Deep Learning (CSL7590)
Assignment 2

Submitted By:

Shivani Tiwari (M24CSA029)
Suvigya Sharma (M24CSA033)

Submitted To:
Dr. Deepak Mishra

Collab link:

https://colab.research.google.com/drive/1PPocx_-sBkQ9U0x4MZmveJ_XPs-v3pR9?usp=sharing

Objective: The goal of this assignment is to build a single Convolutional Neural Network (CNN) in PyTorch that can classify images from the CIFAR-100 dataset into three different categories at the same time:

1. Fine-level classification – Identify the exact object from 100 possible classes.
2. Superclass classification – Group the object into one of 20 broader categories.
3. Synthesized group classification – Further categorize the object into one of 9 custom-defined groups based on shared characteristics (e.g., vehicles, aquatic animals, plants, etc.).
    - Plants/Parts of Plants → Superclasses: flowers, trees, fruits and vegetables
    - Vehicles → Superclasses: vehicles1, vehicles2
    - Invertebrates → Superclasses: non-insect invertebrates, insects
    - Aquatic Animals → Superclasses: fish, aquatic mammals
    - Large Animals → Superclasses: large carnivores, large omnivores and herbivores
    - Man-made Articles → Superclasses: food containers, household electrical devices, household furniture, large man-made outdoor things
    - People → Superclass: people
    - Normal Terrestrial Animals → Superclasses: reptiles, medium-sized mammals, small mammals
    - Outdoor Scenes → Superclass: large natural outdoor scenes

4. To improve performance and handle class imbalances, we introduce a severity-based loss function that penalizes misclassifications based on their severity. The model is trained and evaluated across multiple train-test splits (70:30, 80:20, and 90:10).

Methodology:

1. **Data Preprocessing:**
    - The CIFAR-100 dataset is loaded and preprocessed using normalization.
    - A custom PyTorch dataset class (CIFAR100_Custom) is created to map each fine class to its respective superclass and group category.
    - The dataset is split into training and testing sets with three different ratios (70:30, 80:20, 90:10).
    - To handle data imbalance, we use a weighted sampler, ensuring that underrepresented classes contribute equally to training.
2. **CNN Architecture model:**
    - Feature Extraction Layers: 3 sets of convolutional layers with Batch Normalization and ReLU activation. Max-pooling layers to reduce dimensionality.
    - Fully Connected Layers: Two dense layers with Batch Normalization and Dropout to prevent overfitting.
    - Three Classification Heads: One output layer for each classification task (fine class, superclass, and group classification).
3. **Loss Function & Training Process**

    The cross-entropy loss is used for classification.

    A severity penalty matrix is introduced to assign different levels of penalty based on the type of misclassification:

    - Same superclass → Minor penalty.
    - Same group but different superclass → Moderate penalty.
    - Different group → Highest penalty.

    The final severity-weighted loss function adjusts training based on these penalties.

    Training Process:

    - The model is trained for 20 epochs using the Adam optimizer (learning rate = 0.001).
    - The dataset is divided into three train-test splits to analyze model performance across different training size.

4. **Evaluation & Performance Metrics**
   - After training, the model is tested on the validation dataset.
   - Accuracy is computed for all three classification heads (fine class, superclass, and group classification).
   - Loss trends over training epochs
   - Confusion matrices are plotted for each classification task to visualize misclassifications.
   - The total number of trainable and non-trainable parameters.

Results

Total Trainable Parameters: 13488001

Split Ratio	Final Loss	Class Accuracy	Superclass Accuracy	Group Accuracy
70:30	0.492	52.06	52.55	54.95
80:20	0.4829	53.56	53.79	57.44
90:10	0.459	55.65	56.01	59.25

Observations

- Accuracy improves with larger training sets: As the training data increases, model accuracy improves across all classification tasks.

- Group classification performs better than fine classification: This suggests that distinguishing broad categories is easier than differentiating fine-grained classes.

- Loss reduction is significant over epochs: The final loss values show significant improvement over training.

Bonus part: Enhancing CNN Training with Severity-Based Misclassification Penalties

Split Ratio	Final Loss	Class Accuracy	Superclass Accuracy	Group Accuracy
70:30	0.3622	55.31	55.18	58.04
80:20	0.3507	56.75	56.67	59.05
90:10	0.4401	59.52	59.36	62.48

Observations

1. Bonus approach reduces severe misclassification: Accuracy gains indicate a lower frequency of misclassification across groups.
2. Computational cost increases slightly: Since the loss function includes additional penalty terms, training takes marginally longer.
3. Fine-class accuracy benefits the most: The model learns to make more precise predictions within the superclass and groups

References

1. https://www.kaggle.com/datasets/fedesoriano/cifar100
2. https://www.geeksforgeeks.org/introduction-convolution-neural-network/
3. https://www.coursera.org/learn/convolutional-neural-networks
4. https://pytorch.org/docs/stable/nn.html
5. https://www.geeksforgeeks.org/handling-imbalanced-data-for-classification/