CS464 Introduction to Machine Learning
Fall 2023
Homework 2

Due: December 13, 2023 23:59

Instructions

- For this homework, you may code in any programming language of your choice.
- You are NOT allowed to use any machine learning packages, libraries or toolboxes for this assignment (such as scikit-learn, tensorflow, keras, theano, MATLAB Statistics and Machine Learning Toolbox functions, e1071, nnet, kernlab etc.) unless otherwise stated.
- Submit a soft copy of your homework to Moodle.
- Upload your code and written answers to the related assignment section on Moodle (.TAR or .ZIP). Submitting hard copy, handwritten or scanned files is NOT allowed.
- The name of your compressed folder must be "`CS464_HW2_Section#_Firstname_Lastname`" (i.e., `CS464_HW2_1_sheldon_cooper`). Please do not use any Turkish characters in your compressed folder name.
- Your code should be in a format that is easy to run and must include a driver script serving as an entry point. You must also provide a README file with clear instructions on how to execute your program.
- This is an individual assignment for each student. That is, you are NOT allowed to share your work with your classmates.
- If you do not follow the submission routes, deadlines and specifications (codes, report, etc), it will lead to significant grade deduction.
- If you have any questions about the questions, you can contact:
  - a.yildirim@bilkent.edu.tr for Q1
  - kubra.caglar@bilkent.edu.tr for Q2

1 PCA Analysis [50 pts]

In this question, you are expected to analyze the MNIST [1] dataset using PCA. Since the dataset website requires authorization, the dataset files and a script to read these files properly are provided on Moodle. The dataset consists of 70,000 grayscale digit images, 60,000 being the training data and the remaining 10,000 being the test data. The images have 28x28 pixel resolution and their corresponding digit label are provided in the dataset. **For this part of the assignment, you will be using only the training data**. The corresponding dataset files are as follows:

- `train-images-idx3-ubyte.gz`
- `train-labels-idx1-ubyte.gz`

For this question, the use of any library for PCA calculations is not allowed. You are requested to implement the PCA algorithm by yourself. It is allowed to use the `numpy.linalg.eig` function to find the eigenvalues and the eigenvectors in your calculations.

Before the analysis, flatten all images of size 28×28 to get a 784 dimensional vector for each image. The shape of the final data should be $60,000 \times 784$, where $60,000$ is the number of samples.

Question 1.1 [15 pts] Apply PCA on the dataset to obtain the principal components. Report the proportion of variance explained (PVE) for the first 10 principal components and discuss your results.

Question 1.2 [5 pts] Report at least how many of the principal components should be used to explain the 70% of the data.

Question 1.3 [10 pts] Using the first 10 principal components found in Question 1.1, reshape each principal component to a 28×28 matrix. Apply min-max scaling to each principal component to set the range of the values to $[0, 1]$ so that the principal components can be visualized. After scaling, display the obtained grayscale principal component images of size 28×28. Discuss your results.

Question 1.4 [10 pts] Project the first 100 images of the dataset onto the first 2 principal components. Plot the projected data points on the 2-D space by coloring them according to the labels provided in the dataset. Label the axes by the index of their corresponding principal components. Each digit label should be colored with a different color, 10 colors in total. Discuss the distribution of the data points according to their labels by considering the visuals of the first 2 principal components found in Question 1.3.

Question 1.5 [10 pts] Describe how you can reconstruct an original digit image using the principal components found in Question 1.1. Use first k principal components to analyze and reconstruct the first image[1] in the dataset where $k \in \{1, 50, 100, 250, 500, 784\}$. Discuss your results.

Hint 1: Do not forget to use the mean values in the reconstruction process, which you subtracted from the data to calculate the principle components. (Question 1.5)

Hint 2: For the single channel images, the default colormap that the Matplotlib library uses is different than grayscale. To make it suitable for the MNIST images, you can use "cmap='Greys_r'" for the imshow function. (Question 1.3 & Question 1.5)

[1] The index of the image in the training dataset is 0.

2 Logistic Regression [50 pts]

For this question, you are asked to develop a Multinomial Logistic Regression Classifier model to classify digit images extracted from the MNIST database. As mentioned in the previous question, the MNIST database is a large collection of handwritten digits. It comprises 70,000 examples, 60,000 being the training data and the remaining 10,000 being the test data. The hand-written digits are size-normalized and centered in a 28x28 image. Since the dataset only contains training and test data, you must create your own validation dataset by separating the first 10,000 images from your training data and their corresponding labels. Ultimately, you will have 50,000 training, 10,000 test, and 10,000 validation images. Different from the first question, you will additionally use test images and labels in this question. You are provided with a script to upload and read this data. Please check the script for Question 2 given in Moodle. The corresponding files are as follows:

- `train-images-idx3-ubyte.gz`
- `train-labels-idx1-ubyte.gz`
- `t10k-images-idx3-ubyte.gz`
- `t10k-labels-idx1-ubyte.gz`

Since you are asked to implement multinomial classification, you need to turn the labels into their one-hot-encoded version and initialize a weight matrix with the bias terms, which outputs 10 predictions (corresponding to each label) when multiplied with the input vector. Also, unlike in the Binomial Logistic Regression, you will use Softmax as the activation function instead of Sigmoid. The formula for Softmax activation function is provided in Equation 2.1, where z is the input vector, i is the index of the element in the output vector, and K is the total number of classes.

