CS/SE 4AL3 Homework 4: Image Classification

TAs for questions: Reza Arabpour, Xiao (Ramsay) Shao

Fall 2025
Due: December 2nd

1 Overview

For this assignment, you will continue working with PyTorch. You will have two main tasks for this assignment: image classification on Fashion-MNIST and CIFAR-100. This assignment has 2 parts and will be graded out of 25 points. You must work on this assignment individually. A starter notebook file has been provided to you; your task is to complete it by filling in the required sections.

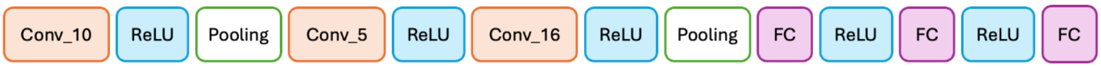
2 Part 1: Basic Image Classification (11 points)

Context: Image classification tasks have benefited significantly from neural networks. In the early days of computer vision, researchers painstakingly crafted features that resembled the real world. The advent of neural networks largely eliminated that effort. This, combined with the massive volume of image data available, lowers the barriers to creating image classifiers.

Dataset: The dataset was collected to serve as an alternative to EMNIST, a popular dataset for computer vision. Fashion-MNIST comprises 28×28 grayscale images of 70,000 fashion products, gathered from the European online platform Zalando. It has 10 categories, with 7,000 images per category. The training set has 60,000 images, and the test set has 10,000 images. The dataset is freely available at this link and you must download it for the assignment (or use the Python code to download it automatically).

Challenge: You will implement a lightweight convolutional neural network with the specifications below, that can run on your CPU. Your goal is to use the images of various clothing to correctly identify their class. For this assignment, select all 10 categories reported in Table 2 (Page 3) of this research paper (Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms). Notice Table 3 of the same paper, which gives a detailed benchmark of different models and their accuracy measures on this dataset. This is a common practice in machine learning, and for this assignment, you will familiarize yourself with this practice. You will evaluate your model against this benchmark and report model performance as compared to other models of different families on the same task.

Goals: Implement a convolutional neural network to classify images from different training sets. The given convolutional architecture is adopted from common CNN architectures, but is smaller in depth and width, so you do not need a high-performance GPU to train it. The layers must be stacked in left-to-right order, as shown here:



1. Build the model above with the following specifications:
   (a) Input size: 28×28 with one gray channel
   (b) Conv_10: Convolution layer with 10 computational units
   (c) Conv_5: Convolution layer with 5 computational units

    (d) Conv_16 Convolution layer with 16 computational units
    (e) Filter/kernel size for convolutional layers: 3
    (f) ReLU: Layers with ReLU activation function
    (g) Pooling: Performs max pooling operation and uses a filter size of 2
    (h) FC: This is a fully connected layer (nn.linear)

2. For training, use a cross-entropy loss (nn.CrossEntropyLoss) and SGD optimizer (torch.optim.SGD)

3. Run the model for at least 15 epochs

4. Feel free to select your own values for batch size and learning rate for the neural network

5. Generate a graph to show training and validation loss. You may want to select the validation set from the training set by randomly splitting the data.

6. Evaluate the model on the test set

7. Report the model performance by showing the test accuracy

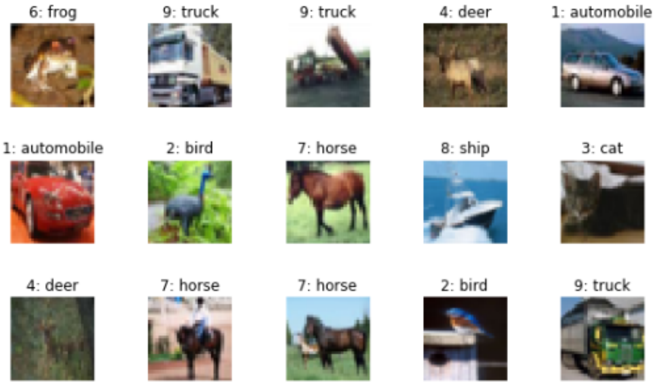
8. In the designated areas of the shared notebook, answer the following questions:

    (a) Which techniques from Table 3 report better performance than your method? Which performs worse? Why do you think this is the case? Briefly explain.

    (b) Do you have any other ideas about how to improve your method? Briefly explain.

