2024

Student Name: Om Patel

Student ID: 400438318

Project Title: COMPSCI 4ML3
Assignment 4

Supervisor: Hassan Ashtiani

DEPARTMENT OF COMPUTER SCIENCE

McMaster University

Contents

1 Task 3: Kaggle Competition

Username : ompatel016
After running 20 epochs, this was the final accuracy:

```
Epoch [20/20], Train Loss: 1.8807,
Train Accuracy: 65.77%,
Val Loss: 2.0699,
Val Accuracy: 59.95%
Model saved with Val Accuracy: 59.95%
```

For the Kaggle competition, I trained a deep CNN for image classification using the CIFAR-100 dataset. The model architecture is based on a Residual Network, ResNet, which is a deep learning model designed to improve training in very deep neural networks through the introduction of residual connections.

For the CIFAR-100 dataset, we are given a collection of 60,000 32x32 RGB images, which are divided into 100 classes. In my code, I dedicated a portion of the training data for validation purposes.

The model which I used in this project is a custom implementation of the ResNet architecture. I also added the residual blocks, which helps prioritize the training of deeper networks by allowing the network to learn identity mappings in some layers. The main reason of the model is to take an image from the CIFAR-100 dataset, pass it through several layers, and produce a final classification output.

Now, if we look at the ResidualBlock, it is the main building "block" of the ResNet model. It has two 3x3 convolutional layers with LeakyReLU and Batch Normalization applied after each convolution. It also has a Skip connection that adds the input to the output after it has passed through the convolutions. The reason why I did this is that I thought it would help mitigate the vanishing gradient problem by allowing gradients to flow directly through the network without passing through non-linear transformations. Lastly, it handles the case when the dimensions of the input and output do not match. This would lead to a 1x1 convolution being applied to the shortcut to match the dimensions.

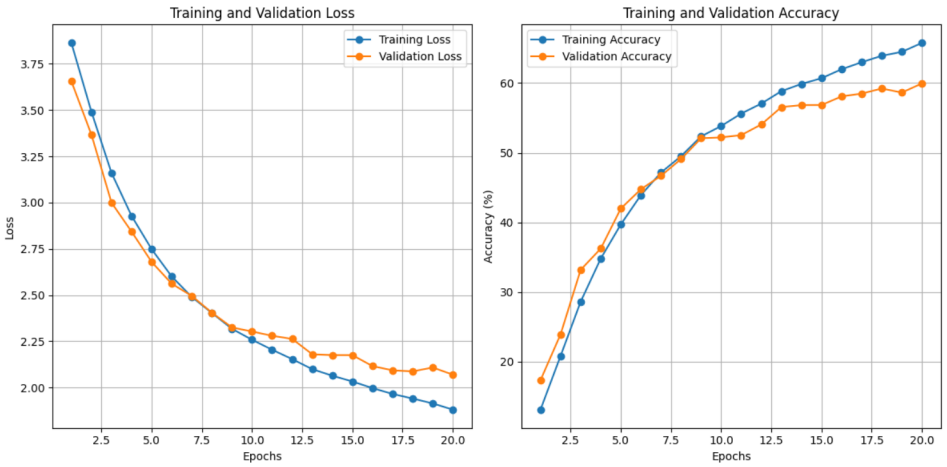
Now, if we look at the ResNet model class, it consists of the architecture by stacking multiple residual blocks. The first layer has a 3x3 convolution with a stride of 1, followed by Batch Normalization and LeakyReLU. Then, we have the layer structure which consists of four layers, layer1, layer2, layer3, and layer4. Each of those layer consists of multiple residual blocks:

- Layer1 has the first set of residual blocks, and does not do down sampling.
- Layer2, Layer3, and Layer4 all contain residual blocks with a stride of 2, and they contain down sampling.

After all the residual blocks, I added an adaptive average pooling layer. This is used to reduce the spatial dimensions to a 1x1 feature map. Next, to help with generalization and prevent over-fitting, a Dropout layer is used before the final fully connected layer. For the final connected layer, the final output is obtained by passing the pooled features through a fully connected layer that outputs the predictions for 100 classes.

The next thing, I added the Adam optimizer with a learning rate of 0.001 and a weight decay of 1e-4. I chose the Adam optimizer because it has good adaptive learning rate capabilities, which are particularly useful for training deep networks. Next, I added a ReduceLROnPlateau scheduler. This was meant to be used to reduce the learning rate when the validation loss plateaus, with a patience of 2 epochs. This helped me in optimizing the learning rate to prevent over-fitting.

Next, to visualize the performance of the model, the code plots both the loss and accuracy over the epochs for both the training and validation sets. This helped me understand how well the model is performing, whether it's over-fitting, and whether further adjustments are needed. This helped me modify my model and other aspects to help improve the accuracy. After running 20 epochs, this was the following graph outputted:



Moving on, after we have trained the model, it is evaluated on the test set to estimate its performance on unseen data. The model predicts the class labels for each image in the test set, and then saved in a CSV file for submission.

Overall, the my code uses a ResNet architecture for image classification on the CIFAR-100 dataset. Then the model is trained effectively using a series of data augmentation techniques and is evaluated on both the training and validation

sets. Then, the best model is saved and used for final predictions on the test dataset.