Assignment Analysis (Roll no. 3)

Task 1:

Single-Layer Perceptron (SLP)
The objective was to classify document images using a basic SLP. This model, being purely linear, achieved a maximum of 16.25% accuracy. This starkly demonstrated its fundamental limitation: an SLP cannot capture the complex, non-linear patterns essential for image recognition, especially given that flattening the images discarded crucial spatial information. Different learning rates showed sensitivity, but none could overcome the model's inherent architectural shortcomings for this complex dataset.

Task 2:

Multi-Layer Perceptron (MLP)
Moving to an MLP with hidden layers

aimed to overcome the SLP's linearity. We explored ReLU, Sigmoid, and Tanh activations. Tanh performed best at 21.67% accuracy, showing the MLP's increased capacity. However, Sigmoid performed poorly due to the vanishing gradient problem, while an extremely low learning rate for ReLU also crippled performance, highlighting the critical role of both activation function choice and learning rate tuning.

Task 3:

Activation and Loss Functions

This task compared activations and loss functions, emphasizing initialization. Cross-Entropy Loss (23-26% accuracy) proved far superior for multi-class classification compared to MSE Loss. Crucially, using "large" weight initialization (std=10) consistently led to unstable training and model failure across all activations, demonstrating the absolute necessity of proper initialization techniques (like PyTorch's defaults) to prevent vanishing/exploding gradients.

Task 4:
Optimizers (SGD, Adam, RMSprop)
Comparing optimization algorithms revealed Adam (28–30% accuracy) and RMSprop to be significantly more effective than SGD with momentum (24–26% accuracy). The adaptive learning rates of Adam and RMSprop allowed for faster, more stable convergence and higher final performance by navigating the complex loss landscape more efficiently, making them preferred choices for deep learning.

Task 5:
Batch Sizes & Momentum
This task investigated how batch size affects training dynamics. Moderate batch sizes (e.g., 64) provided a good balance, yielding 25–27% accuracy. Smaller batches (e.g., 8) could generalize slightly better (up to 28%) due to increased stochasticity but

were slower, while very large batches converged smoothly but sometimes led to lower generalization. Momentum was critical for SGD's effective convergence across all batch sizes tested.

Task 6:

Regularization (L1, L2, Dropout)
To combat overfitting, regularization methods were applied. Dropout (28–30% accuracy) and L2 regularization (27–29% accuracy) successfully reduced the generalization gap. They improved the model's performance on unseen data by preventing over-reliance on specific training examples and forcing the network to learn more robust, generalizable features from the high-dimensional image data.

Task 7:

Data Augmentation
This technique provided a dramatic leap in

generalization, boosting test accuracy to 35–40%. By artificially expanding the training dataset with various transformations, it greatly reduced overfitting and made the MLP more robust to real-world variations in document images, effectively increasing the "effective" training data size without collecting new images.

Task 8:

Transfer Learning (Pre-trained CNN Features)
This task achieved the most revolutionary performance leap, with test accuracy soaring to 75–80%. By utilizing high-level features extracted from a CNN pre-trained on ImageNet, the model effectively bypassed the need to learn basic visual features from scratch, demonstrating the immense power and efficiency of leveraging vast external knowledge for limited-data image tasks like RVL-CDIP.

Task 9:

Convolutional Neural Network (CNN) from Scratch
Building a CNN from scratch significantly outperformed all previous MLP models, reaching 45–50% test accuracy. This highlighted the CNN's inherent architectural superiority for image data, as its convolutional layers effectively captured local spatial features and hierarchical patterns, a critical advantage over MLPs that flatten image inputs.

Task 10: Learning Rate Schedules
Implementing dynamic learning rate schedules led to smoother convergence and modest gains, pushing the CNN's accuracy to 50–52%. By allowing larger steps initially and then gradually reducing the rate, these schedules enabled the optimizer to navigate the complex loss landscape more effectively, achieving better fine-tuning and slightly

higher final performance than a fixed learning rate.

Task 11: Hyperparameter Tuning (Grid Search/Random Search)
The final task involved systematically optimizing hyperparameters using methods like Random Search. This process allowed for the discovery of the absolute best configuration, further pushing the CNN's performance to 54-56% accuracy. It underscored that meticulous tuning is crucial for extracting peak performance, revealing the interdependencies between various hyperparameters and the computational cost involved in achieving optimal results.