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{2.1}$$

Following this formulation, the update rule for weights is based on the derivative of the cross-entropy loss. It can be calculated using the difference between the target values and softmax outputs. For a detailed mathematical derivation of softmax, please visit the link provided[2]. While updating the weights of your model, remember to add the $L2$ regularization term given in Equation 2.2. You need to take its derivative and combine it with the gradient in your loss formula.

$$L2_{\text{reg}} = \frac{\lambda}{2} \sum_{i=1}^{N} w_i^2 \tag{2.2}$$

As in the first question, you need to flatten your images of size 28x28 to get a 784 dimensional vector for each image. Also, in the dataset, feature scales are significantly different from each other. You need to normalize the data to train a model not influenced by feature scales. You can apply min-max normalization to scale the features in the range [0,1]. Formulation of this normalization is provided in Equation 2.3. Since images consist of arrays of integers ranging from 0 to 255, X_{min} here is 0, and X_{max} is 255. So, according to Equation 2.3, you need to divide your flattened image arrays into 255 to normalize them. The normalization is already applied in the script provided in Moodle.

$$\hat{x} = \frac{x - X_{min}}{X_{max} - X_{min}} \tag{2.3}$$

Note: Use the same data split in all parts of the assignment to perform a fair split between classifiers for parameter selection. Also, you should train each model for 100 epochs for your experiments

Hint: Try to use as much as vectorized operations using numpy instead of for loops.

[2] https://peterroelants.github.io/posts/cross-entropy-softmax/

Implement a Logistic Regression Classifier for the aforementioned task. For this part, initialize your weights from a Gaussian distribution, where the weights are initialized as $\mathcal{N}(\mu = 0, \sigma = 1)$. Also, you should initialize the batch size as 200, the learning rate as 5x10^{-4}, and the L2 regularization coefficient (λ) as 10^{-4}, this will be your default model. Afterward, you will experiment with these hyperparameters to find the best model. While doing your experiments, you will only change the requested hyperparameters and keep the others as their mentioned default values.

Question 2.1 [15 pts] Train the default model described above. Display the test accuracy and confusion matrix for that case.

Question 2.2 [15 pts] For this part of the question, you will do separate experiments on the hyperparameters mentioned. Remember, you only need to change the mentioned hyperparameter types according to the given values and keep the other default ones. You will change one hyperparameter at a time. Try your model using the hyperparameters given below and compare their performances:

- Batch size: 1, 64, 50000
- Weight initialization technique: zero initialization, uniform distribution, normal distribution
- Learning rate: 0.1, 10^{-3}, 10^{-4}, 10^{-5}
- Regularization coefficient (λ): 10^{-2}, 10^{-4}, 10^{-9}

To exemplify, you need to evaluate your model performance based on batch sizes 1, 64, and 50000 by keeping other default hyperparameter values. After you run your model with these values, you need a graph having epochs at the x-axis and resulting accuracies at the y-axis. You need to use legends to show the individual performances of given batch sizes. You should perform this procedure for each hyperparameter given above. Write the titles accordingly.

Question 2.3 [5 pts] After you perform the above experiments, you need to select the best values for each of the hyperparameters (and the best-performing initialization technique for weights) and create the optimal model. You need to display the test accuracy and confusion matrix for the best model.

Question 2.4 [10 pts] As mentioned in the earlier parts of this section, you have initialized 10 (number of labels) weight vectors for your classification task. In this part, you need to visualize your finalized weight vectors (after your best model is trained) and print them as images. One line of code is provided for you to visualize your weights; please check the script that you are given in Moodle. Keep in mind that we expect some blurriness in the finalized weight images. After you obtain your results, comment on their look and what they might represent.

Question 2.5 [5 pts] Using the best model, calculate precision, recall, F_1 score and F_2 score for each class. Comment on the results using the confusion matrix you obtained in Question 2.3 and the weight images you obtained in Question 2.4.

References

[1] Y. LeCun, C. Cortes, and C. Burges, "Mnist handwritten digit database," *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, vol. 2, 2010.