TASK OVERVIEW

DISTRIBUTED IMAGE RECOGNITION

Create an advanced neural network model capable of identifying various medical conditions from image data, enhancing the accessibility and reliability of early disease detection.

By : pewds

MOTIVATION

FOLDING@HOME

Distributed computing project aimed to help scientists develop new therapeutics for a variety of diseases by the means of simulating protein dynamics.

The project utilizes graphics processing units (GPUs) and central processing units (CPUs) for distributed computing and scientific research.

OS type	Active users (CPUs)
Windows	55879
Mac OS	49100
Linux	22426
Total	127405

Number of active users for Folding@home by 2018

MAIN STEPS

DATA COLLECTION

Gathering a comprehensive dataset of medical images across various conditions.

NEURAL NETWORK DESIGN

Selecting a Model Architecture such as Convolutional Neural Networks (CNNs)

MODEL TRAINING

Task Distribution: Distribute subsets of the dataset across participants' computers for training, enhancing the diversity of data the network learns from. Aggregate Insights: Systematically compile and merge model improvements from all contributors. Employ Federated Learning methods to refine the central model while maintaining data privacy.

TESTING AND VALIDATION

Post-training, assess the model against a separate test dataset not previously used in training or validation phases.

DATA COLLECTION

Obtaining a large - reliable - wide - labeled dataset is critical for the accuracy of the model not to overfit it and to get good accuracy results.

an example could be a public dataset called ISIC (International Skin Imaging Collaboration).

This dataset includes images labeled with diagnostic information, distinguishing between melanoma and benign lesions and could be used to create neural network model that can diagnose melanoma from images of skin lesions.

NEURAL NETWORK DESIGN

Selecting a proven deep learning architecture suitable for image recognition tasks is critical, and Convolutional Neural Networks (CNNs) could a starting point. Models like ResNet, Inception, or EfficientNet can serve as the foundation.

Designing the training process:
Train the model in stages, using distributed computing to handle different batches of the dataset.
Ensure the training process includes validation steps to avoid overfitting.

MODEL TRAINING

- Initial Setup:
  - Start with 1100 images intended for training a model.
  - Aim to divide these images into evenly distributed batches.
- Batch Formation:
  - Divide images into 10 batches, each containing 110 images.
  - Designate 100 images in each batch for training and 10 images for validation.
- Two Approaches Overview:
  - The processes are largely similar but focus on different objectives: one aims at optimizing model parameters, and the other emphasizes training speed.
- Approach Specifics:
  - Approach 1 (Parameter Optimization):
    - Each device receives the batch of data with the intent of fine-tuning the model parameters.
    - Model parameters are adjusted uniquely on each device to explore the parameter space efficiently.
  - Approach 2 (Training Speed):
    - Devices also receive different batches of data, but here, the model parameters are fixed across all devices.
    - This approach seeks to expedite the training process by paralleling the workload without altering the parameters per device.

MODEL TRAINING

- Training and Validation Process:
  - Each device trains the model using the 100 training images from its batch.
  - Post-training, the device validates the model using the designated 10 validation images to assess accuracy.
- Output and Feedback Loop:
  - After completing the training and validation phases, each device compiles the trained model (encoded as a string), the achieved accuracy, and the used parameters.
  - This data (model, accuracy, parameters) is then sent back as task value for aggregation or further analysis.
- Federated Learning:
  - is used to aggregate the best trained models to create a centralized model that will be used as initial model for the next round.

AUDIT SUBMISSIONS

Check if submission value for the node has a valid model with accuracy.

DISTRIBUTE REWARDS

The better accuracy the submission has the better rewards could the node get.