Coding Challenge Final Write-Up

Problem Statement

Given a dataset of approximately 1,000 images, the objective is to develop a generalizable model capable of classifying four distinct animal expressions: *angry*, *happy*, *sad*, and *other*. The goal is to ensure that the model performs reliably not only on the provided dataset but also on new, unseen images.

Methods

The solution begins with data augmentation and normalization to enhance the diversity and generalizability of the dataset. Augmentation techniques applied include *random flipping*, *random color jittering*, and *random rotation*. These transformations help simulate various real-world scenarios, making the dataset more representative and improving the model's ability to generalize to unseen data. Normalization using the z-score technique ensures that input data is on a consistent scale, which helps optimize the learning process by preventing large gradients during backpropagation. Since the original dataset was already divided into *training*, *validation*, and *testing* subsets, we proceeded directly to model training.

For the architecture, we use a Convolutional Neural Network (CNN) and a Multi-Layer Perceptron (MLP) to perform feature extraction and classification, respectively. CNNs are particularly suited for image-based tasks because they exploit the inherent spatial relationships between neighboring pixels. Convolution operations extract local patterns, such as edges and textures, while pooling operations reduce dimensionality, making CNNs an effective embedding function for images.

After feature extraction, the MLP takes over to perform the classification. An MLP is composed of several layers of activated linear transformations. The concept behind these transformations is that applying non-linear activation functions, such as ReLU or sigmoid, between linear layers enables the model to capture complex, non-linear patterns in the data. This is necessary because the composite of linear functions alone are still linear, which is insufficient for fitting non-linear structures present in most real-world datasets. The final output layer of the MLP assigns probabilities to each of the four expression classes.

For the training process, *Adam* optimizer and `ReduceLROnPlateau` scheduler for possible early stopping needed [1][2]. Adam optimizer is chosen because it follows less hyperparameter sensitivity, faster convergence, less computation, and is a widely used effective optimizer.

Results and Discussion

Accuracy is used as the evaluation metric since the dataset is balanced (each class contains 250 samples).

$$\text{Accuracy} = \frac{\text{Number of Correctly Predicted}}{\text{Number of Images}}$$

Given that no specific focus on false positives or false negatives is required (as mistakes aren't critical), overall correctness—measured by accuracy—is the most appropriate metric. Unfortunately, the model achieves only **0.236** accuracy, which is slightly worse than random guessing.

The primary issue lies in the shallow architecture of CNN. This design limits the depth of feature extraction, relying heavily on max-pooling layers, which reduce dimensionality too aggressively. As a result, the network fails to generalize effectively. Moreover, training the model longer isn't feasible since the training loss has already plateaued, indicating that further improvement is unlikely.

Interestingly, this problem may not appear in MLP models. Expanding the MLP architecture (e.g., making it wider or deeper) could lead to overfitting, but its training loss behavior shows more potential for improvement.

Overall, the current CNN structure isn't producing meaningful embeddings, as excessive pooling likely discards critical information.

Future work should focus on using deeper CNN architectures to enhance feature extraction and improve generalization.

Reference

[1] Keras API Documentation, Adam, https://keras.io/api/optimizers/adam

[2] Keras API Documentation, https://keras.io/api/callbacks/reduce_lr_on_plateau/

[3] Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017, July). On calibration of modern neural networks. In *International conference on machine learning* (pp. 1321-1330). PMLR.

NOTE: My knowledge for this project refers to Deep Learning by Goodfellow et al (2016). in general. *http://www.deeplearningbook.org*