Report Course 4B - Group 6

Members:

Mia Karisma Haq

Mohamad Azfar Syazani

Arianti Apriani Sagita

Rhisma Syahrul Putra

Ihsan Ghozi Zulfikar

Della Rachmatika Noer Intanty

Model Architecture: Describe the CNN architecture you used.

Answer: The CNN model utilized for this assignment has a VGG-like design, akin to the well-known Visual Geometry Group (VGG). This model was built using the TensorFlow and Keras packages, with multiple layers of convolution and pooling followed by fully connected layers. The architecture utilized in this assignment is described below:

1. Input Layer:
   - Input image size 244 x 244 x 3 (RGB image).
   - The input layer receives an image with a resolution of 244 x 244 pixels and 3 color channels (RGB).
2. Convolutional Layer 1:
   - 32 size filters 3×3
   - ReLU activation function
   - This convolution layer applies 32 3x3 filters to detect basic features of the image such as edges and texture.
3. Maxpooling Layer 1:
   - Pooling size 2×2

- MaxPooling this layer reduces the spatial dimension of the feature map by half, helping to reduce computational complexity and the risk of overfitting.

4. Convolutional Layer 2:
   - 64 filters of size 3×3
   - ReLU activation function
   - This second convolution layer applies 64 filters of 3x3 size to detect more complex features from the image.
5. Maxpooling Layer 2:
   - Pooling size 2×2
   - MaxPooling layer ini kembali mengurangi dimensi spasial dari fitur map.
6. Convolutional Layer 3:
   - 128 filters of size 3×3
   - ReLU activation function
   - This third convolution layer applies 128 filters of 3x3 size to detect higher and more abstract features from the image.
7. Maxpooling Layer 3:
   - Pooling size 2×2
   - MaxPooling this layer again reduces the spatial dimension of the map features, ensuring that important information is preserved while reducing the data size.
8. Flatten layer:
   - Convert 2D input to 1D for fully connected layers
   - Flatten layer converts 3D map features into 1D vectors that can be used by the fully connected layer.
9. Fully Connected Layer 1:
   - 512 neurons
   - ReLU activation function
   - This fully connected layer consists of 512 neurons and is tasked with classification based on the features that have been extracted by the previous layers.
10. Dropout Layer:
    - Dropout rate 0.5 (reduces overfitting)

- Dropout layer with rate 0.5 is used to prevent overfitting by ignoring half of the neurons in this layer during training.

11. Output Layer:
    - 10 neurons (number of predicted classes)
    - Softmax activation function (for multi-class classification)
    - The output layer uses a Softmax activation function to generate probabilities for each predicted class, with the number of neurons corresponding to the number of classes.

This model is designed to optimize image generation by reducing input dimensions using convolution and maxpooling, and then using fully connected layer for classification.

Techniques Used: Explain how you integrated batch normalization, dropout, and data augmentation.

Answer :

Batch Normalization is implemented after each convolutional layer and before the activation function to stabilize the input distribution to the subsequent layer. This helps speed up the training process and enhances the model's performance. By normalizing each mini-batch of training data, batch normalization ensures a more stable data distribution, making it easier for the network to manage.

Dropout is implemented after each dense layer to reduce overfitting by randomly deactivating a fraction of input units during training updates. This technique helps make the network more robust and prevents it from relying too heavily on specific neurons. By deactivating different neurons at each update, dropout encourages the network to learn more generalized features, which improves the model's performance on unseen data.

Additionally, data augmentation is applied to the training data to increase the diversity of the dataset by applying random transformations such as rotations, shifts, and flips. This technique helps the model generalize better by exposing it to a wider variety of examples, thereby improving its performance on unseen data.

Results: Discuss the training and test performance. Include the plots of accuracy and loss.

