ASSIGNMENT (CONVOLUTIONAL NEURAL NETWORKS)

NAVTTC PROGRAM

ARTIFICIAL INTELLIGENCE (MACHINE & DEEP LEARNING)

Instructor: Azam Rashid

Submission:

- Make a Google Collab notebook Or VS Code Editor (Jupiter Extension) to implement this Assignment.
- Submit a .ipynb file detailing all the information. No other format will be accepted
- Submission file should be named as Assignment_CNN_Students-BatchNo.ipynb
- Deadline for this Mid-Term Project Assignment is Monday 25-09-2024.
- Strictly follow the submission deadline.
- Make Submission in the Midterm Project Assignment on Google Form and press the submit button.
- Click here to submit the Project Assignment

Solve the Following Task

Question 1: Design and implement a basic Convolutional Neural Network (CNN) architecture from scratch to classify images in the CIFAR-10 dataset. Your CNN should include at least two Convolutional layers, two pooling layers, and two fully connected layers. **(Marks 20)**

Instructions:

- Use the CIFAR-10 dataset which contains 60,000 32x32 color images in 10 different classes.
- The dataset can be directly imported using Tensor Flow or Keras libraries.
- Two convolutional layers with ReLU activation functions and appropriate filters (e.g., 32 and 64).
- Two max-pooling layers to reduce the spatial dimensions.
- Flatten the output of the last pooling layer.
- Two fully connected layers (dense layers) with ReLU activation.
- A final output layer with 10 neurons and softmax activation for classification.
- Compile the model using an optimizer like Adam and a loss function such as categorical crossentropy.
- Train the model on the CIFAR-10 training set.
- Use validation data to monitor the training process and adjust parameters if necessary.

- Evaluate the model on the test set and report accuracy, loss, and confusion matrix.
- Visualize the training and validation accuracy/loss curves.

Question 2: Utilize a pre-trained ResNet50 model for image classification and fine-tune it on the Cats and Dogs dataset from Kaggle. **(Marks 20)**

Instructions:

- Download the Cats and Dogs dataset from Kaggle. The dataset contains images of cats and dogs in separate folders.
- Split the dataset into training and validation sets.
- Load the ResNet50 model pre-trained on ImageNet.
- Freeze the initial layers of the model to retain the learned features.
- Replace the top layers of the model with custom layers suitable for binary classification.
- Fine-tune the model on the Cats and Dogs dataset by unfreezing some of the deeper layers.
- Use an optimizer like SGD with a low learning rate for fine-tuning.
- Evaluate the fine-tuned model on the validation set using accuracy, precision, and recall.
- Compare the performance with the base ResNet50 model.

Dataset: https://www.kaggle.com/c/dogs-vs-cats/data

Resources:

https://youtu.be/JcU72smpLJk?si=3x1BBOso88VtFilh

https://pytorch.org/vision/stable/models.html

https://huggingface.co/microsoft/resnet-50

https://youtu.be/0MVXteg7TB4?si=z3xHL92i802fJvOI

https://youtu.be/PN4asCDITNg?si=DZW_brJKmqAZtORhttps://

vtiya.medium.com/why-we-freeze-some-layers-for-transfer-learning-f35d9f67f99c

https://youtu.be/n-PgCuD2DNg?si=esV7LLeK1HS288VM

https://youtu.be/V1-Hm2rNkik?si=iARdeV02mAk3-R-s

Question 3: Apply CNNs to detect and classify objects in images using the Pascal VOC 2012 dataset from Kaggle. **(Marks 20)**

Instructions:

- Download the Pascal VOC 2012 dataset from Kaggle.
- Preprocess the images by resizing them to a consistent size (e.g., 300x300) and normalizing pixel values.
- Use a pre-trained object detection model like SSD or Faster R-CNN.
- Customize the model to detect objects in the Pascal VOC dataset.
- Train the model on the training set of the Pascal VOC dataset.
- Monitor performance using mAP (mean Average Precision) and adjust hyperparameters as needed.
- Evaluate the model on the test set using metrics like mAP and Intersection over Union (IoU).
- Discuss the challenges of object detection and possible improvements.

Dataset: https://www.kaggle.com/datasets/gopalbhattrai/pascal-voc-2012-dataset

Question 4: Develop an image segmentation model using U-Net architecture to segment medical images from the Brain MRI Segmentation dataset. **(Marks 20)**

Instructions:

- Download the Brain MRI Segmentation dataset from Kaggle.
- Preprocess the images and masks by resizing them to a consistent size (e.g., 128x128) and normalizing the pixel values.
- Implement the U-Net architecture for image segmentation.
- Use appropriate layers such as convolutional, pooling, upsampling, and skip connections to capture both local and global features.
- Train the U-Net model on the training set of the Brain MRI Segmentation dataset.
- Use Dice coefficient as the loss function and monitor the performance on the validation set.
- Evaluate the model on the test set using segmentation metrics like Dice coefficient and IoU.
- Visualize the segmentation results by overlaying the predicted masks on the original images.

Dataset: https://www.kaggle.com/datasets/mateuszbuda/lgg-mri-segmentation

Question 5: Apply CNNs to classify X-ray images as normal or pneumonia using the Chest X-ray Images dataset from Kaggle. **(Marks 20)**

Instructions:

- Download the Chest X-ray Images dataset from Kaggle.
- Preprocess the images by resizing them to a consistent size (e.g., 224x224) and normalizing pixel values.
- Design a CNN architecture for binary classification. You may use architectures like VGG16 or a custom CNN.
- Use binary cross-entropy as the loss function and an optimizer like Adam.
- Train the model on the training set of the Chest X-ray Images dataset.
- Monitor the performance using validation accuracy and adjust hyperparameters as needed.
- Evaluate the model on the test set using accuracy, precision, recall, and F1-score.
- Discuss the impact of using CNNs for medical diagnostics, including the potential benefits and challenges.

Dataset: https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia