Comprehensive Deep Learning Project:
From Theory to Deployment

Weeks 5–10: Regularization, Sequences, Transformers, Generative Models, Vision & Ethics

Total Points: 25
Due Date: Feb 12

Project Overview

Motivation: Why This Project Matters

In real-world machine learning, you rarely work on isolated problems. Instead, you must:

- Build models that *generalize* well to unseen data (not just memorize training examples)
- Choose appropriate architectures for different data types (sequences vs images)
- Navigate trade-offs between model complexity, performance, and computational cost
- Consider ethical implications of your deployed systems
- Communicate technical decisions to diverse stakeholders

This project simulates a realistic ML engineering workflow where you'll design, implement, analyze, and present a complete intelligent system. You'll apply concepts from regularization, sequence modeling, transformers, generative models, computer vision, and deployment—demonstrating not just *what* these techniques do, but *when* and *why* to use them.

Learning Objectives

By completing this project, you will:

1. Apply regularization techniques to combat overfitting and interpret bias-variance trade-offs
2. Implement and compare recurrent architectures (RNN/LSTM) with transformer models
3. Design and train a generative model (VAE or GAN) for creative applications
4. Build a vision system using modern object detection or segmentation methods
5. Deploy a model with optimization techniques (quantization/pruning)
6. Critically evaluate ethical considerations in your system's design and deployment

Project Task: Build an Intelligent Healthcare Assistant System

Scenario

You are hired as an ML engineer at a healthcare technology company. Your task is to build an intelligent healthcare assistant system that can:

1. Analyze medical text reports and patient notes to identify key clinical information (sequence modeling)
2. Detect and classify medical conditions from chest X-rays or skin lesion images (computer vision)
3. Generate synthetic medical data for training while protecting patient privacy (generative models)
4. Deploy efficiently on mobile devices for point-of-care diagnostics (optimization)
5. Operate fairly across diverse patient populations (ethics)

Detailed Requirements & Grading Rubric

Part 1: Medical Text Analysis with Sequence Models (7 points)

Task: Build a text classifier to extract clinical information from medical reports or classify patient sentiment/urgency from medical queries.

Requirements:

a) **Dataset:** Use a public dataset such as:
   - MIMIC-III Clinical Notes (requires credentialing but free)
   - PubMed abstracts for disease classification
   - Medical Question-Answer datasets (e.g., MedQuAD)
   - SMS Spam Collection adapted for medical appointment reminders classification

b) **Implementation:** Train *both* of the following:
   - An LSTM/GRU-based classifier
   - A Transformer-based classifier (you may use pretrained embeddings like BioBERT, ClinicalBERT, or standard BERT)

c) **Regularization Analysis:** For each model, systematically apply:
   - Dropout (test at least 3 different rates: 0.1, 0.3, 0.5)
   - L2 regularization (test at least 3 different λ values)
   - Early stopping with validation monitoring

d) **Required Deliverables:**
   - Training and validation loss/accuracy curves for both architectures

- Comparison table showing: number of parameters, training time, final accuracy, overfitting severity (training acc - validation acc)
- Analysis of vanishing gradient problem: plot gradient norms at different layers during training for the RNN/LSTM
- Written explanation (150–250 words): Why does the Transformer outperform or underperform the LSTM on this task? Reference attention mechanisms and sequence dependencies.

Grading Breakdown (7 points):

- Correct implementation of both models: 2 points
- Regularization experiments with proper documentation: 2 points
- Training curves and comparison table: 1.5 points
- Gradient analysis and architecture comparison: 1.5 points

Part 2: Medical Image Analysis with Vision Models (6 points)

Task: Build a medical image classification or detection system for automated diagnosis assistance.

Requirements:

a) **Dataset Options (choose one):**

- **Chest X-rays:** ChestX-ray14, CheXpert, or COVID-19 chest X-ray datasets
- **Skin Lesions:** HAM10000 or ISIC skin lesion datasets
- **Retinal Images:** Diabetic Retinopathy Detection (Kaggle)
- **Brain MRI:** Brain Tumor Classification datasets
- **General Option:** COCO or ImageNet for general object detection/classification

b) **Implementation:** Implement *one* of the following:

- Multi-class classification (e.g., classify disease types)
- Object detection using YOLO, SSD, or Faster R-CNN (e.g., detect lesions, tumors, or anatomical structures)
- Semantic segmentation for region analysis (e.g., segment organs or lesions)

c) **Bias-Variance Analysis:**

- Train models with different capacities (e.g., ResNet-18 vs ResNet-50, or different backbone architectures)
- Plot learning curves showing training and validation performance
- Identify whether your model exhibits high bias or high variance
- Apply appropriate regularization techniques based on your diagnosis

d) **Required Deliverables:**

- Visualization of 5–10 prediction results with class labels, bounding boxes, or segmentation masks
- Learning curves demonstrating bias-variance trade-off
- Quantitative metrics: precision, recall, F1-score, AUC-ROC, or mAP (if applicable)
- Written analysis (200–300 words): How did you diagnose and address overfitting/underfitting? What regularization strategies worked best?

