CSCI 677 –Advance Computer Vision- Fall 2024 - HW 4

USC ID: 1154164561
NAME: Srija Madarapu
EMAIL: madarapu@usc.edu

Code:

This code implements a machine learning pipeline using PyTorch to classify images from the CINIC-10 dataset. It includes the definition of LeNet-5 and its variations (with Batch Normalization and L2 Regularization), training these models, evaluating their performance, and visualizing the results using confusion matrices.

https://colab.research.google.com/drive/1ZEKg8j18YelSM3mRSp4H89xaktxegNL9?usp=sharing

FinalData:

For a comprehensive view of the data, please refer to the attached spreadsheet. This document provides a detailed breakdown

https://docs.google.com/spreadsheets/d/13P8In0AXoNKVXOAACyjM5bjd1eYhNAdmnW1bxU08KFA/edit?usp=sharing

Images:

https://drive.google.com/drive/folders/1UlEfaZshnCrUVwJw5MVlM-zoAwgF-0DU?usp=sharing

LeNet-5 and its Variants:

- LeNet-5: This is a simple, classic CNN architecture. It consists of:
  - Two convolutional layers (with ReLU activations and average pooling).
  - Three fully connected layers.
- LeNet-5 with LR=0.01: LeNet-5 with a higher learning rate (0.01) compared to the baseline (0.001).
- LeNet-5 with SGD: LeNet-5 using Stochastic Gradient Descent (SGD) optimizer instead of Adam.
- LeNet-5 with Batch Normalization: This variant adds batch normalization layers after each convolutional layer (conv1 and conv2). Batch normalization helps stabilize the learning process and improve model convergence by normalizing the activations within each mini-batch.
- LeNet-5 with L2 Regularization: This version uses L2 regularization (weight decay) during optimization. L2 regularization penalizes large weights, preventing overfitting by forcing the model to find smaller weights that better generalize.

ResNet-9:

- The BasicBlock class defines the building block for ResNet-9. It includes two convolutional layers, batch normalization, and a residual connection for efficient learning.
- The ResNet9 class builds the entire network with four stages. Each stage uses the _make_layer function to create two BasicBlocks with increasing output channels (64 -> 128, 128 -> 256, 256 -> 512). This allows for learning more complex features as the network progresses.
- The model uses global average pooling and a fully connected layer for classification of 10 classes.

LeNet-5 and its Variants:

1. For your main experiment setting, show the evolution of training losses and validation accuracy with multiple steps (training log + curves) for the two networks

LeNet-5 (Standard)

Seems to have a good balance between training and validation accuracy, suggesting it's not overfitting or underfitting significantly.

- **Training Loss:**
  This steady reduction indicates the model is learning effectively and improving during training.
- **Validation Accuracy:**
  While some minor fluctuations are present, the overall pattern demonstrates a consistent improvement in the model's ability to generalize to unseen data.

LeNet-5 with LR=0.01

Lower learning rate might lead to slower convergence and potentially underfitting.

- **Training Loss:**
  The training loss decreases, but at a slower rate compared to the standard LeNet-5.
- **Validation Accuracy:**
  The validation accuracy increases, but it reaches a plateau earlier, suggesting potential underfitting or slower convergence due to the lower learning rate.

LeNet-5 with SGD

SGD can be less stable than more advanced optimizers like Adam.

- **Training Loss:**
  The training loss decreases, but with more fluctuations compared to the standard LeNet-5.
- **Validation Accuracy:**
  The validation accuracy increases, but it's less consistent, indicating that SGD might be less stable than more advanced optimizers.

LeNet-5 with batch normalization

BatchNorm often helps in stabilizing training and improving generalization.

- **Training Loss:**
  The training loss decreases rapidly and stabilizes at a lower value.
- **Validation Accuracy:**
  The validation accuracy increases significantly, suggesting that BatchNorm helps in stabilizing training and improving generalization.

LeNet-5 with L2 reguralization

L2 regularization can help prevent overfitting by penalizing large weights.

- **Training Loss:**
  The training loss decreases, but it might be higher than the standard LeNet-5 due to the regularization penalty.
- **Validation Accuracy:**
  The validation accuracy increases, but it might not be as high as the BatchNorm version, indicating that the regularization strength might be too high or too low.

2. Show the confusion matrix and per-class classification accuracy for this setting.

Included in the spreadsheet in final data link above.

Base LeNet-5: Shows relatively balanced performance across classes, with some classes (e.g., 2, 3, 5) being more challenging.

LeNet-5 with LR=0.01: Struggles particularly on classes 2 and 5, indicating potential underfitting or poor optimization.

LeNet-5 with SGD: Shows a similar trend to the base model but with slightly lower overall performance.

LeNet-5 with BatchNorm: Improves performance across most classes, especially for classes 2 and 3.

LeNet-5 with L2 Regularization: While it improves performance on some classes, it significantly degrades performance on classes 2, 3, 8, and 9, suggesting excessive regularization.

