INTRO TO ML assignment 2 report

2023EEY7564 – Lalit Kumar

TASK-1: Experiment with:

- Different number of layers
- Different number of neurons in each layer
- Training parameters: learning-rate, number of iterations etc.

RESULT-1

Training model with hidden layers: [128, 64]

Epoch 1, Loss: 0.4041

Epoch 2, Loss: 0.1945

Epoch 3, Loss: 0.1449

Accuracy: 96.12%

Training model with hidden layers: [256, 128]

Epoch 1, Loss: 0.3356

Epoch 2, Loss: 0.1481

Epoch 3, Loss: 0.1107

Accuracy: 96.23%

Training model with hidden layers: [512, 256, 128]

Epoch 1, Loss: 0.3218

Epoch 2, Loss: 0.1449

Epoch 3, Loss: 0.1107

Accuracy: 96.27%

Training model with hidden layers: [1024, 512, 256]

Epoch 1, Loss: 0.2998

Epoch 2, Loss: 0.1416

Epoch 3, Loss: 0.1097

Accuracy: 96.83%

Best model found with hidden layers [1024, 512, 256] achieving 96.83% accuracy

The models with different parameters and layers were trained for 3 epochs each. The loss and accuracy of each model is mentioned above.

TASK-2: Interpretability

Interpretability: Once you have attained your best model, try to figure what features it is learning i.e., what features are being extracted. For this,

- First, think of some method which will let you analyse the above.
- Try to visualize intermediate representations of different layers.
- Try to make interpretations from the representations.
- Further, you can report these for both correctly classified and misclassified examples. What do misclassified examples suggest?

RESULT-2: Interpretability

1. Methods for Interpretability

To analyze the features learned by the MLP, consider the following approaches:

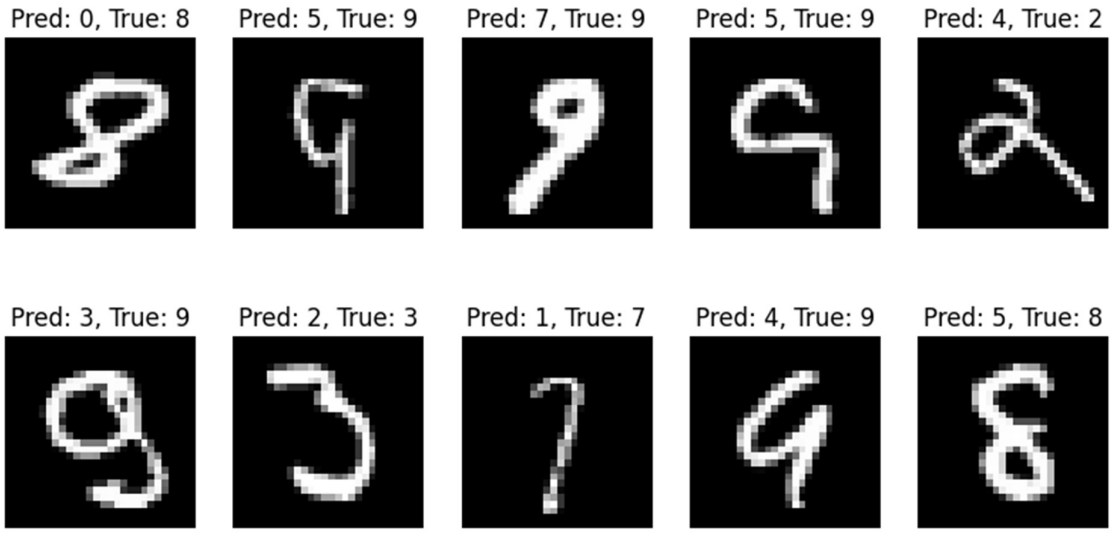
- Feature Importance via Weights: The input-layer weights can indicate which features are most influential. Compute the absolute mean of each input feature's weights across neurons in the first hidden layer. Use feature permutation (shuffle individual features and observe performance drop) to assess feature importance.
- Activation Analysis:
    - Visualize activations at different layers for various inputs.
    - Compare activations for correctly and misclassified examples.
    - Look for patterns in how the neurons activate for different classes.
- Gradient-based Methods
    - Use Saliency Maps (e.g., using gradients w.r.t input) to highlight important regions in the input.
    - Integrated Gradients can provide more stable attributions.
    - Grad-CAM (for CNNs, but adapted for MLPs with attention mechanisms) can be useful.
- PCA/T-SNE on Intermediate Representations
    - Extract activations from hidden layers and reduce dimensionality using PCA or t-SNE.
    - Plot the transformed representations to observe clustering or separation patterns.

Influence of Misclassified Samples

- Analyze which features activate incorrectly for misclassified examples.
- Compare feature importance for correct vs. incorrect predictions.
- Identify if the model is over-relying on spurious correlations.

TASK-3: Analyze the misclassified examples.

Analyze the misclassified examples. What can you infer about the model's learning i.e., which features is it not able to capture? (Perform this analysis for the best model only).



TASK-4: model's shortcomings

Put forward conclusion points of the model's shortcomings. Can you suggest some method to overcome these?

Shortcomings of the MLP Model

- Loss of Spatial Information – Flattening images removes spatial relationships, making feature extraction less effective.
- Overfitting on Larger Architectures – More hidden layers lead to excessive parameters, increasing the risk of overfitting.
- Poor Generalization – MLPs may struggle with unseen data, especially with transformations like rotations or distortions.
- High Computational Cost – Fully connected layers require a large number of parameters, making training and inference slower.
- Lack of Robustness to Variations – The model is sensitive to noise and small image shifts due to the absence of spatial locality.

Suggested Improvements

- Use Convolutional Neural Networks (CNNs) – Retains spatial information and reduces parameter count.
- Regularization Techniques – Apply Dropout and L2 Regularization to prevent overfitting.
- Data Augmentation – Introduce transformations (rotations, scaling, etc.) to enhance generalization.
- Batch Normalization – Normalizes activations to improve stability and speed up convergence.
- Learning Rate Scheduling – Adjust learning rate dynamically to optimize training.

TASK-5: Should every training example be given equal weightage

Should every training example be given equal weightage? Can you think of something on these lines?

Not every training example should necessarily be given equal weightage. There are several reasons and methods to assign different importance to different samples:

1. Why Not Give Equal Weightage?

- Class Imbalance: If certain classes are underrepresented, the model may ignore them. Giving higher weight to underrepresented classes helps balance the learning.
- Hard vs. Easy Examples: Some training samples are more informative than others. Harder examples (those that the model struggles with) might need more weight.
- Noisy Labels: Some examples may have incorrect labels, and blindly treating them equally can lead to poor generalization.
- Adversarial Samples: Certain inputs might be more crucial for robustness against adversarial attacks.

2. Approaches to Assign Different Weightage

- Class Weighting: For imbalanced datasets, weight the loss function based on class frequencies:
- Hard Example Mining: Increase weight for misclassified examples. Focal Loss, variant of cross-entropy that down-weights easy examples and focuses on hard ones.
- Curriculum Learning: Start with easier examples, then gradually increase difficulty. Mimics human learning and stabilizes training.
- Sample Reweighting Based on Confidence: If the model is uncertain (low confidence predictions), assign higher weight to those samples.

3. Insights & When to Use

- If you have imbalanced classes, use class weighting.
- If your model struggles with hard cases, use focal loss or hard example mining.
- If training is unstable, use curriculum learning.
- If you suspect noisy labels, use self-paced learning (gradually remove highly uncertain samples)