also download images from the Kaggle site (https://www.kaggle.com/c/dogs-vs-cats/data)

- Unzip the downloaded (zip) file. There are 25,000 (dog and cat images, 12,500 for each category) in the "train" folder.
- Create a training dataset with least 2000 images (1000 for each category), a validation set with at least 1000 images (500 for each category), and a test set with at least 1000 images (500 for each category) [1 pt]
- Create a CNN classifier with at least three convolution layers, two pooling layers, and two dense layers. Train the network on the training set, and report the performance of the classifier on the test set. [1 pt]
- For the same network structure created above, please add a dropout layer (with a selected dropout rate) and add Batch Normalization. Train the network on the training set, and report the performance of the classifier on the test set. [1 pt]
- For the above network (including a mixed use of dropout layer and Batch Normalization), please use image_data_generator with rotation, width_shift, height_shift, shear, and zoom to create image distortions for training. Train the network on the training set, and report the performance of the classifier on the test set. [1 pt]