3. Show some examples of failed cases, with some analysis if feasible.

Classes 2, 3, and 5: These classes consistently show lower accuracy across all models. This could indicate that these classes might have more intra-class variability (differences within the same class) or inter-class similarity (similarities to other classes). This makes it more challenging for the models to accurately distinguish between instances of these classes.
L2 Regularization Model: The significant drop in accuracy for classes 2, 3, 8, and 9 in the L2 Regularization model suggests that the model might be over-regularized. This means that the regularization penalty is too strong, leading the model to underfit the training data. As a result, the model becomes too simple and fails to capture the complex patterns in these classes.
BatchNorm Model: The BatchNorm model generally performs better across most classes. This suggests that BatchNorm helps in stabilizing the training process and improving the model's generalization ability. By normalizing the input to each layer, BatchNorm can help the model learn more efficiently and effectively.

4. Compare your results for the two networks and any variations you may have tried

Base LeNet-5:

- **Strengths:**
  - Achieves a decent overall accuracy of 51.95%.
  - Shows a steady improvement in training loss and validation accuracy over epochs.
- **Weaknesses:**
  - Could potentially benefit from further tuning of hyperparameters or architectural modifications to improve performance.

LeNet-5 with LR=0.01:

- **Strengths:**
  - Provides a stable training process.
- **Weaknesses:**
  - Lower learning rate might lead to slower convergence and suboptimal performance.
  - The lower validation accuracy compared to the base model suggests potential underfitting.

LeNet-5 with SGD:

- **Strengths:**
  - A simple and classic optimization algorithm.
- **Weaknesses:**
  - Can be less stable than more advanced optimizers like Adam or RMSprop.
  - The fluctuations in validation accuracy indicate potential instability during training.

LeNet-5 with BatchNorm:

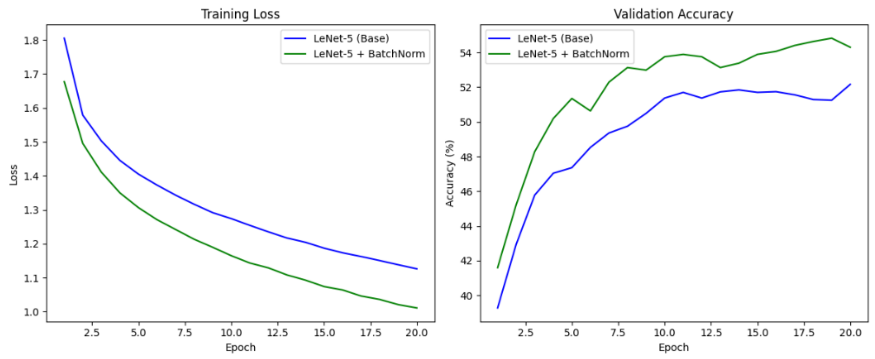
- **Strengths:**
  - Significantly improves validation accuracy compared to the base model.
  - Helps stabilize training and prevents overfitting.
- **Weaknesses:**
  - Might introduce additional computational overhead.

LeNet-5 with L2 Regularization:

- **Strengths:**
  - Can help prevent overfitting by penalizing large weights.
- **Weaknesses:**
  - In this case, the regularization strength might be too high, leading to underfitting and reduced performance.

Overall:

- **LeNet-5 with BatchNorm** appears to be the most effective variation, significantly improving performance over the base model.
- **LeNet-5 with L2 Regularization** seems to be over-regularized, leading to a decrease in performance.
- **LeNet-5 with LR=0.01** and **LeNet-5 with SGD** might benefit from further tuning or the use of more advanced optimization techniques.



ResNet-9:

1. For your main experiment setting, show the evolution of training losses and validation accuracy with multiple steps (training log + curves) for the two networks

ResNet-9 (Standard)

- **Training Loss:**
  The training loss decreases steadily over epochs, indicating effective learning.
- **Validation Accuracy:**
  The validation accuracy increases initially and then plateaus, suggesting potential overfitting.

ResNet-9 (LR 0.001)

- **Training Loss:**
  The training loss decreases steadily, but at a slower rate compared to the standard model.
- **Validation Accuracy:**
  The validation accuracy increases, but it doesn't reach the same peak as the standard model.

ResNet-9 (SGD Optimizer)

- **Training Loss:**
  The training loss decreases, but the convergence is slower compared to the other two models.
- **Validation Accuracy:**
  The validation accuracy is lower overall, indicating poorer performance.

2. Show the confusion matrix and per-class classification accuracy for this setting.

Included in the spreadsheet in final data link above.

ResNet-9(base):

- Achieved a higher peak validation accuracy.
- However, experienced overfitting later in the training process.

ResNet-9(LR 0.001):

- Showed more consistent improvement in validation accuracy.
- Didn't reach the same peak accuracy as ResNet-9(base).

ResNet-9 (SGD Optimizer):

- Lower overall performance compared to the other two models.
- Might benefit from more careful hyperparameter tuning.

