Benjamin Luo (6/9/2024)

Problem 1.

Solution.

- 16 neurons are needed to produce the results of the convolutional filter, which in this case is a 4x4 feature map.
- $36 \times 16 + 16 = 594$ parameters needed for the dense layer neural network.
- 10 weight values are needed for the convolutional filter; 9 for the each dimension of the filter, and 1 taking in account for bias.
- Dense networks are designed to find global patterns due to the output being a classification based on every pixel in the image as individual input. On the other hand, the convolutional filter intends to learn and find local patterns, by scanning through the image with convolutional filters, which can be multi-selections in some cases, which is thus designed to train more specific features of an image.

□

Problem 2.

Solution.

$$v = w_1x_1 + w_2x_2 + \cdots + w_9x_9$$

$$v = 1(1) + (-1)(-1) + (-1)(-1) + (-1)(-1) + 1(1) + (-1)(-1) + (-1)(-1) + (-1)(-1) + 1(1)$$

$$v = 9$$

- The purpose of the weight values in the 3x3 convolutional filter is to look through the image to see if there are patterns like the filter that are present in the image.
- When moving the filter across the image, 66% of the image is preserved between any two movements across one stride (assuming stride = 1).
- The size of the feature map after applying the filter to the whole image is 4x4. The factors that affect the feature map size is the size of the convolutional filter (i.e. 1x1, 3x3, 5x5, etc) as well as stride.

□

Problem 3.

Solution.

Resulting feature map output:

-10	-9	4	1
-2	1	-10	-8

-10	-9	4	1	$\Rightarrow$ Result of the 2x2 max pooling $\Rightarrow$	1	4
-2	1	-10	-8			

- The role of the convolutional filter in Table 2 is to identify a specific pattern, a presence of a horizontal line pattern in the image. It achieves this by using a filter to run through the image in order to see what sections of the image respond most strongly to the filter's parameters.
- The 2x2 max pooling layer is then used to reduce the noise and train the relevant neurons towards the pattern outlined by the convolutional filter.

□

Problem 4.

Solution.

- The name of the C1 layer is Convolutional Layer 1; The size of the C1 layer filters is 5x5. Convolutional Layer 1 has $((5 \times 5 \times 1) + 1) \times 6 = 156$ weight values.
- C3 layer's filter size is also 5x5. C3 layer has $((5 \cdot 5 \cdot 6) + 1) \cdot 16 = 2,416$ weight values.
- C5 and F6 have 2,040 and 10,164 weight values, respectively.
  $((1 \cdot 1 \cdot 16) + 1) \cdot 120 = 2,040$
  $(120 \cdot 84) + 84 = 10,164$

□

Problem 5.

Solution. (image of the designed network) □

Problem 6.

Solution.

- GoogleNet has 22 layers; The main difference between GoogleNet and VGG-16 is how the networks are structured. The VGG-16 use a conventional approach, using a combination of convolutional, pooling, and dense layers in a single path. GoogleNet is different from VGG-16 in that GoogleNet has multiple network paths running parallel, using inception networks.
- The motivation of the inception modules lies in the search of being able to capture a more sparse variety of patterns with a deep learning neural network. Inception modules enable this by allowing the network to have multiple parallel running pipelines.
- The purpose of the 1x1 convolutional layers in the inception module is to be able to reduce dimensionality before running the convolutional layers. It has been shown that doing this can drastically reduce the amount of weight parameters if the number of feature arrays can be reduced, which is the purpose of the 1x1 convolutional filters.

- The 2nd convolutional layer has $((1 \cdot 1 \cdot 64) + 1) \cdot 64 = 4,160$ tunable parameters.
- The 3rd convolutional layer has $((3 \cdot 3 \cdot 64) + 1) \cdot 192 = 110,784$ tunable parameters.
- The inception module layer 3(a) is as listed:
  `First path:`
  1×1 conv $\Rightarrow ((7 \cdot 7 \cdot 192) + 1) \cdot 64 = 9409 \cdot 64 = 602,176$
  First path has 602,176 tunable parameters.

  `Second path:`
  1×1 conv $\Rightarrow ((1 \cdot 1 \cdot 192) + 1) \cdot 96 = 193 \cdot 96 = 18,528$
  3×3 conv $\Rightarrow ((3 \cdot 3 \cdot 96) + 1) \cdot 128 = 865 \cdot 128 = 110,720$
  Second path has a total of 129,248 tunable parameters.

  `Third path:`
  1×1 conv $\Rightarrow ((1 \cdot 1 \cdot 192) + 1) \cdot 16 = 193 \cdot 16 = 3,088$
  5×5 conv $\Rightarrow ((5 \cdot 5 \cdot 16) + 1) \cdot 32 = 401 \cdot 32 = 12,832$
  Third path has a total of 13,872 tunable parameters.

  `Fourth path:`
  3×3 maxpool $\Rightarrow 0$
  1×1 conv $\Rightarrow ((1 \cdot 1 \cdot 32) + 1) \cdot 32) = 1,056$
  Fourth path as a total of 1,056 tunable parameters.

- The purpose of the three softwmax outputs at different points in the network is to be able to analyze and assess network result accuracy as we progress through the network; if the classification improve from results earlier in the network compared to results later in the network, we can conclude that the network is improving in accuracy of the classification model.

□

Problem 7.

Solution. See rest of pdf for notebook html output. □