Project Name: Improving CIFAR-10 Dataset Accuracy

Course Name: Machine Learning

Course Number: ECE 532

Student Name: Sanita Haque

Unique ID: haquem

Date: 12/10/2023

Abstract:

The purpose of this project is to improve the accuracy of the CIFAR-10 dataset. To improve this accuracy, we need to build a new CNN model. The new proposed model is called “ImprovedLeNet”. The CIFAR-10 dataset comprises 60,000 32x32 color images in 10 classes, with 6,000 images per class. The revised model architecture incorporated several key modifications, including additional convolutional layers, smaller kernel sizes, batch normalization, and dropout layers. These changes were strategically implemented to capture more complex features in the images and to regularize the model for better generalization. During the training process, techniques like data augmentation, dynamic learning rates, and experimenting with the “RMSProp” optimizer. The enhanced model demonstrated significant improvements in accuracy compared to the standard LeNet (Used in HW-06) architecture.

Dataset Description:

The CIFAR-10 dataset is a fundamental collection of images widely used in the field of machine learning for benchmarking image recognition algorithms. The CIFAR-10 dataset consists of 60,000 32x32 color images divided into 10 classes, with each class containing 6,000 images. The classes represent various objects and animals such as cats, dogs, birds, automobiles, trucks, and ships. The dataset is split into 50,000 training images and 10,000 test images.

Model Architecture:

Convolutional Layers:

- conv1 and conv2 are convolutional layers with 64 filters each and a kernel size of 3x3, with padding set to 1. These layers are designed to extract features from the input images.
- bn1 and bn2 are batch normalization layers corresponding to conv1 and conv2, respectively. Batch normalization helps in stabilizing the learning process and reducing internal covariate shifts.

Pooling Layers:

pool1 is a max pooling layer with a window of 2x2. This layer reduces the spatial dimensions of the feature maps (from 32x32 to 16x16 after pool1).

Other Convolutional and Pooling Layers:

- conv3 and conv4, along with bn3 and bn4, follow the same pattern as the first two convolutional layers but with 128 filters. These layers capture more complex features.
- pool2 further reduces the feature map size to 8x8.

Dropout Layer:

dropout is a dropout layer with a dropout rate of 0.5, used to prevent overfitting by randomly zeroing out some of the elements of the input tensor.

Fully Connected Layers:

- fc1, fc2, and fc3 are linear layers that act as fully connected layers in a traditional neural network.
- fc_bn1 and fc_bn2 are batch normalization layers for the fully connected layers.
- The first linear layer (fc1) transforms the flattened feature maps into a 1024-dimensional space. fc2 further maps this to a 512-dimensional space. Finally, fc3 maps these features to the number of classes in the dataset (10 for CIFAR-10).

Forward Method (forward):

This method defines the forward pass of the network.

- The input x is passed sequentially through the convolutional layers, batch normalization layers, ReLU activation functions, and pooling layers.
- After the convolutional and pooling layers, the output is flattened before being passed through the fully connected layers.
- ReLU activation functions are used after each fully connected layer except the last one.
- The dropout layer is applied after the activations of the fully connected layers to regularize the network.
- The final output is obtained after the last fully connected layer, which can be used in conjunction with a loss function like Cross-Entropy for classification tasks.

The following table summarizes the Improved CNN model with a description of the parameters.

Layer Type	Layer Identifier	Description	Output Size
Input Layer	-	32x32 RGB Image	32x32x3
Convolutional	Conv1	64 filters, 3x3 kernel, padding 1	32x32x64
Batch Normalization	bn1	Normalizes Conv1 output	32x32x64
Activation	ReLU	Applies ReLU to bn1 output	32x32x64
Convolutional	Conv2	64 filters, 3x3 kernel, padding 1	32x32x64
Batch Normalization	bn2	Normalizes Conv2 output	32x32x64
Activation	ReLU	Applies ReLU to bn2 output	32x32x64
Max Pooling	Max Pooling 1	2x2 window	16x16x64
Convolutional	Conv3	128 filters, 3x3 kernel, padding 1	16x16x128
Batch Normalization	bn3	Normalizes Conv3 output	16x16x128

Activation	ReLU	Applies ReLU to bn3 output	16x16x128
Convolutional	Conv4	128 filters, 3x3 kernel, padding 1	16x16x128
Batch Normalization	bn4	Normalizes Conv4 output	16x16x128
Activation	ReLU	Applies ReLU to bn4 output	16x16x128
Max Pooling	Max Pooling 2	2x2 window	8x8x128
Fully Connected	FC Layer1	Maps to 1024 neurons	1024
Batch Normalization	fc_bn1	Normalizes FC Layer1 output	1024
Activation	ReLU	Applies ReLU to fc_bn1 output	1024
Dropout	Dropout	Applies dropout with a rate of 0.5	1024
Fully Connected	FC Layer2	Maps to 512 neurons	512
Batch Normalization	fc_bn2	Normalizes FC Layer2 output	512
Activation	ReLU	Applies ReLU to fc_bn2 output	512
Dropout	Dropout	Applies dropout with a rate of 0.5	512
Fully Connected	FC Layer3	Output layer, maps to 10 neurons	10

Differences from HW06:

The model is different from HW06. In HW06, we have used convolution, pooling, and fully connected layers. The details model is described above. In the proposed model, batch normalization, and dropout layers are added. Also, there are differences in learning rates, model optimizer, and number of epochs. Here the used learning rate is 0.001, the model optimizer is RMSProp, and the number of epochs is 100.

Conclusion:

This improved LeNet architecture is more sophisticated than the original LeNet, with additional layers and modern techniques like batch normalization and dropout. These enhancements make it more suitable for handling the complexity of the CIFAR-10 dataset. This project aims to get above 80% accuracy. This improved LeNet model has gained accuracy above 90% which is **90.25%**