3. Show some examples of failed cases, with some analysis if feasible.

Class 3 and Class 5:There's a significant number of instances where Class 3 is misclassified as Class 5, and vice versa.This suggests that these two classes might have similar visual characteristics, making it difficult for the model to distinguish between them.
Class 4 and Class 9:Similarly, Class 4 and Class 9 seem to be frequently confused.This could be due to overlapping features or a lack of distinguishing characteristics in the training data.
Class 6:Class 6 appears to be particularly challenging for the model, with a high number of misclassifications across multiple classes.This could indicate a lack of representative data for Class 6 or a complex visual pattern that the model struggles to capture.

4. Compare your results for the two networks and any variations you may have tried

Standard ResNet-9

- **Strengths:**

    - Strong Initial Performance: The model quickly achieves high validation accuracy, demonstrating effective learning.
    - Good Generalization: The model maintains a decent level of generalization throughout the training process.
- **Weaknesses:**
    - Potential Overfitting: As training progresses, the model might start to overfit, as indicated by the increasing validation loss and decreasing validation accuracy.

ResNet-9 with Lower Learning Rate (0.001)

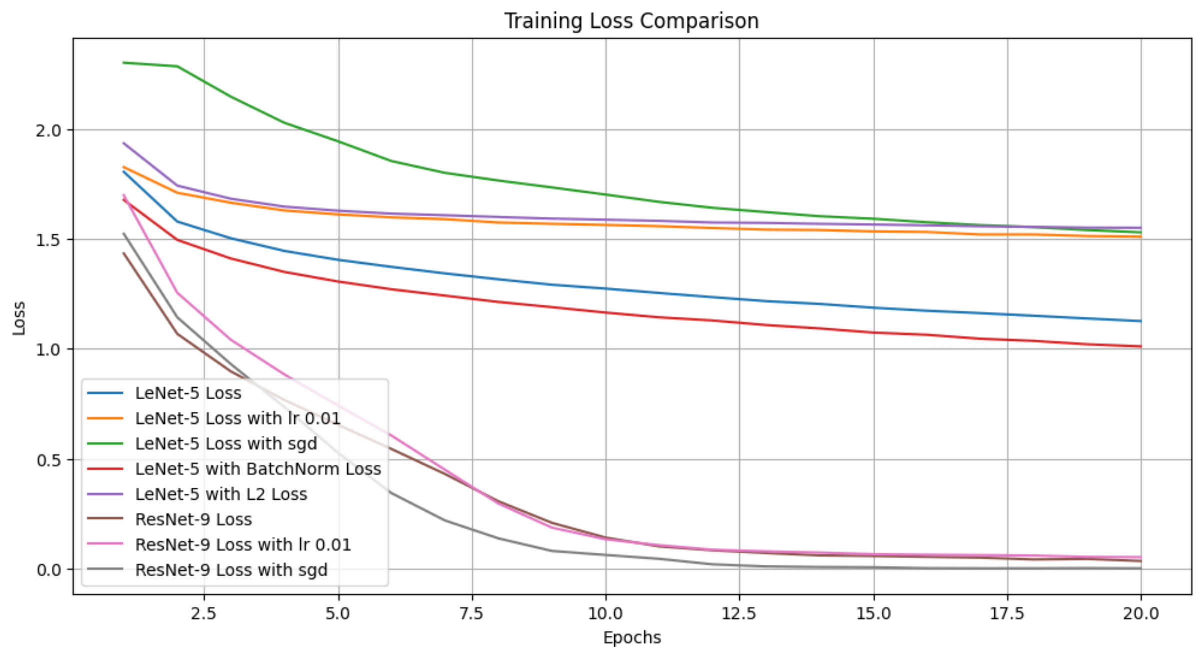
- **Strengths:**
    - More Stable Training: The lower learning rate leads to a more stable training process, preventing rapid oscillations in the loss function.
    - Improved Generalization: The model might be less prone to overfitting due to the slower learning rate.
- **Weaknesses:**
    - Slower Convergence: The lower learning rate can lead to slower convergence, requiring more training epochs to reach optimal performance.

ResNet-9 with SGD Optimizer

- **Strengths:**
    - Simpler Optimizer: SGD is a simpler optimizer that can be effective in certain scenarios.
- **Weaknesses:**
    - Slower Convergence: SGD can be slower to converge compared to more advanced optimizers like Adam.
    - Potential for Suboptimal Performance: SGD can be sensitive to hyperparameter tuning and might not perform as well as more sophisticated optimizers.

Overall:

- **Standard ResNet-9:** Achieves the best peak performance but might overfit.
- **ResNet-9 with Lower Learning Rate:** More stable training but slower convergence.
- **ResNet-9 with SGD:** Simpler but less effective compared to the other two models.


Training Loss Comparison
Loss
Epochs
LeNet-5 Loss
LeNet-5 Loss with lr 0.01
LeNet-5 Loss with sgd
LeNet-5 with BatchNorm Loss
LeNet-5 with L2 Loss
ResNet-9 Loss
ResNet-9 Loss with lr 0.01
ResNet-9 Loss with sgd