**Pneumonia Detection from Chest X-ray Images Using Convolutional Neural Networks**

**Group Members:** Mahmut Sami Yılmaz, Seda Yeler

**1. Problem Definition**

Pneumonia is a significant health concern worldwide, especially in children and the elderly. Early and accurate diagnosis is crucial for effective treatment and reducing mortality rates. Chest X-ray images are a common diagnostic tool for detecting pneumonia. This project aims to develop a convolutional neural network (CNN) to classify chest X-ray images as either showing pneumonia or being normal.

**2. Description of the Neural Network Model**

A convolutional neural network (CNN) is used for image classification tasks due to its efficiency in handling image data. The CNN architecture for this task is as follows:

* **Input Layer:** Accepts images of size 150x150 with 3 color channels (RGB).
* **Convolutional Layers:** Two convolutional layers with 64 and 128 filters respectively, each followed by a ReLU activation function and a max-pooling layer.
* **Flatten Layer:** Converts the 2D feature maps to a 1D feature vector.
* **Fully Connected Layer:** A dense layer with 512 neurons followed by a ReLU activation function and a dropout layer with a 50% dropout rate to prevent overfitting.
* **Output Layer:** A dense layer with 1 neuron and a sigmoid activation function for binary classification.

The model is compiled using the Adam optimizer with a learning rate of 1 x 10-6 and the binary cross-entropy loss function, with accuracy as the performance metric.

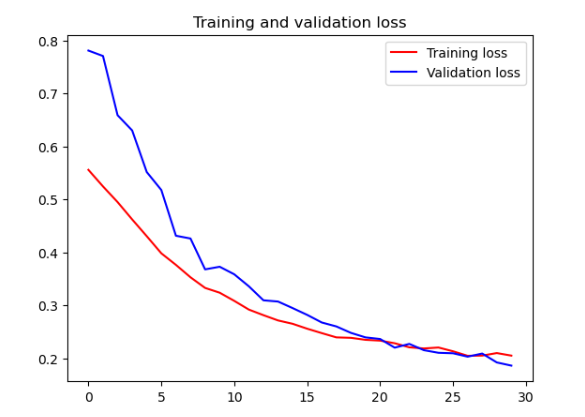
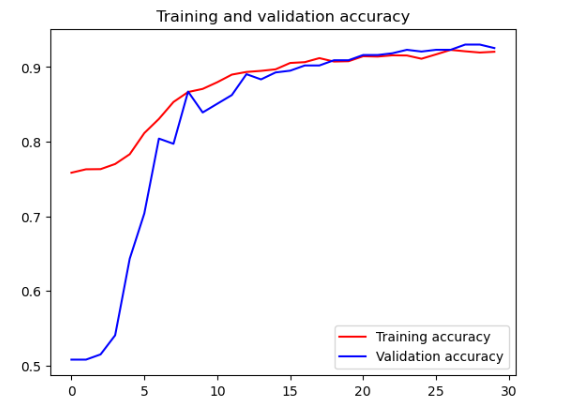
**3. Description of the Dataset and How It Was Obtained**

The dataset used for this project is the "Chest X-ray Images (Pneumonia)" dataset available from Kaggle. It contains 5,863 images belonging to two classes: 'Pneumonia' and 'Normal'.

* **Training Set:** Contains images used to train the model.
* **Validation Set:** Used to validate the model during training to tune hyperparameters and prevent overfitting.
* **Test Set:** Used to evaluate the final performance of the model on unseen data.

Data augmentation techniques, such as rescaling, shearing, zooming, and horizontal flipping, are applied to the training images to enhance the model's generalization ability.

**4. Results**

The model was trained for 30 epochs. The training and validation accuracy and loss are plotted to evaluate the model's performance over epochs.

**5. Discussion**

The results indicate that the CNN model can effectively classify chest X-ray images with a high degree of accuracy. The use of data augmentation techniques helped in preventing overfitting. The training and validation curves show that the model generalizes well to unseen data. Future work may involve further tuning the hyperparameters, trying different CNN architectures, or using transfer learning to potentially improve the model's performance.

**6. How to Use the Test Model**

In this section, we provide detailed instructions on how to use the trained model to classify chest X-ray images and evaluate its performance on a test dataset containing 100 images. The following steps outline the process of loading the trained model, preparing the test data, and making predictions.

**6.1 Prerequisites**

Before starting, ensure you have the following files and directories:

* **pneumonia\_detection\_model.h5**: The trained CNN model.
* **test\_model.ipynb**: The notebook for testing the model.
* **testForChestXray/**: Directory containing the test images with subdirectories **NORMAL/** and **PNEUMONIA/**.

**6.2 Detailed Instructions**

1. **Load Required Libraries:**

Open the test\_model.ipynb notebook and start by loading the necessary libraries, such as TensorFlow, Keras, NumPy, and Matplotlib. These libraries are essential for loading the model, processing the images, and visualizing the results.

1. **Load the Trained Model:**

Load the pre-trained CNN model saved as pneumonia\_detection\_model.h5. This model will be used to make predictions on the test images.

1. **Prepare the Test Data:**

Use the ImageDataGenerator from Keras to rescale the pixel values of the test images. Load the test images from the testForChestXray directory using the flow\_from\_directory function. Ensure that the images are resized to 150x150 pixels to match the input size expected by the model.

1. **Evaluate the Model:**

Evaluate the model's performance on the test dataset by using the evaluate function. This function will provide the loss and accuracy of the model on the unseen test data.

1. **Make Predictions and Visualize Results:**

To understand how well the model performs, randomly select 10 images from the test dataset. Make predictions on these images using the loaded model. Visualize the actual labels and the model's predictions by displaying the images along with their labels and predicted classes. Use Matplotlib to create a subplot grid for displaying the images and their corresponding predictions.

