CS512 - Assignment 3

Mohammadreza Asherloo
Department of Mechanical, Materials and Aerospace Engineering
Illinois Institute of Technology

November 10, 2020

Question 1.a:

In template matching interpretation, we basically compare the input with the units that are templates by computing the dot product of the input with each template. If they are perpendicular, result would be zero which means they're not similar at all. In decision boundary interpretation, each node is considered as a decision boundary that divides the input in two parts and classifies the data between these two parts (basically divides one part from other data). We will have as many decision boundaries as nodes that classifies the data.

Question 1.b:

For converting similarity scores to probabilities, we will use activation functions such as sigmoid or softmax. For a 2-class classification we will use sigmoid which is:

$sigmoid(x) = \frac{1}{1+e^{-}x}$

For a k-class classification we will use softmax which is:

$\hat{y}_j^{(i)} = \frac{exp(S_j^{(i)})}{\sum_{l=1}^{n} exp(S_l^{(i)})}$

Question 1.c:

L1:
$L_i(\theta) = \sum_{l=1}^{k} |\hat{y}_j^{(i)} - y_j^{(i)}|$

L2:
$L_i(\theta) = \sum_{l=1}^{k} (\hat{y}_j^{(i)} - y_j^{(i)})^2$

Huber loss:

$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{for} |y - f(x)| \leq \delta, \\ \delta \left| y - f(x) \right| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$$

Cross entropy loss (log likelihood):
$L(\theta) = -\sum_{i=1}^{m} \sum_{j=1}^{n} y_j^{(i)} log(\hat{y}_j^{(i)})$

Question 1.d:

Regularization basically ensures that the wights get smaller values as possible because this way model will generalize better on data other than the training dataset. Regularization does this by punishing the model when the weights are getting bigger.

Question 1.e:

We will go in the opposite direction of the gradient because if we go in the same direction, means that we are amplifying the trend and moving away from the minimum of the function. We go backward to find the minimum until the gradient is close to zero.

Question 1.f:

Difference is that in gradient descent, a batch of points are considered and the gradient of them is computed and the average of these gradients are used to update the parameters but in stochastic gradient descent a point is selected randomly and its gradient is used to update the point itself.

Question 1.g:

Learning rate doesn't need to be fixed through the training process. We can use different choices of strategies to make the learning rate after each pass in order to prevent the model from overshooting in around the minimum. We can use step decay, exponential decay or fraction decay strategies to do this task.

Question 1.h:

We are using momentum in gradient descent to use previous results in order to smooth out the update. With updating using normal gradient descent, we will have two direc-

tions (vertical and horizontal) toward the minimum. By using momentum (or exponentially weighted average) we can bring the main direction toward the horizontal direction and have much more straight path toward local optima.

Question 1.i:

Through the forward pass we use the weights in each layer to compute the main output. After this computation, we will use this output to compute the gradient of each layer and node. Then we will use these gradients to update the weights through backward pass. Gradients of each node will be propagated through the backward pass.

Question 1.j:

Fully connected layer takes the image and flattens it to make a single vector out of the image. This vector will be the input for the next layer. But the in convolution layer a filter will be passed through the image and scans few number of pixels in each step and produces a feature map that will be used for predicting the class of each feature.

Question 1.k:

Dropout is a type of regularization to prevent large neural nets from overfitting on the training dataset and losing their generalization ability. By using dropout, a fraction of nodes will be shut down during each pass. In other words, a number of neural nets with different architectures will be trained on the dataset in a parallel manner to update the weights and create the final model.

