CS 583-WN: Deep Learning
Quiz 2

Nicolas Jorquera

Problem 1

Given the input tensor, $T \in \mathbb{R}^{a_1 \times a_2 \times a_3}$, a filter, $F \in \mathbb{R}^{b_1 \times b_2 \times b_3}$, stride $= n$, no zero-padding, determine the dimensions (c_1, c_2) of the convolution, $C \in \mathbb{R}^{c_1 \times c_2}$. The dimensions $a_1, a_2, ..., c_2$ are all scalars.

The convolution of an input tensor with a filter results in a spatial reduction based on the size of the filter and the stride. The general formula for each spatial dimension after convolution, without zero-padding, is:

$$c_i = \left\lfloor \frac{a_i - b_i}{n} \right\rfloor + 1$$

where c_i is the resulting dimension, a_i is the input dimension, b_i is the filter dimension, and n is the stride.

Solution: Using the formula above, the spatial dimensions of the convolution result are:

$$c_1 = \left\lfloor \frac{a_1 - b_1}{n} \right\rfloor + 1$$

$$c_2 = \left\lfloor \frac{a_2 - b_2}{n} \right\rfloor + 1$$

Problem 2

Given an algorithm executed with n_1 GPUs taking t_1 minutes per iteration and the same algorithm executed with n_2 GPUs taking t_2 minutes per iteration, determine the speed-up ratio.

Speed-up measures the performance improvement an algorithm experiences when its resources are increased. It's calculated by the ratio of old performance to new performance.

Solution: The speed-up ratio S is given by:

$$S = \frac{n_1 \times t_1}{n_2 \times t_2}$$

Problem 3

Determine the optimal libraries for the operations: PCA, Matrix Algebra/Calculations, SVD, Eigenvalue Decomposition from the given libraries: Level 1 BLAS, Level 2 BLAS, Level 3 BLAS, LAPACK.

BLAS (Basic Linear Algebra Subprograms) provides levels of operations, from vector operations (Level 1) to matrix-matrix operations (Level 3). LAPACK (Linear Algebra PACKage) is built on top of BLAS and provides routines for advanced linear algebra operations.

Solution:

- PCA - LAPACK (since PCA often involves eigenvalue decomposition)
- Matrix Algebra/Calculations - Level 3 BLAS (for matrix-matrix operations)
- SVD - LAPACK (it directly provides routines for Singular Value Decomposition)
- Eigenvalue Decomposition - LAPACK (it's designed for such advanced operations)

Problem 4

Determine the best choice of activation function for the given classification problems: Multi-class and Binary.

Activation functions introduce non-linearity into the network and dictate the output of a neuron. The choice of activation function often depends on the nature of the problem and the desired output range.

Solution:

- Multi-class - Softmax (It provides a probability distribution across multiple classes)
- Binary - Sigmoid (It outputs a value between 0 and 1, ideal for binary classification)

Problem 5

Given the neural network model:

```
model = models.Sequential()
model.add(Conv2D(20, (5, 5), input_shape=(30, 30, 3)))
model.add(MaxPooling2D(pool_size=(2, 2)))
```

Determine the parameters in the Convolutional layer, the MaxPooling2D layer, and their output shapes/sizes.

In a Conv2D layer, the number of parameters is calculated by (filter width$\times$filter height$\times$input channels$+$ 1) $\times$ number of filters. The "+1" accounts for the bias term for each filter.

Solution:

- Conv2D Layer: Parameters $= (5 \times 5 \times 3 + 1) \times 20 = 1520$. Output shape: $26 \times 26 \times 20$.
- MaxPooling2D Layer: Parameters $= 0$. Output shape: $13 \times 13 \times 20$.

Visualization -

Conv2D Layer - The input shape is 30×30×3 representing an image of size 30 by 30 with 3 color channels (RGB). The layer uses 20 filters of size 5×5. Thus, due to no padding and a default stride of 1, the output shape reduces by 4 units on both width and height, resulting in an output shape of 26×26×20.

MaxPooling2D Layer - With a pooling size of 2×2, it downsamples the spatial dimensions by half, yielding an output shape of 13×13×20.

Problem 6

Given the neural network model:

```
model = models.Sequential()
model.add(Flatten(input_shape=(10, 10, 3)))
model.add(Dense(100, activation='relu'))
model.add(Dense(10, activation='softmax'))
```

Determine the parameters in the Convolutional layer, the Dense layers, and their output shapes/sizes.

In a Dense (or Fully Connected) layer, the number of parameters is (input units$+1$)$\times$output units. The "+1" accounts for the bias term for each output unit. The Flatten layer reshapes the input but doesn't have trainable parameters.

Solution:

- Flatten Layer: Parameters $= 0$. Output shape: 300.
- First Dense Layer: Parameters $= (300 + 1) \times 100 = 30100$. Output shape: 100.
- Second Dense Layer: Parameters $= (100 + 1) \times 10 = 1010$. Output shape: 10.

Visualization -
Flatten Layer - The input tensor of $10\times10\times3$ is reshaped into a 1D tensor with 300 elements.
Dense Layer (ReLU Activation) - The 300 nodes from the Flatten layer connect to 100 neurons, making the output shape as 100.
Dense Layer (Softmax Activation) - The 100 nodes from the previous Dense layer connect to 10 neurons.