Assignment 2: EN4553 - Deep Learning for Vision

University of Moratuwa

September 18, 2024

1. Briefly explain the numpy broadcasting rules. Mention the output shapes of these operations with the given input array shapes; mention "Error" if an operation are not supported.

Operation	`a.shape`	`b.shape`
`a + b`	(256, 256, 3)	(3,)
`a - b`	(10, 1, 6, 1)	(9, 1, 7)
`a * b`	(2,)	(4,)
`a / b`	(4, 1)	(8 , 5, 3)
`a - b`	(5, 3, 2)	(5, 1, 2)
`a - b.mean()`	(128, 128, 3)	(256, 256, 5)
`a - b.mean(axis=(1, 2))`	(3, 256, 256)	(3, 256, 256)
`a - b.mean(axis=(1, 2), keepdims=True)`	(3, 256, 256)	(3, 256, 256)
`np.matmul(a, b)`	(6, 5, 3)	(3, 4)

2. Given two design matrices: X of shape $m \times d$ and Y of shape $n \times d$, where each row of each matrix represents a d-dimensional data point, write a Python function to compute the pairwise squared Euclidean distances between the rows of the two matrices. The output of your function should be an $m \times n$ matrix Z with $Z_{i,j} = \|\mathbf{x}_i - \mathbf{y}_j\|^2$, where $\mathbf{x}_i$ and $\mathbf{y}_j$ are the i^{th} and j^{th} rows of X and Y, respectively. Use only numpy operations (higher-level libraries such as SciPy are not allowed). Your implementation should be efficient, therefore `for`/`while` loops must not be used.

3. In this problem, you will develop a model to identify objects classes in the Caltech-101 Dataset. You could either use the raw images available at https://doi.org/10.22002/D1.20086, or the processed version in https://www.tensorflow.org/datasets/catalog/caltech101. This dataset has 101 different object classes. Given an image, your model will output its label. If you use the raw dataset, randomly split it into a training set and a test set, with approximately two-thirds of the images in the training set and one-third in the test set. If you use the Tensorflow datasets version, use the provided train and test partitions. The test partition is only used to report the final results, it should not be used to train the network or tune the hyperparameters. You can use any Python-based deep learning framework such as Tensorflow, PyTorch, or JAX to solve this problem. Specific instructions are intentionally omitted to give students freedom to decide how they want to solve the problem.
   (a) Take an Imagenet pre-trained network such as ResNet-50. Do k-NN classification with the embeddings produced by the pre-trained network. You may use the function you wrote in the previous question as a utility function. Report the accuracy on the test set.
   (b) Perform linear classification (multi-class logistic regression) with the pre-trained embeddings and report the accuracy on the test set. (Note that this is equivalent to training only the last fully-connected layer of a classification network with the feature extractor frozen).
   (c) Fine-tune an image classification network end-to-end (while changing the pre-trained weights as well). Report the accuracy on the test set. Use data augmentation and other methods you learned in the class to obtain good test accuracy.

Deliverables: Upload a single .zip file containing a PDF file and your Python source code files (.py files). The report should not exceed three A4 pages and should not include unreasonably small or large fonts. You may include snippets of your code, plots produced, etc. in the PDF. Theoretical details of the methods you use are not expected. Each group should do only one submission with the names and the index numbers of all members on the cover page. Late submissions will not be accepted.