Advanced Machine Learning – Assignment 01

Arnab Bera (MDS202409) Biswajit Kala (MDS202412)

September 28, 2025

Abstract

This assignment explores two fundamental aspects of deep learning: spatial locality in CNNs through pixel permutation experiments (Task 1) and transfer learning for emotion classification using fine-tuning strategies (Task 2).

1 Task 1: Spatial Locality in CNNs

1.1 Abstract

In this task, we investigate the importance of spatial locality in Convolutional Neural Networks (CNNs) by comparing standard training versus pixel-permuted training on the NotMNIST dataset. We train two identical CNN architectures: one on original letter images and another on pixel-scrambled versions to demonstrate the critical role of spatial structure in CNN performance.

1.2 Dataset and Methodology

1.2.1 Dataset

The NotMNIST dataset contains grayscale 28×28 images of 10 letter classes $\{A, B, C, D, E, F, G, H, I, J\}$ with corresponding labels $\{0, 1, 2, \ldots, 9\}$.

1.2.2 CNN Architecture

We employ a deep convolutional network with 5 layers:

$$\begin{aligned}
&\text{Conv2d}(1 \to 64) + \text{BatchNorm} + \text{ReLU} \\
&\to \text{Conv2d}(64 \to 128) + \text{MaxPool} \\
&\to \text{Conv2d}(128 \to 256) + \text{MaxPool} \\
&\to \text{Conv2d}(256 \to 512) + \text{MaxPool} \\
&\to \text{Linear}(512 \to 10)
\end{aligned}$$

1.2.3 Experimental Design

- **Model 1:** Standard training on original 28×28 images
- **Model 2:** Training on pixel-permuted images using a fixed permutation π

1.2.4 Pixel Permutation Process

For each image $X \in \mathbb{R}^{28 \times 28}$, we apply:

$$X_{\text{permuted}} = \text{reshape}\left(\text{flatten}(X)[\pi], 28, 28\right) \tag{1}$$

where $\pi : \{1, 2, \ldots, 784\} \to \{1, 2, \ldots, 784\}$ is a fixed bijective mapping that scrambles spatial pixel positions while preserving all pixel intensity information.

1.3 Training Configuration

- **Optimizer:** SGD with learning rate $\alpha = 0.01$, momentum = 0.5
- **Loss Function:** Negative Log-Likelihood (NLL)
- **Training:** 20 epochs, batch size 64
- **Data Preprocessing:** Resize to 28×28, normalize to $[-1, 1]$

1.4 Expected Results and Significance

1.4.1 Hypothesis

We expect standard CNN training to significantly outperform pixel-permuted training:

$$\text{Accuracy}_{\text{Standard}} \gg \text{Accuracy}_{\text{Permuted}} \tag{2}$$

This performance gap demonstrates that CNNs rely heavily on spatial locality and translation invariance properties inherent in image data.

1.5 Conclusion

This comparison shows that the arrangement of pixels in an image is very important for CNNs. When we train the same network on normal images and on scrambled images, its accuracy drops a lot if the image structure is broken.

2 Task 2: Emotion Classification with Fine-Tuning

2.1 Abstract

This task explores transfer learning for emotion classification using ResNet18. We compare pre-trained model performance against fine-tuned models with differential learning rates across network layers, demonstrating significant improvements in emotion recognition accuracy.

2.2 Dataset and Methodology

2.2.1 Dataset

Emotion Detection FER dataset containing 7 emotion classes:

- **Classes:** Angry, Disgusted, Fearful, Happy, Neutral, Sad, Surprised
- **Image Size:** Resized to 224×224
- **Channels:** Grayscale converted to 3-channel RGB

2.2.2 Data Preprocessing

```
emotion_train_transform = transforms.Compose([
                              transforms.Resize((224, 224)),
                              transforms.Grayscale(num_output_channels=3),
                              transforms.ToTensor(),
                              transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
                          ])
 emotion_test_transform = same as train with identical normalization
```

2.2.3 Model Architecture

- **Base Model:** ResNet18 pre-trained on ImageNet
- **Final Layer:** Modified to 7 output classes
- **Fine-tuning:** Differential learning rates per layer group

2.2.4 Training Strategy

- **Optimizer:** SGD with momentum = 0.9
- **Loss Function:** CrossEntropyLoss
- **Epochs:** 10
- **Batch Size:** 64

2.3 Fine-Tuning Methodology

2.3.1 Differential Learning Rates

$$\begin{aligned}
\text{Layer 1 (early)} &: \text{LR} = 1 \times 10^{-5} \\
\text{Layer 2} &: \text{LR} = 5 \times 10^{-5} \\
\text{Layer 3} &: \text{LR} = 1 \times 10^{-4} \\
\text{Layer 4} &: \text{LR} = 5 \times 10^{-4} \\
\text{Fully Connected} &: \text{LR} = 1 \times 10^{-3} \\
\text{Other layers} &: \text{LR} = 1 \times 10^{-5}
\end{aligned}$$

2.4 Results and Analysis

2.4.1 Overall Performance Comparison

Table 1: Overall Performance Metrics Comparison

Metric	Pre-trained Model	Fine-tuned Model	Improvement
Test Loss	1.4385	1.2444	**-13.5%**
Test Accuracy	45.46%	62.69%	**+17.23%**
Macro Avg F1-Score	0.38	0.59	**+55.3%**
Weighted Avg F1-Score	0.44	0.62	**+40.9%**

2.4.2 Per-Class Performance Analysis

Table 2: Detailed Per-Class F1-Score Comparison

Emotion	Pre-trained	Fine-tuned	Improvement	Support
Angry	0.32	0.54	**+69%**	958
Disgusted	0.06	0.49	**+717%**	111
Fearful	0.25	0.46	**+84%**	1024
Happy	0.60	0.83	**+38%**	1774
Neutral	0.46	0.58	**+26%**	1233
Sad	0.40	0.51	**+28%**	1247
Surprised	0.59	0.77	**+31%**	831

2.5 Conclusion

Performance Gains:

- **Accuracy increased by 17.23%** (45.46% → 62.69%)
- **Loss decreased by 13.5%** (1.4385 → 1.2444)
- **Macro F1-Score improved by 55.3%** (0.38 → 0.59)
- **All individual classes showed improvement**

Final Conclusion: Fine-tuning the pre-trained ResNet18 model with differential learning rates dramatically improved emotion classification performance.