Grading Breakdown (6 points):

- Correct model implementation and training: 2 points
- Bias-variance analysis with learning curves: 2 points
- Visualizations and metrics: 1 point
- Written analysis of regularization strategies: 1 point

Part 3: Synthetic Medical Data Generation with Generative Models (5 points)

Task: Generate synthetic medical data to augment your dataset, handle class imbalance, or protect patient privacy.

Requirements:

a) **Implementation:** Build *one* of the following:

- A Variational Autoencoder (VAE) to generate synthetic medical images (e.g., chest X-rays, skin lesions)
- A Generative Adversarial Network (GAN) to generate synthetic medical images
- A VAE or GAN to generate synthetic tabular patient data or text embeddings

b) **Training Challenges:**

- Document challenges encountered (e.g., mode collapse in GANs, posterior collapse in VAEs)
- Describe strategies used to stabilize training
- If using GAN: show discriminator and generator loss curves
- If using VAE: show reconstruction loss and KL divergence curves

c) **Required Deliverables:**

- 10–20 generated samples (medical images, tabular data, or text embeddings visualized)
- Loss curves during training
- Qualitative evaluation: Do generated samples look realistic? Are they diverse? Do they maintain medical plausibility?
- Written reflection (150–200 words): What was the biggest training challenge? How did you address it? How could synthetic data help with privacy concerns?

Grading Breakdown (5 points):

- Working generative model implementation: 2 points
- Documentation of training challenges and solutions: 1.5 points
- Generated samples and evaluation: 1 point
- Written reflection: 0.5 points

Part 4: Model Deployment & Optimization (4 points)

Task: Optimize one of your models for deployment on resource-constrained devices (e.g., mobile devices for point-of-care diagnostics).

Requirements:

a) **Choose** one model from Part 1 or Part 2 to optimize

b) **Apply** at least two of the following optimization techniques:
   - Pruning (structured or unstructured)
   - Quantization (int8 or mixed precision)
   - Knowledge distillation
   - Model export to ONNX or TensorRT format

c) **Benchmark:**
   - Model size (MB) before and after optimization
   - Inference time (ms) before and after optimization
   - Accuracy drop (if any) after optimization

d) **Required Deliverables:**
   - Comparison table showing size, speed, and accuracy trade-offs
   - Written explanation (100–150 words): Which optimization technique was most effective? What are the deployment constraints you considered? Why is model optimization important for healthcare applications?

Grading Breakdown (4 points):

- Correct implementation of optimization techniques: 2 points
- Benchmark results and comparison: 1.5 points
- Written explanation: 0.5 points

Part 5: Ethics, Bias & Fairness Analysis (3 points)

Task: Critically evaluate ethical considerations in your healthcare AI system.

Requirements:

a) **Bias Audit:**

- Test your model on different demographic subgroups (if metadata is available: age groups, genders, skin tones for dermatology, etc.)
- Report performance disparities (e.g., accuracy across different patient populations)
- If demographic data is unavailable, discuss what biases might exist and how you would test for them

b) **Ethical Analysis:** Write a 400–600 word report addressing:

- *Privacy*: What are the privacy implications of using patient medical data? How does synthetic data generation help or hinder privacy protection?
- *Fairness*: How might your model perform differently across patient populations (e.g., different ethnicities, ages, socioeconomic backgrounds)? What mitigation strategies could you employ?
- *Transparency & Explainability*: How would you explain model predictions to healthcare providers and patients? Why is interpretability critical in medical AI?
- *Clinical Validation*: What steps would be necessary before deploying this system in a real clinical setting? What are the risks of false positives and false negatives?
- *Dual-use risks*: Could your generative model be misused (e.g., creating fake medical records)? How would you prevent harmful applications?

c) **Required Deliverables:**

- Bias audit results (quantitative if possible, qualitative discussion if data limitations exist)
- Ethics report (400–600 words)
- At least 2 concrete recommendations for improving fairness, transparency, or reducing harm in your healthcare AI system

Grading Breakdown (3 points):

- Bias audit with documentation: 1 point
- Ethics report quality and depth: 1.5 points
- Concrete recommendations: 0.5 points

Submission Guidelines

Required Files:

1. **Code:** Jupyter notebooks or Python scripts for all implementations (well-commented)
2. **Report:** A single PDF document (maximum 12 pages) containing:
   - All required analyses, plots, tables, and written sections
   - Clear section headers corresponding to Parts 1–5
   - References to figures and tables

Submission Format:

- Create a single ZIP file named: `LastName_FirstName_DL_Project.zip`
- Upload to the course submission portal by the deadline
- Late submissions are not accepted. Last time, I accepted a few due to extenuating circumstances, but this time there will be no exceptions.

Evaluation Criteria Summary

Component	**Points**
Part 1: Medical Text Analysis (Sequence Models)	7
Part 2: Medical Image Analysis (Vision Models)	6
Part 3: Synthetic Medical Data (Generative Models)	5
Part 4: Deployment & Optimization	4
Part 5: Ethics & Fairness	3
Total	**25**

Questions?

Post clarification questions on the course forum or attend office hours. Good luck, and have fun building your intelligent healthcare system!

This project is designed to challenge you while preparing you for real-world ML engineering in healthcare. Remember: the goal is not perfection, but thoughtful application of deep learning principles to a complex, multi-faceted problem with real-world impact.