ENEE633/CMSC828C

Project 2 **Due 12/17/2021**

Introduction

The aim of this project is to implement different conventional classifiers to achieve hand-written digits recognition, as well as performing transfer learning in an image classification task. The whole pipeline is described in the following sections.

Part I: Hand-written Digit Recognition

SVM Implement linear and kernel SVM on MNIST dataset. You have to try different kernels (linear, polynomial, RBF) and compare results in your report. You can use any online toolbox for this, e.g. LIBSVM (`https://www.csie.ntu.edu.tw/~cjlin/libsvm/`) or any MATLAB built-in function. You can apply PCA and LDA here for dimensionality reduction.

Logistic Regression Read Chapter 7.6 of the textbook and implement multiclass logistic regression for MNIST. You should try both PCA and LDA for dimensionality reduction, first.

In brief, multiclass logistic regression models the class posteriors of an image log-linearly in terms of its features so that $P(y_{m,n} = 1|\mathbf{x}_n) = \frac{\exp(\boldsymbol{\theta}_m^\top \mathbf{x}_n)}{\sum_{i=1}^{M} \exp(\boldsymbol{\theta}_i^\top \mathbf{x}_n)}$, i.e. the softmax function, where $y_{m,n}$ is the indicator of $\mathbf{x}_n$ being in class ω_m. The model parameters are estimated to minimize the negative log-likelihood of the class labels, which are encoded as categorical vectors. The negative log-likelihood takes the form $\ell(\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_M) = -\sum_{n=1}^{N} \sum_{m=1}^{M} y_{m,n} \ln \left(\frac{\exp(\boldsymbol{\theta}_m^\top \mathbf{x}_n)}{\sum_{i=1}^{M} \exp(\boldsymbol{\theta}_i^\top \mathbf{x}_n)} \right)$; this is sometimes called the cross-entropy loss function.

Deep Learning Build a Convolutional Neural Network. Train it on MNIST training set and test it on testing set. You can design your architecture or use the architecture introduced in LeCun's paper [1]. You can use any of the following toolboxes:

Caffe If you are Linux or OSX user, you can follow the instruction from `http://caffe.berkeleyvision.org` to install it.

LightNet[2] It is a pure MATLAB toolbox. If you are a Windows user, this is an alternative choice. You can download it from `https://github.com/yechengxi/LightNet` and the tutorials are also included.

Deep Learning Toolbox It is a pure MATLAB toolbox. If you are a Windows user, you can also try this. You can download it from `http://www.mathworks.com/matlabcentral/fileexchange/38310-deep-learning-toolbox`. However, the original version may not be enough to build your CNN, so you can use the supplementary package posted (DeepLearningToolboxSupplementary.zip).

Dataset You can download **MNIST** from `http://yann.lecun.com/exdb/mnist/`. The description of Dataset is also on the website. Basically it has a training set with 60000 28 x 28 grayscale images of handwritten digits (10 classes) and a testing set with 10000 images.

Part II: Transfer Learning

Transfer Learning For this part of the project, you will be applying a deep learning technique commonly referred to as Transfer Learning. Specifically, Transfer Learning can be used for deep learning applications where either data or computational power are restricted. Here you are given a data set with ten classes (ten different monkey species) with only 140 images per class. The first task will be to train a simple convolutional neural network using these images (a very simple 3-5 convolutional network will suffice) and test the accuracy of the model using the validation set. Because of the low number of training samples, you will see that the test accuracy of the model will be lower than expected. This is where transfer learning can help. The next task will be to download a pretrained model for image classification (for example VGG, ResNet, InceptionNet) and use it as a feature extractor. To do this, you will remove the last fully connected layers of the pretrained model and replace it with untrained fully connected layers to classify the monkey species. During training, you will freeze the convolutional layer parameters so that they remain the same and only update the fully connected layers at the end. In this way, the convolutional layers act as generalized feature extractors that have already been pretrained on millions of other images (that werent necessarily all monkeys) while the fully connected layers are able to take these features and classify our images. You should see a substantial boost in accuracy even though we have the same amount of training samples. To further boost the performance of the network, you can unfreeze the convolutional layers and train for a few more epochs with a small step size to fine tune the network to extract even more predictive power.

Dataset You can download the dataset from here: `https://www.kaggle.com/slothkong/10-monkey-species/home`.

Submission Guidelines

In this project, you can use any programming language you want, such as Matlab(recommend), C++, Python, ...etc. You need to submit your project in the following structure: A zip file with the name **YourDirectoryID_Proj2.zip** on to ELMS/Canvas. A main folder with the name **YourDirectoryID_P2** and the following sub-folders/files:

Code A folder contains all your code

Report Your report should be in pdf format and no more than 5-6 pages(excluding figures). You should concentrate on the employed methods, the experiments and discuss the result you have obtained.

Readme A .txt file describing how to run your code.

References

[1] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, Nov 1998.

[2] Chengxi Ye, Chen Zhao, Yezhou Yang, Cornelia Fermüller, and Yiannis Aloimonos. Lightnet: A versatile, standalone matlab-based environment for deep learning. In *Proceedings of the 2016 ACM on Multimedia Conference*, MM '16, New York, NY, USA, 2016. ACM.