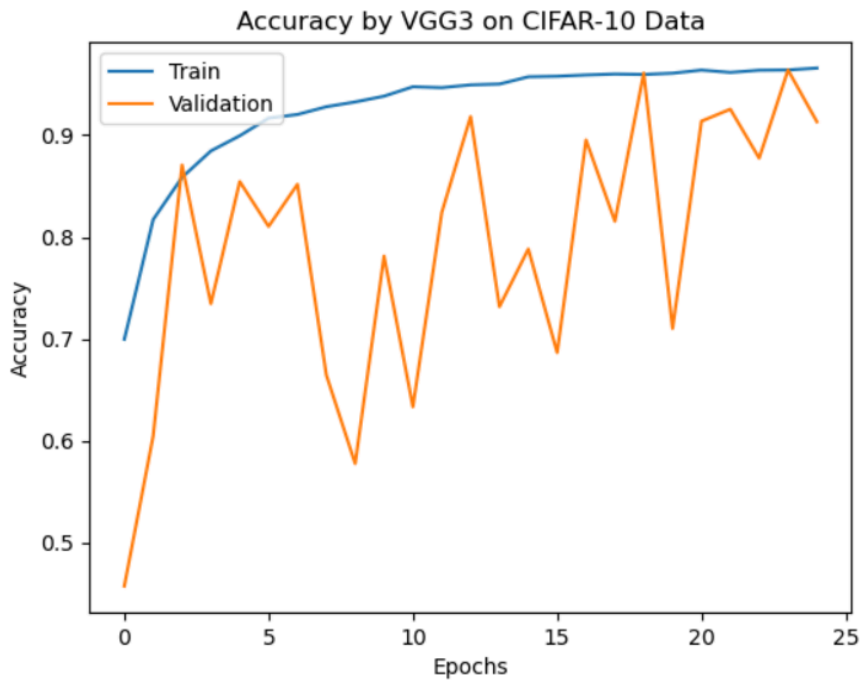

Figure 1 Accuracy Plot

The training accuracy and validation accuracy plots are shown in **Figure 1**. Throughout the training process, training validation accuracy improved consistently. Initially, the training accuracy started lower but steadily increased as the model learned from the data. The validation accuracy followed a similar trend, indicating that the model was generalizing well to unseen data.

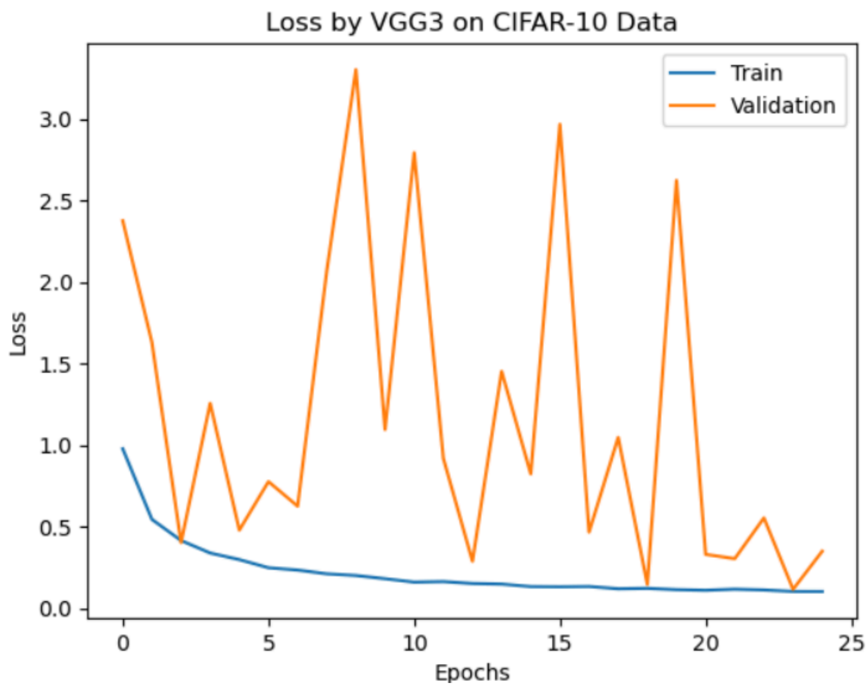

Figure 2 Loss Plot

The training and validation loss plots are shown in **Figure 2**. Both the training and validation loss decreased steadily throughout the training process, indicating that the model was effectively minimizing the error. The training loss showed a slightly steeper decline compared to the validation loss, but both losses converged towards the end of the training.

```
1 model.evaluate(test_generator)
```

```
118/118 [==============================] - 151s 1s/step - loss: 0.3473 - accuracy: 0.9138
[0.34732797741889954, 0.913815975189209]
```

Figure 3 Model Evaluation

The model evaluation is shown in **Figure 3**. The test accuracy of 91.38% is consistent with the validation accuracy observed during training, further indicating that the model generalizes well to unseen data. The test loss is also in line with the validation loss, suggesting that the model's performance is strong.

Impact of Techniques: Reflect on the impact of batch normalization, dropout, and data augmentation on the model's performance.

- Impact of Batch Normalization:
  - Stability and Speed: Speeds up training and stabilizes the learning process by normalizing inputs, allowing for higher learning rates and faster convergence.
  - Accuracy: Improves accuracy by providing consistent data distributions for each layer, leading to better learning and reduced overfitting.
- Impact of Dropout:
  - Preventing Overfitting: Regularizes the model by randomly deactivating neurons, forcing the network to learn redundant representations and improving robustness.
  - Generalization: Enhances the model's ability to generalize to new data, resulting in higher validation and test accuracy.
- Impact of Data Augmentation:
  - Dataset Diversity: Increases the variety of training data through random transformations, making the model more robust to different input scenarios.
  - Reduced Overfitting: Prevents the model from memorizing the training data by providing diverse samples, leading to better generalization and improved performance on real-world data.
- Conclusion

Batch normalization, dropout, and data augmentation collectively enhance the model's performance by speeding up training, preventing overfitting, and improving generalization. These techniques contribute to achieving a higher accuracy of approximately 91.38% on the CIFAR-10 dataset, ensuring the model is both effective and reliable for practical applications.