Assignment 6

Apetria George

November 2023

1 Handmade implementatiton for conv2d layer

For this task, the explanations are represented by the detailed comments provided in the loaded python file - "HandmadeConv2d".

2 CIFAR-10 classification

The training images consist of 50,000 32x32 color images, 5,000 for each class. The class to which each image belongs is represented by an integer from the set {0, noOfClasses - 1}. For each image, this integer is called the label. In the present case, there are 10 classes (airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks) represented by the labels 0, 1, 2, ..., 9. Therefore, the data training are pairs made up of images and labels that indicate the contained figure.

The constructor receives as input parameters the dimensions of an image. The dimensions of the input data are needed to define the size of the fully-connected layer; the convolutional one will know how to deduce the size from the tensors it receives as input.

CONVOLUTIONAL LAYER: the first parameter refers to the number of pixel values of the input images (i.e. the number of color values, color channels). Since we are working with color images, this parameter has the value 3. The convolutional layer generates 3 feature maps, applying 3 filters of size 5x5.

POOLING LAYER: reduces the size of input feature maps by 4 times. Max refers to the fact that the reduction is done by choosing the maximum value from each pixel group of the input feature maps.

FULLY CONNECTED LAYER: the input size of this layer is determined based on the results of the previous convolution and pooling operations as follows: the convolutional layer generates 3 feature maps using 5x5 filters. After applying the filters on the 32x32 size input image, 3 28x28 size feature maps result. 2 pixels from each end of the initial image are lost because the convolution is performed with a 5x5 filter, without padding or other additional operations. A 4x4 aggregation operation is applied to these filters, which reduces the size of the feature maps by 4 times on each dimension. Therefore, the size of the fully connected layer will be: 6x6x3.

The method that performs forward propagation is called forward. The following operations are performed in order:

- the image is propagated through the convolution layer and feature maps are generated at the output;
- a 4x4 aggregation is applied to these feature maps. Each group of 4x4 values from the initial feature map is reduced to a single value, equal to the maximum of 16;
- the ReLU activation function is applied to the previous result;
- the previous result is flattened by the fully connected layer;
- the output values consist of a set of 10 numerical values (one for each class) which, by normalization, indicate the probabilities that the value from the input of the forward method falls into each of the 10 classes.

The train method is the one that performs the training of the network. The training algorithm uses the training images for which the classes they belong to (labels) are known. Optimization involves searching for the values of the network weights that minimize the error generated by the network. For this, the search method uses the variations of the error in relation to the weights of the network, expressed by the components of the gradient of the error in relation to the weights. These gradient components must be reset at the beginning of the weight adjustment process. In the train method, the output is the result generated by the neural network for the image given as an input parameter. In the present case, the output is a tensor containing 10 values, one for each class. The higher the value associated with a class, the more it is considered that the network associates that class with the input data to a greater extent. The function that evaluates the error between the values provided by the neural network and the labels in the training data set is CrossEntropyLoss.