Chennai Mathematical Institute
Computer Vision
April 11, 2020

Due Date: Sunday, April 19, 2020 at 10:00PM

Final Project – Part 1
Image Classification with Convolutional Neural Nets

Instructions:

1. Integrity and collaboration: Students are encouraged to work in groups, but each student must submit their own work. If you work as a group, include the names of your collaborators in your write-up. Code should NOT be shared or copied. Please DO NOT use external code unless permitted. Plagiarism is strongly prohibited and may lead to failure of this course.

2. Start early! Training, evaluating and fine-tuning neural networks is a time-consuming task. If you start late, you will not have enough time to achieve good results.

3. If you have any question, please post your question on Piazza. Other students may have encountered the same problem, and it helps to keep communication both efficient and uniform.

4. Write-up: Items to be included in the write-up are summarized in the Write-up section. Please note that we DO NOT accept handwritten scans for your write-up in this assignment.

5. Late Policy: The assignment will not be accepted past 96 hours beyond the specified due date. You will lose 1 point for each hour you are late beyond the deadline for the first 6 hours. Beyond that, the delay will be counted in day units with a penalty of 10 points for each day you are late.

Overview

This project is an introduction to deep learning tools for computer vision. You will design and train deep convolutional network for image classification using Keras and TensorFlow, an open source deep learning platform.

In programming assignment 4, you worked on image classification using K-nearest neighbor and SVM classifiers for a subset of images from the CalTech-101 dataset. In this final project, you will build and train a deep neural networks to work on the Fashion MNIST dataset.

The Fashion MNIST dataset contains 70,000 grayscale images in 10 categories. The images show individual articles of clothing at low resolution (28 by 28 pixels), as seen here:

