Project 1 – Image Classification using Convolutional Neural Networks

Due 03/01/2022 by 11:59pm

1. Overview & Task Description

Image classification is a fundamental task in computer vision. It has helped fuel major advancements in medicine, robotics, manufacturing, and autonomous vehicles. Considering this, your task to develop and fine-tune a convolutional neural network (CNN) for image classification using the CIFAR-10 dataset.

The CIFAR-10 dataset is widely used in the machine learning and computer vision community for benchmarking. It contains 60,000 images (50,000 for training and 10,000 for testing) with an image size of $32 \times 32 \times 3$. The dataset includes the following 10 classes: **airplane, frog, horse, automobile, bird, cat, dog, deer, ship, and truck.** More details on the dataset can be found at cifar10 | TensorFlow Datasets.

2. Programming Requirements

For this project, you will need to implement a baseline CNN as can be seen in Table 1.

Table 1. Baseline CNN architecture.

Layer	# Filters/neurons	Filter size	Activation Function
Conv Layer 1	32	3x3	relu
Max pooling	-	2x2	-
Conv Layer 2	64	3x3	relu
Max pooling	-	2x2	
Conv Layer 3	128	3x3	relu
Conv Layer 4	256	3x3	relu
Dense Layer 1	2048	-	relu
Dense Layer 2	2048	-	relu
Output Layer	10	-	softmax

It is important to note that the size of images in CIFAR-10 is $32 \times 32 \times 3$, so you will have to take that into account in the first layer of your network. Once you have the baseline network implemented, you will need to train and test your network using the following hyper-parameters.

1. Number of epochs – 25 (on a Surface Pro (CPU), each epoch takes ~1minute)
2. Optimizer - Adam
3. Loss function – SparseCategoricalCrossEntropy
4. Metrics – SparseCategoricalAccuracy
5. Learning rate – 0.001
6. Batch size – 128

Given this baseline implementation, you should achieve an accuracy around 70% for the testing data. Consider this your baseline accuracy that you want to improve. The next part of the project involves fine-tuning your network to improve this accuracy. Currently, state of the art is over 99% accuracy on the CIFAR-10 dataset. You do NOT have to achieve this. The purpose of this project is for you to see how changing hyper-parameters impacts network performance.

You will need to run the following experiments (Make these changes)

1. Number of epochs – 1, 5, 10, 15, 25 (Run these experiments with Adam optimizer only)
2. Optimizer – Adam, SGD, RMSprop,
3. Loss function– SparseCategoricalCrossEntropy, Poisson, KLDivergence
4. Learning rate – 0.1, 0.01, 0.001,
5. Batch size – 128, 256, 512
6. Regularization (two dense layers) – dropout (0.2), l1 regularization, l2 regularization (We did not cover l1 or l2 regularization in class. You can find a lot of details about how to do this online.)

Run experiment #1 with the baseline network (just change the number of epochs). Use the number of epochs with the highest accuracy to run experiments 2-6. You should budget around 10-15 hours for experiments for this project (although it could take more/less depending on your hardware).

It is important that you save the output of each experiment (each time you fine-tune) as you will need to include these details in your paper (see Section 3 for more details on this). You should save the loss and accuracy of the test set for each experiment. For each of these, it is important to make sure you know which experiment goes with each loss.

3. Paper Requirements

For the paper, you will need to turn in a 4-page plus reference paper in the IEEE double column format. The 4-pages plus references is a hard limit. You can NOT have any content, other than references, on the 5th page. You can find Latex and Word templates on Canvas. Do not change the templates (e.g., font size, etc.) This paper needs to be written like a scientific article that would be submitted to a conference so they all should have the same format. The same sections/ideas will apply in this paper, and you need to at least have the following sections (you can add more if needed).

1. Abstract (~150 words)
    a. What the problem is
    b. How you solved it
    c. What your results are
    d. Impact of work can be included here as well
2. Introduction

a. “Deeper abstract”
b. Talk about what you are doing (image classification)
c. Why is it important?
d. What did you do to solve the problem (e.g., implemented and fine-tuned a CNN)?
e. What are the major contributions of your work (e.g., experiments ran)

3. Related works (related works can also be combined with intro)
    a. Benchmarks and papers that use CIFAR-10 - CIFAR-10 on Benchmarks.AI
    b. Give a brief overview of at least 5 papers that use CIFAR-10
4. Method
    a. This is an overview of your CNN architecture
    b. Need figure that shows the baseline architecture
5. Experiments and results
    a. Go over/list all experiments that you ran
    b. Gives details about your lowest accuracy and highest accuracy. Why do you think you got those?
    c. Make a table that compares against 5 state-of-the-art works that use CIFAR-10. Show your best accuracy against 5 others. These need to be different than the works you talk about in related works. You can use *any* of the works from link in point 3 above. Make sure you list them in the references and cite them in the table. You table can have two columns. Left column has authors name e.g. Uddin et al [1], and right column can have accuracy. Make sure to have the top row with a title for each column (e.g., method and accuracy).
    d. Show figures comparing loss and accuracy compared to hyper-parameters. For example, the y-axis of a graph can be accuracy and the x-axis can be number of epochs. Plot a line graph of results. Note: for this one, you can make figures that are the most interesting to you. You will be limited by space, but you need to show at least two figures. Make sure to have a descriptive caption and discuss and reference the figure in the text (e.g., as can be seen in Figure 1…)
6. Conclusion
    a. Concise overview of paper and what you did
7. References
    a. At least 10 references needed (at least 5 in related works and 5 comparisons). Note all references must be cited in the paper. Do not include a reference that does not have a citation in the paper. In the word document, there are some references. Make sure to remove these. They are not related to the work. Reference must be in IEEE format as well. The following is an example in IEEE format.

        Md T. Uddin and S. Canavan, Quantified facial expressiveness for affective behavior analytics, WACV, 2022.

4. Grading

Paper – 60% of grade

1. Abstract – 5% of grade
2. Introduction – 5% of grade
3. Related works – 10% of grade
4. Method – 5% of grade
5. Experiments and Results – 30% of grade
6. Conclusion – 5% of grade.

Code – 40% of grade

1. Correct baseline architecture with highest accuracy when your script is run. This will result in full credit for code portion of grade. If there are errors and your code does not run correctly, the most you can get is 20% for the code portion of your grade.

5. Deliverables

1. Python script(s) that have your best model. This is the final architecture/tuning that gave you the highest accuracy.
2. PDF of your paper detailing the project.
3. Create directory called <lastName_Project1> and place scripts and PDF in this directory.
4. Zip directory up and upload to Canvas.