Research Paper Assignment 4

Applied AI I - Neural Network Architecture Comparison Study

IMPORTANT: This is a LEARNING EXERCISE!

Don't panic! This assignment is designed to help you:

- Understand neural network architectures deeply
- Compare MLPs/FNN and CNNs on image data
- Analyze training dynamics and convergence
- Practice hyperparameter tuning
- Build intuition for network design

Remember: The goal is understanding HOW neural networks behave, not just achieving highest accuracy. Focus on analysis and learning!

Assignment Overview

Research Question:

"How do architectural choices (depth, width, regularization) affect neural network performance and training dynamics on image classification?" **Goal:** Compare MLP and CNN architectures systematically, analyzing how design decisions impact accuracy, training time, and generalization.

Deliverables:

1. Research paper (4-6 pages PDF)
2. Jupyter notebook with all experiments
3. All code must be reproducible
4. Use Weights & Bias as Experiment Tracking Tool.

Value: This demonstrates your understanding of neural network fundamentals!

Learning Objectives

By completing this assignment, you will:

1. **Build** MLP and CNN architectures from scratch
2. **Train** neural networks on real image data
3. **Compare** different architectural choices systematically
4. **Analyze** training dynamics (loss curves, accuracy curves)
5. **Apply** regularization techniques (dropout, batch normalization)
6. **Visualize** learned features and predictions

7. **Communicate** findings clearly with proper visualizations
8. Use Experiment Tracking Platform for your experiments (Weights & Biases)

Key Skill: Understanding how architectural choices affect neural network behavior!

Part 1: Dataset Selection

Requirements

Your dataset MUST be: - Image classification dataset - Minimum 10,000 training samples - 3-10 classes - Manageable image size (32x32 or similar) - Publicly available with proper citation

Recommended Datasets

Option 1: Fashion-MNIST (Recommended for beginners) - 28x28 grayscale images - 10 clothing categories - 60,000 training, 10,000 test - Easy to train, good for comparisons

Option 2: CIFAR-10 (More challenging) - 32x32 color images - 10 object classes - 50,000 training, 10,000 test - Requires CNNs to perform well

Option 3: MNIST (Simpler) - 28x28 grayscale digits - 10 digit classes - Good baseline, but may be too simple

Option 4: Custom Dataset (With approval) - Discuss with instructor first - Must meet size requirements - Must be appropriate for comparison

Dataset Requirements

You MUST: 1. Load and explore the dataset 2. Visualize sample images from each class 3. Check class balance 4. Normalize images appropriately 5. Create train/validation/test splits

Part 2: Methodology

Experimental Design

You will compare multiple architectures systematically:

Experiment 1: MLP Baseline - Flatten images to vectors - Compare: shallow (1 hidden) vs. deep (3+ hidden) - Compare: narrow (64 neurons) vs. wide (512 neurons)

Experiment 2: CNN Architecture - Build CNN with conv + pooling layers - Compare: shallow CNN vs. deeper CNN - Analyze parameter efficiency vs. MLP

Experiment 3: Regularization Effects - Without regularization (baseline) - With dropout (vary rates: 0.2, 0.3, 0.5)

Experiment 4: Training Dynamics - Plot learning curves (loss and accuracy) - Analyze convergence speed - Identify overfitting patterns - Compare learning rates

Required Architectures

Architecture A: Simple MLP

```
Input (784 or 3072) → Dense(128) → ReLU → Dense(10) → Softmax
```

Architecture B: Deep MLP

```
Input → Dense(256) → ReLU → Dense(128) → ReLU → Dense(64) → ReLU → Dense(10)
```

Architecture C: Simple CNN

```
Input → Conv(32, 3x3) → ReLU → MaxPool(2x2) → Flatten → Dense(128) →
Dense(10)
```

Architecture D: Deeper CNN

```
Input → Conv(32) → BN → ReLU → MaxPool → Conv(64) → BN → ReLU → MaxPool →
Flatten → Dense(256) → Dense(10)
```

Part 3: Required Experiments

Prerequest Experiments: For Experiment Tracking please use: Weights & Bias as Experiment Tracker. How to get an Account: https://moodle.hs-ansbach.de/pluginfile.php/252051/mod_resource/content/0/lesson16_wandb_login.pdf

Example of how to integrate Weights & Bias: Week 9 in Moodle: Lab01 Experiment Tracking with WandB 2.ipynb Datei

Experiment 1: MLP Depth and Width Study

Steps: 1. Train Simple MLP (Architecture A) 2. Train Deep MLP (Architecture B) 3. Vary width: 64, 128, 256, 512 neurons 4. Record training time, final accuracy, parameter count 5. Plot learning curves for all variants

Analysis Questions:

- Does deeper always mean better?
- What is the effect of width?
- How many parameters does each variant have?
- Which shows overfitting?

