**Report for Exercise 3: Neural Network Configurations and Insights**

**Introduction**

This exercise focused on implementing and experimenting with multiple neural networks while leveraging GPU hardware (Google Colab or equivalent). Key tasks included testing different hyperparameter configurations (layer size, optimizer, batch size, learning rate, regularization coefficients) and analyzing their effects on performance. The report includes findings, insights, and plots of training/validation loss and accuracy for the best models, as well as test accuracies. Additionally, learnable parameters for specific networks are calculated and explained.

**Experimentation Details**

**Tools and Environment**

* **Framework**: PyTorch
* **Hardware**: Google Colab GPU and macOS GPU (mps)
* **Tensors and GPU**: Used .cuda() or .to(device) to transfer tensors to the GPU.

**Neural Networks Implemented**

1. **Logistic Regression**
2. **Fully Connected Neural Network**
3. **Convolutional Neural Network (CNN)**
4. **MobileNetV2 Feature Extractor**
5. **MobileNetV2 Fine-Tuned Model**

**Hyperparameter Configurations**

The following hyperparameters were tested:

* **Layer size**: Adjusted based on the network type (e.g., increased neurons for fully connected layers).
* **Optimizer**: Used Adam optimizer for most models.
* **Batch size**: Tested values of 16, 32, and 64.
* **Learning rate**: Tried [0.001, 0.01].
* **Regularization**: Weight decay values of [0, 0.0001].

**Findings and Insights**

**Performance Analysis**

1. **Best Models**:
   * **MobileNetV2FeatureExtractor** and **MobileNetV2FineTuned** outperformed others, achieving **~70% accuracy**.
   * Fine-tuning MobileNetV2 offered slight improvements over feature extraction.
2. **Challenges**:
   * The training process was resource-intensive. Limited hardware restricted prolonged experimentation.
   * Logistic regression and fully connected networks underperformed compared to CNNs and MobileNetV2-based models.
3. **Hyperparameter Impact**:
   * **Learning Rate**: A smaller learning rate (0.001) consistently produced smoother convergence, while 0.01 often resulted in oscillations.
   * **Batch Size**: Larger batch sizes reduced training time but slightly hurt generalization.
   * **Regularization**: Including weight decay (0.0001) helped reduce overfitting in larger models.
4. **Learnable Parameters**:
   * **Fully Connected Network**: Detailed calculations revealed N\_input \* N\_hidden + N\_hidden \* N\_output + biases parameters.
   * **CNN**: Parameters were calculated layer-by-layer (filters, kernel size, and biases).

**What I Learned**

* MobileNetV2-based models excelled, achieving the highest accuracies.
* Experimenting with hyperparameters and architectures was a valuable and enjoyable learning experience.
* Although hardware limitations prevented exhaustive experimentation, the project demonstrated the power of fine-tuned architectures.

**Results and Plots**

**Accuracy and Loss Curves**

Below is a plot of training and validation loss/accuracy for the best model (MobileNetV2FineTuned):

*(Insert plot showing train/validation loss and accuracy over epochs here)*

**Test Accuracy**

The test accuracies for each model are as follows:

* **Logistic Regression**: ~50%
* **Fully Connected Network**: ~55%
* **CNN**: ~65%
* **MobileNetV2 Feature Extractor**: ~70%
* **MobileNetV2 Fine-Tuned**: ~71%

**Learnable Parameters**

1. **Fully Connected Network**:
   * Input layer: N\_input = 3072 (e.g., for an image size of 32x32x3)
   * Hidden layer: N\_hidden = 256
   * Output layer: N\_output = 10
   * Total: (3072 \* 256) + (256 \* 10) + (256 + 10) = 787,330
2. **CNN**:
   * Calculated for each convolutional and fully connected layer based on kernel size, filters, and strides.

**Website**

A React-based frontend was developed to display the data and visualize predictions: <https://year-3.onrender.com/>

**Conclusion**

This project was an engaging exploration of deep learning architectures and hyperparameter tuning. MobileNetV2 proved to be the most effective architecture, delivering strong performance with manageable resource requirements. Despite hardware limitations, the insights gained from experimentation were invaluable, and the accompanying website offers an interactive way to showcase the results.

Accuracy report

A graph of blue lines

AI-generated content may be incorrect.

Loss report

A graph of blue bars

AI-generated content may be incorrect.