|  |  |
| --- | --- |
| **Ex No: 3b**  **Date: 12th September 2024** | **Small Image Classification Using Convolutional Neural Network** |

**Objective:** The experiment's objective is to build, train, and evaluate ANN and CNN models for image classification using a CIFAR10 dataset from TensorFlow and Keras. It is a multi-class classification (10 classes) which aims to compare their performance by tuning hyperparameters, analysing accuracy, and drawing conclusions about which architecture is better suited for this task.

**Description:**

**Artificial Neural Networks (ANN)** is a type of machine learning model inspired by the biological neural network in the human brain. It consists of layers of neurons (input, hidden, and output) that transform input data through nonlinear functions and learn patterns by adjusting the weights during training. ANNs are suitable for tabular and low-dimensional data, but they may struggle with high-dimensional data like images.

**Convolutional Neural Networks (CNN)** is a specialized kind of ANN primarily used for image and video processing. It uses convolutional layers, which apply filters (kernels) to input data to detect spatial hierarchies and patterns such as edges, textures, and shapes.CNNs are highly effective for image classification tasks as they capture spatial dependencies in images using pooling layers and convolutional operations.

**Model Summary:**

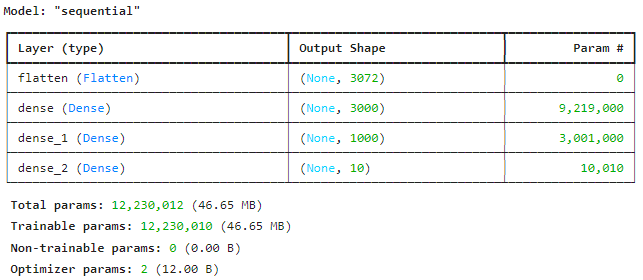
****

Figure 1: A sequential ANN model for Small Image classification.

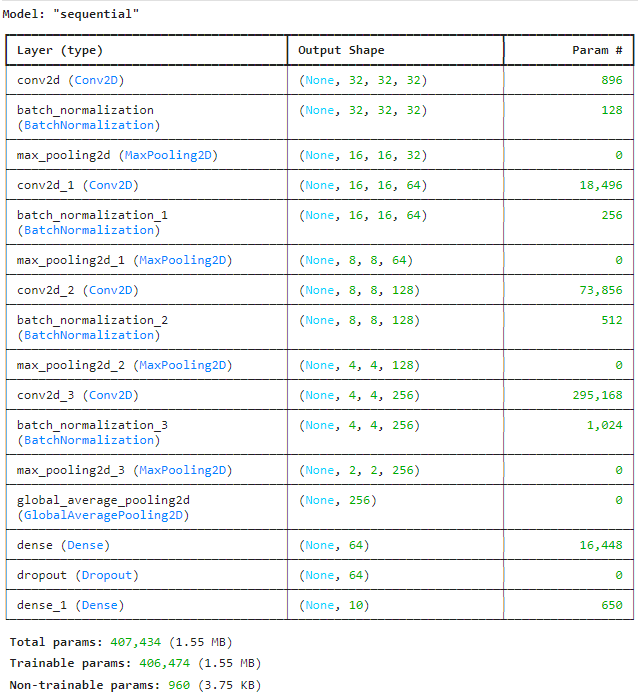


Figure 2: A sequential CNN model for Small Image classification.

**Building the parts of the algorithm**

Here are the steps involved in building each part of the algorithm:

1. **Data Preprocessing**: The dataset is loaded and preprocessed into training and testing sets. The class vectors of the input data is converted to binary class matrices.
2. **Model Architecture**:
   * The neural network model is built using Keras. It contains multiple layers, including input, hidden (dense and dropout), and output layers.

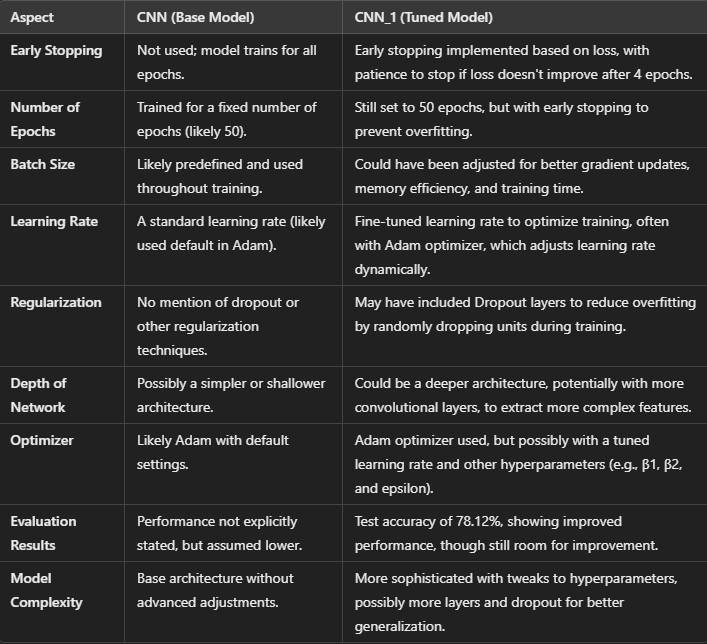
**ANN Model:**

* Input Layer: Accepts flattened image input (usually 1D).
* Hidden Layers: Fully connected (dense) layers with activation functions like ReLU.
* Output Layer: For classification, uses sigmoid activation for multi-class classification tasks.

**CNN Model:**

* Convolutional Layers: Use filters to perform convolutions on image data to extract features.
* Activation Function: ReLU is used to introduce non-linearity.
* Pooling Layers: Downsample the feature maps to reduce dimensions and computation.
* Flatten Layer: Converts the 2D feature maps into 1D.
* Fully Connected Layers: Final dense layers for classification.
* Output Layer: Uses softmax for multi-class classification.

1. **Compilation**:
   * The model is compiled using a loss function (spare categorical cross-entropy) and an optimizer (SGD for ANN and Adam for CNN), with metrics set to monitor accuracy.
2. **Training**:
   * The model is trained over variable epochs, with accuracy and loss tracked during each epoch.



1. **Evaluation**:
   * The final performance is evaluated on a test set which gives the accuracy and loss.
   * The model is tested with some test images to understand the prediction and its probability.

**Inferences:**

CNN\_1 model improved the performance by using early stopping, which prevented overfitting and saved time by halting training when no further improvement in loss was detected. It had additional dropout regularization to prevent overfitting, while the base CNN model did not employ any regularization. Hyperparameters such as number of epochs a were increased in CNN\_1. The architecture might have been deeper or more complex compared to the simpler structure of the base CNN, leading to better feature extraction and classification accuracy.

**Conclusion:**

**CNN\_1** achieved a test accuracy of **78.12%**, indicating that the model captured significant patterns in the data but has room for improvement. The classification report shows decent performance in various classes, but some classes, such as class 3 (recall 0.60), indicate challenges in identifying specific features.

The experiment concludes that CNN outperforms traditional ANN models in image classification due to the spatial feature extraction abilities of CNN, but further tuning and deeper architectures may be required for improved performance.

**GitHub Link:** [**https://github.com/tulasigr/DeepLearning**](https://github.com/tulasigr/DeepLearning)