Experiment 2: MLP vs. CNN Comparison

Steps: 1. Train best MLP from Experiment 1 2. Train Simple CNN (Architecture C) 3. Train Deeper CNN (Architecture D) 4. Compare on same data with same epochs 5. Analyze accuracy vs. parameter count

Analysis Questions: - How much better is CNN than MLP? - How many fewer parameters does CNN need? - Where does MLP fail that CNN succeeds? - Visualize misclassified examples for both

Experiment 3: Regularization Study

Steps: 1. Train CNN without any regularization 2. Add dropout (0.2, 0.3, 0.5) - compare 3. Record train/validation accuracy gap

Analysis Questions: - How does dropout affect the train-val gap? - Which dropout rate works best?

Experiment 4: Learning Rate Study

Steps: 1. Train best architecture with lr = 0.1, 0.01, 0.001, 0.0001 2. Plot loss curves for all learning rates 3. Note convergence speed and stability 4. Try learning rate schedulers (optional)

Analysis Questions: - Which learning rate converges fastest? - Which learning rate gives best final accuracy? - Any learning rates that fail to converge? - Trade-off between speed and final performance?

Part 4: Paper Structure

Required Sections (4-6 pages)

Template IEEE is necessary.

1. Abstracht
2. Introduction
3. Related Work
4. Dataset and Preprocessing
5. Methodology
6. Experiments & Results
7. Discussion
8. Conclusion
9. References

Part 5: Code Requirements

Jupyter Notebook Structure

```
# ==========================================

# Week 8: Neural Network Architecture Study

# Student: [Your Name]
```

```python
# Dataset: [Dataset Name]
# ========================================

# -- IMPORTS --
import torch
import torch.nn as nn
import torch.optim as optim
from torchvision import datasets, transforms
from torch.utils.data import DataLoader
import matplotlib.pyplot as plt
import numpy as np

# -- CONFIG --
RANDOM_SEED = 42
BATCH_SIZE = 64
EPOCHS = 20
DEVICE = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

torch.manual_seed(RANDOM_SEED)

# -- 1. DATA LOADING --
# [Load and visualize dataset]
```

```
# -- 2. MODEL DEFINITIONS --
# [Define all architectures]

# -- 3. TRAINING FUNCTION --
# [Reusable training loop]

# -- 4. EXPERIMENT 1: MLP STUDY --
# [MLP depth/width experiments]

# -- 5. EXPERIMENT 2: CNN COMPARISON --
# [MLP vs. CNN comparison]

# -- 6. EXPERIMENT 3: REGULARIZATION --
# [Dropout and batch norm experiments]

# -- 7. EXPERIMENT 4: LEARNING RATE --
# [Learning rate study]

# -- 8. VISUALIZATION --
# [Filters, activations, confusion matrix]

# -- 9. RESULTS SUMMARY --
# [Tables and final analysis]
```

Part 6: Evaluation Rubric

Total: 100 points

Dataset and Setup (10 points)

- 10: Properly loaded, visualized, preprocessed
- 7: Minor issues
- 4: Significant issues
- 0: Incorrect or missing ### Architecture Implementation (20 points)
- 20: All architectures correctly implemented
- 15: Minor implementation issues
- 10: Some architectures incorrect
- 5: Major errors ### Experiments (25 points)
- 25: All 5 experiments complete and correct
- 20: 4 experiments complete
- 15: 3 experiments complete
- 10: Partial experiments ### Analysis Quality (20 points)
- 20: Deep analysis of all results
- 15: Good analysis
- 10: Superficial analysis
- 5: Minimal analysis ### Writing Quality (15 points)
- 15: Clear, well-organized, professional (and low to middle amount of AI generated text)
- 10: Good quality (and middle amount of AI generated text)
- 5: Adequate but issues (and high amount of AI generated text)
- 0: Poor quality ### Visualizations (10 points)
- 10: All required visualizations, professional quality
- 7: Most visualizations, good quality
- 4: Some missing or poor quality
- 0: Missing or very poor

BONUS POINTS (up to +10)

- +3: Additional architecture experiments
- +3: Data augmentation study
- +2: Learning rate scheduler comparison
- +2: Exceptional visualizations
- +2: Early stopping implementation

Part 7: Tips for Success

DO

- Start with simple architectures, add complexity
- Plot learning curves for EVERY experiment
- Compare parameter counts, not just accuracy
- Analyze WHERE models fail (misclassifications)
- Set random seeds for reproducibility
- Save training history for plotting
- Run multiple times to verify results ### DON'T
- Use pre-trained models (build from scratch!)
- Skip experiments to save time
- Only report accuracy (show full analysis)
- Forget to normalize inputs
- Train for too few epochs
- Ignore validation performance

Part 9: Final Encouragement

You Can Do This!

This assignment teaches you: - How neural networks really work - Why CNNs are better for images - How to tune hyperparameters - How to analyze results systematically

Success means: - Implementing networks from scratch - Running systematic experiments - Analyzing and explaining results - Writing a clear paper

Remember: - Start simple, build up complexity - Every experiment teaches something - Negative results are valuable too - Understanding > perfect accuracy - **Start early. Experiment systematically. Use MLFlow. Analyze deeply.**

You've got this!

Happy training!