Question 2.a:

Image:

$$\text{R} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \text{G} = \begin{bmatrix} 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \end{bmatrix}, \text{B} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{bmatrix}$$

$$\text{Filter: } \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\text{Convolution} = \begin{bmatrix} 9+18+18 & 9+18+18 \\ 9+18+27 & 9+18+27 \end{bmatrix} = \begin{bmatrix} 45 & 45 \\ 54 & 54 \end{bmatrix}$$

Question 2.b:

Image:

$$\mathrm{R} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathrm{G} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 2 & 2 & 2 & 0 \\ 0 & 2 & 2 & 2 & 2 & 0 \\ 0 & 2 & 2 & 2 & 2 & 0 \\ 0 & 2 & 2 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathrm{B} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 2 & 2 & 2 & 2 & 0 \\ 0 & 3 & 3 & 3 & 3 & 0 \\ 0 & 4 & 4 & 4 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\text{Filter: } \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\text{Convolution} = \begin{bmatrix} 4+8+6 & 6+12+9 & 6+12+9 & 4+8+6 \\ 6+12+12 & 9+18+18 & 9+18+18 & 6+12+12 \\ 6+12+18 & 9+18+27 & 9+18+27 & 6+12+18 \\ 4+8+14 & 6+12+21 & 6+12+21 & 4+8+14 \end{bmatrix} = \begin{bmatrix} 18 & 27 & 27 & 18 \\ 30 & 45 & 45 & 30 \\ 36 & 54 & 54 & 36 \\ 26 & 39 & 39 & 26 \end{bmatrix}$$

Question 2.c:

Image:

$$\mathrm{R} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathrm{G} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 2 & 2 & 2 & 0 \\ 0 & 2 & 2 & 2 & 2 & 0 \\ 0 & 2 & 2 & 2 & 2 & 0 \\ 0 & 2 & 2 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathrm{B} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 2 & 2 & 2 & 2 & 0 \\ 0 & 3 & 3 & 3 & 3 & 0 \\ 0 & 4 & 4 & 4 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\text{Filter: } \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

$$\text{Convolution} = \begin{bmatrix} 4+8+12 & 4+8+12 \\ 4+8+8 & 4+8+8 \end{bmatrix} = \begin{bmatrix} 24 & 24 \\ 20 & 20 \end{bmatrix}$$

Question 2.d:

By using convolution we are computing the weighted sum of a group of pixel with size of the filter. By applying the filter on this group of filters we basically are looking for the

similar features to the that of filter is showing. For example, when we use the edge detection kernel, we can see the edge in the kernel itself. So we are looking for the same edges in the image as the edge in the kernel.

Question 2.e:

We can do subsampling and smoothing on an image in different levels to create a pyramid with that image which is basically a series of decreasing resolution images and a series of decreasing resolution details. By using this technique we can better find the features in an image by using even a small kernel.

Question 2.f:

As spatial dimensions decrease, number of depth channels increase to compensate for reduced coefficients or in other words, keep the number of coefficients the same.

Question 2.g:

formula $= \frac{W-K+2P}{S} + 1 = \frac{128-3+0}{1} + 1 = 126 \Longrightarrow$ output size $= 126 \times 126 \times 16$

Question 2.h:

formula $= \frac{W-K+2P}{S} + 1 = \frac{128-3+0}{2} + 1 = 63.5 \Longrightarrow$ output size $= 63 \times 63 \times 16$

Question 2.i:

Convolution of multi channel images is the summation of convolution of each channel. If we use a 1×1 filter, we are basically summing up the value of each pixel in different channels so number of channels will be reduced. We can control the number of channels in output by number of filters in each layer.

Question 2.j:

Each node in each convolution layer looks for a specific feature. For example in first layer we can look for edges with different orientations and when we go deeper in the network we can look for more complex features such as basket weaves or some complex shapes such as cat's nose or ears.

Question 2.k:

Pooling layer will downsample the image with different methods such as taking the average of a group of pixels and put the result in one pixel in the output. Methods can be max pooling, average pooling and etc. We can reduce the spatial dimension of image with pooling layer.

Question 2.l:

Output Image:

$$\mathrm{R} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \mathrm{G} = \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}, \mathrm{B} = \begin{bmatrix} 2 & 2 \\ 4 & 4 \end{bmatrix}$$