CS5783: Machine Learning
Assignment 3

Prof. Christopher Crick

1 The convolution process

In your Google Colab Jupyter notebook, answer the following in a text cell. Consider the below matrices as input (X) and convolutional kernel f. The input depth is 1.

$$X = \begin{bmatrix} 7 & 5 & 0 & 0 & 3 & 2 \\ 6 & 4 & 5 & 1 & 4 & 8 \\ 9 & 0 & 2 & 2 & 5 & 4 \\ 6 & 3 & 4 & 7 & 9 & 8 \\ 5 & 7 & 5 & 6 & 9 & 0 \\ 7 & 9 & 0 & 8 & 2 & 3 \end{bmatrix} \quad f = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

Compute the following:

1. What are the dimensions of the input and the kernel (or filter)? How many parameters are there in the kernel f?

2. What is the output activation map when you apply the convolutional operation using the filter f on the input X without padding?

3. What is the output when you apply a max-pooling operation on the output from the previous question?

Note: For parts 2 and 3 in question 3, please provide the actual output. You can work it out by hand or include a Python code cell to do it.

2 Convolution on MNIST

Design a convolutional neural network using Tensorflow and Keras of at least 10 convolutional layers. Use the MNIST dataset for evaluation (you may use the loaders in `tf.keras.datasets` to obtain the data). You must try three designs as detailed below and provide your observations (in Jupyter text cells in between the code cells) on the performance of each:

1. A regular CNN where the number of fliters in each layer increases as the depth of the network grows, i.e. the L^{th} layer will have more filters than the $(L-1)^{th}$ layer.

2. An inverted CNN where the number of filters in each layer *decreases* as the depth of the network grows, i.e. the L^{th} layer will have fewer filters than the $(L-1)^{th}$ layer.

3. An hourglass-shaped CNN where the number of filters increases until the L^{th} layer and decreases afterwards.

Your goal is to design these networks and optimize them to their best performance by choosing the right hyperparameters *for each network*, such as the learning rate, batch size and the choice of optimizer ('SGD', 'adam', 'RMSProp'). You must provide a detailed report of what values you tried for each hyperparameter, your observations on why the network performed well (or not), and the final accuracy for each network on the MNIST dataset.

You can refer to the Keras documentation at keras.io. In particular, an example MNIST CNN is shown at https://keras.io/examples/vision/mnist_convnet/ – this assignment asks you to do more than that simple system.

3 Convolution on CIFAR

Implement the LeNet CNN using Keras. It is a seven-layer network with three convolutional layers, two max-pooling layers and two dense layers. The structure is as follows:

1. Convolution layer with 6 5x5 kernels with stride 1
2. Max-pooling layer with 2x2 kernels with stride 2
3. Convolution layer with 16 convolution kernels of 5x5 with stride 1
4. Max-pooling layer with 2x2 kernels with stride 2
5. Convolution layer with 120 convolution kernels of 5x5
6. Dense layer with 84 neurons
7. Output layer

Use the 'Adam' optimizer to train your network on the CIFAR-10 dataset (another one that is included in `tf.keras.datasets`) for a fixed set of 25 epochs. Each image is a 32x32x3 matrix, and there are 60,000 training and 10,000 test images, with ten classes.

Perform the following analyses and answer each question briefly (3-5 sentences) in a Jupyter text block. Use plots and figures as necessary.

1. What is the effect of learning rate on the training process? Which performed best?
2. What is the effect of batch size on the training process? Which performed best?
3. Try different hyperparameters to obtain the best accuracy on the test set. What is your best performance and what were the hyperparameters?
4. Implement an equivalent dense feed forward network for the same task which each hidden layer containing the same number of neurons as the number of filters in each convolution layer. Use the 'Adam' optimizer to train your network on the CIFAR-10 dataset for a fixed set of 25 epochs. Compare its performance with your LeNet implementation based on the following questions:

- What is its performance?
- How many parameters are there in this network compared to the LeNet implementation? Are they worth it?

4 Turning in

Submit the link to your Google Colab notebook to Canvas. This assignment is due Monday, October 28.