**Department of Artificial Intelligence**

**College of Computer and Cyber Sciences**

**Introduction to Deep Learning**

***Fine-Tuning a Pre-Trained Model in PyTorch***

1. **Learning Objectives**

By the end of this lab, students will:

* Learn what is a fine-tuned model.
* Be able to use a fine-tuned ResNet model for MNIST Classification.
* Differentiate between a pre-trained model and a fine-tuned model.

1. **Explanation of Key Concepts**

* **Fine-Tuning:**

Fine-tuning in deep learning refers to the process of adapting a pre-trained model to perform better on specific tasks or datasets. This technique has become essential, especially when working with foundation models in generative AI. By starting with a model that has already learned general patterns from large datasets, fine-tuning allows for more efficient training and better performance on specialized tasks.

* **ResNet Model:**

ResNet (Residual Network) is a deep convolutional neural network (CNN) that introduced the concept of residual connections, enabling the training of much deeper networks. By utilizing these connections, ResNet alleviates the vanishing gradient problem, allowing for more effective learning in networks with hundreds or even thousands of layers. Trained on large datasets like ImageNet, ResNet has become a powerful model for image classification and other computer vision tasks.

1. **Activities**

* **Exercise 1: Using** **ResNet for MNIST Classification.**

In this exercise, we will fine-tune a pre-trained ResNet model for image classification on the MNIST dataset using PyTorch. You will load the model, modify its architecture to match the number of classes in MNIST, preprocess the dataset, train the model, and evaluate its performance.

|  |
| --- |
| # Code provided in the notebook |

Screenshot of the result:

|  |
| --- |
|  |

1. **Tasks**

* **Task 1:**

In this task, you will modify the code from Exercise 1 to fine-tune ResNet on Cifar10 dataset instead of MNIST.

Submit your jupyter notebook & Add a screenshot of the result:

|  |
| --- |
|  |

* **Task 2:**

Complete the table below by identifying the key differences between **pre-training** and **fine-tuning**.

(Using ChatGPT or any other Chatbot is completely prohibited).

|  |  |  |
| --- | --- | --- |
| **Feature** | **Pre-Training** | **Fine-Tuning** |
| **Definition** | A model that has already been trained on a large set of data insuring generalization on most | Adjusting the parameters and operations of the existing model so that we reach the best wanted result |
| **Dataset** | For the pre-trained you need a data set that is large to make a model that can be used for many other set sizes like image net (1000 classes) | You need a dataset with reasonable and relatively small and manageable set for the purpose of getting your results and checking your progress like MNIST (10 classes) |
| **Training Time** | The training time would be huge because of the dataset size and the full training needed | Should be relatively small in comparison |
| **Example** | ResNet Trained on Imagenet | ResNet Trained on MNIST |

1. **References**

[What is Fine-Tuning? | IBM](https://www.ibm.com/think/topics/fine-tuning)

[Fine tuning Vs Pre-training. The objective of my articles is to… | by Eduardo Ordax | Medium](https://medium.com/@eordaxd/fine-tuning-vs-pre-training-651d05186faf)

[Residual Networks (ResNet) - Deep Learning - GeeksforGeeks](https://www.geeksforgeeks.org/residual-networks-resnet-deep-learning/)

[Fine-Tuning a Pre-Trained Model in PyTorch: A Step-by-Step Guide for Beginners - DEV Community](https://dev.to/santoshpremi/fine-tuning-a-pre-trained-model-in-pytorch-a-step-by-step-guide-for-beginners-4p6l)