3 Part 2: CIFAR-100 Classification (14 points)

Goals: In this part, you will reuse a convolutional neural network to classify images from a more challenging dataset with 100 classes. We will move from Fashion-MNIST (10 classes, grayscale) to CIFAR–100 (100 classes, RGB images). This will let you see how model capacity and dataset difficulty affect performance. CIFAR–100 consists of 32×32 colour images in 100 object categories. Dataset description: `https://www.cs.toronto.edu/~kriz/cifar.html`. The example images of CIFAR–100 are shown as follows:



1. You can use the similar convolutional architecture from Part 1, adapted to the CIFAR–100 input:

    (a) Input size: 32×32 with three colour channels (RGB);

    (b) Convolutional blocks: reuse the convolutional architecture from Part 1 (three convolutional layers with ReLU and max pooling), but modify only what is strictly necessary to accept 3-channel input and produce logits for 100 classes.

    (c) The final fully connected layer should have output dimension 100 (one logit per CIFAR–100 class).

2. For training, use a cross-entropy loss and SGD optimizer. You may reuse your learning rate, momentum, and batch size from Part 1, or adjust them for CIFAR–100 if needed.

3. Split the CIFAR–100 dataset into a training subset and a validation subset, you may follow the same rules in Part 1.

4. Run the model for at least 15 epochs

5. Generate a graph to show training and validation loss

6. For this assignment, your implementation should achieve at least **30%** test accuracy on CIFAR–100. If your results are significantly below this, you should treat that as a signal that something is likely wrong with your model. ResNet-style models around 2016 report test error near 20% on CIFAR–100 (i.e., about **80%** accuracy) when trained with strong data augmentation and many layers.[1] This is evidence that this task is much more difficult than the task in Part 1 and high performance is not necessary to receive credit.

7. In the designated areas of the shared notebook, answer the following questions:

   (a) Compared to your Fashion-MNIST results in Part 1, how did the training and validation curves change on CIFAR–100? Did you observe more overfitting, underfitting, or slower convergence? Explain briefly.

   (b) Based on the ResNet results above (around 80% accuracy) and your own results (around 35–45%), what architectural differences do you think account for most of the performance gap? Consider depth, width (number of channels), skip connections, data augmentation, and any modifications you have made to your own architecture.

   (c) What change did you make to your architecture that you were most surprised about? This could be a large increase or decrease in performance, or something you added that didn't change anything. What is your big takeaway from that experience?

4 Deliverables

- Please submit your work as **a single Jupyter Notebook**. The starter Jupyter Notebook has been provided. Your code should be well-commented and described. There is no restriction on the notebook file name.

- **Please do not alter** the notebook structure. You should only fill in the blocks of the code that say "#CODE HERE" and explain in text blocks noted "#EXPLAIN HERE". For the explanations, you should write at least 3-5 sentences.

- You should include any disclaimers about the use of AI in the first text block of the Jupyter notebook, as dedicated in the notebook, citing which parts of code the AI have been used.

- The Jupyter Notebook should include all the **codes already executed and saved, before submission**. TAs won't run your code blocks. Hence, ensure that all code blocks are executed by clicking "Run All". Failure to provide a previously run Jupyter notebook will result in grade deductions.

- You are **only** permitted to use the libraries provided in the Jupyter notebook. You will receive a zero if any other libraries are used.

- Note that scikit learn is available for this assignment. You may add import statements within these packages for your convenience, e.g. `from sklearn.PACKAGE import CLASS`.

- You need to use Python version 3.12. Do not upload any datasets.

[1] Zagoruyko S, Komodakis N. Wide residual networks[J]. arXiv preprint arXiv:1605.07146, 2016.

5 Help

Please use the Teams channel and tutorials to ask questions about the assignment when you need guidance or pointers on this homework. You are free to discuss your approach and ideas with classmates, but should not share code or reuse data. You may use generative AI tools if you find them helpful, but please clearly document how they were used as described above and follow the guidelines in the syllabus for what you **must** include when using generative AI. If you use generative AI and do not report it, you may receive a 0 for the assignment. You take full responsibility for the deliverables you submit.

You are welcome to use code snippets from examples in class, things you find online, or from AI code generation tools. Just make sure to give proper attribution to code you did not write. Follow the syllabus instructions for how to report the use of AI tools. However, you may not copy code that does the entire assignment (e.g. someone who did this assignment in a previous semester).