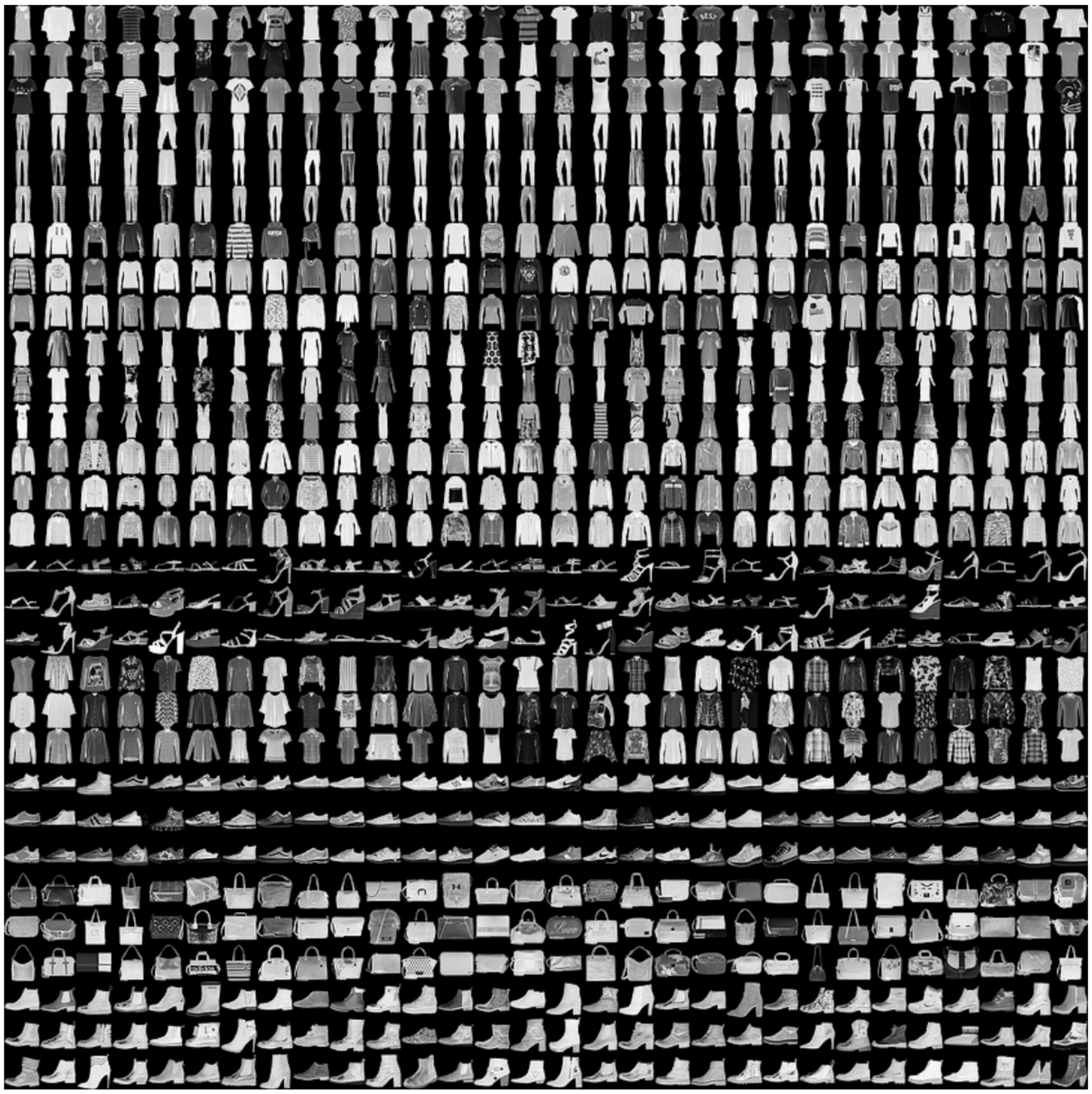


Figure 1. Fashion-MNIST samples (by Zalando, MIT License).

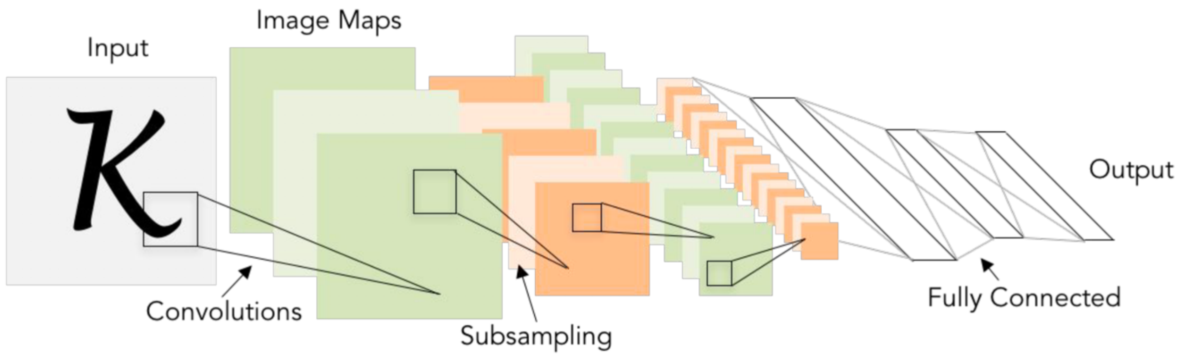
The Fashion-MNIST dataset is similar to the MNIST dataset in image size (28x28), number of classes (10), and the size of the training and test datasets (60,000 and 10,000 images, respectively).

Your task is to build and train a Convolutional Neural Network using the training set, and evaluate the model using the test set.

Detailed Instructions:

1. The dataset can be accessed in Keras as `keras.datasets.fashion.mnist`

2. Your ConvNet model should include at least one convolutional layer. You can use the LeNet model used for MNIST (shown below) as a baseline



Conv filters were 5x5, applied at stride 1
Subsampling (Pooling) layers were 2x2 applied at stride 2
i.e. architecture is [CONV-POOL-CONV-POOL-FC-FC]

In the following, k refers to kernel size, s is the stride, and p is the padding.

- Input – 28 x 28 x 1
- Conv_1 – $k = 5; s = 1; p = 0$; depth = 20
- ReLU
- MAX Pooling - $k = 2; s = 2; ; p = 0$
- Conv_2 – $k = 5; s = 1; p = 0$; depth = 50
- ReLU
- MAX Pooling - $k = 2; s = 2; ; p = 0$
- Fully Connected layer - 500 neurons
- ReLU
- Classification layer – 10 neurons

3. You may add or subtract convolution or pooling layers, change the kernel size, stride or padding.

4. You can change the depth (i.e., number of filters used in any of the convolutional layers).

5. You may use data augmentation or Dropout to overcome overfitting.

6. Choose your optimizer and the relevant parameters, such as learning rate, and momentum. See https://keras.io/optimizers/

7. Choose your loss function. For classification where the target value is an integer, use the "*sparse_categorical_cross_entropy*" loss function. See https://keras.io/losses/

8. Plot the training and test accuracy, as well as training and validation loss values.

- It is fairly easy to achieve an accuracy of 90%. So, play with the dials and see what you are able to achieve.

- Keep your number of trainable parameters under 120,000. Training can take too long, otherwise.

What to submit:

1. Submit your Google Colab notebook with your best performing network, and any other experiments relating to the project that you want to share. See Muthuvel's instructions on Piazza regarding what to specify in your submission on Moodle.

2. A report detailing your network architecture, number of trainable parameters, your experiments, findings, and choices for your final architecture, and results. Your report should include your testing accuracy plots, and your interpretation.

Good Luck!