Assignment Name: ***L05-AWS Machine Learning University Module 2 Lab Exploration***

Course Name: ***ITAI 2376 - Deep Learning in Artificial Intelligence (Spring 2025)***

Lab Module: ***Module 2, Labs 1-4***

Instructor Name: ***Professor Patricia McManus***

Prepared by: ***Olugbenga Adegoroye***

Date: ***25th February, 2025***

Reflective Journal on Machine Learning Labs (Module 2, Labs 1-4)

Introduction

Diving into Module 2 of "Application of Deep Learning to Text and Images" through Labs 1 to 4 has been a transformative experience, immersing me in the intricate world of natural language processing (NLP) and its synergy with machine learning. These notebooks, spanning text preprocessing to advanced neural networks, offered a hands-on exploration of how raw language can be distilled into forms computers can understand and leverage. As someone new to this domain, I approached these labs with a mix of excitement and trepidation, unsure of how theoretical concepts would translate into practical skills. What unfolded was a progressive journey—from cleaning text and visualizing word frequencies to training a recurrent neural network on Amazon reviews—each step building on the last. This journal reflects my insights, struggles, growth, and critical thoughts over roughly three pages, capturing not just what I learned, but how I evolved through the process. It's a personal narrative of grappling with code, uncovering meaning in data, and finding my footing in a field that's as challenging as it is rewarding.

Learning Insights

Engaging with Labs 1 through 4 opened my eyes to core machine learning ideas, especially how unstructured text morphs into structured, numerical data for analysis. Lab 1 kicked things off with text preprocessing—think word clouds, stemming, lemmatization, and named entity recognition (NER)—showing me the first steps to tame messy text. Lab 2 brought in the bag-of-words (BoW) approach, where I learned to turn sentences into vectors using binary flags, word counts, and TF-IDF, revealing how to measure word significance. Lab 3 upped the game with GloVe word embeddings, which map words into a high-dimensional space to capture their meanings and relationships. The Lab 4 tied it all together, using a Recurrent Neural Network (RNN) to classify text, highlighting how machines can process sequences like human speech.

These labs tie into bigger machine learning concepts like feature engineering and model design. Moving from BoW's manual feature crafting to embeddings' learned representations echoes a shift I've read about in deep learning's evolution. Lab 3's cosine similarity, used to compare word vectors, feels like a cousin to techniques in clustering or recommendation systems. Lab 4's RNN introduced sequence modeling, a principle I now see as foundational to modern NLP giants like transformers, where context is king.

The standout moments hit me hard. In Lab 3, running simCompare and finding "car" closer to "truck" than "bike" in embedding space was a lightbulb moment—numbers reflecting meaning blew my mind. In Lab 4, seeing the RNN's validation accuracy hit 0.79 after 65 epochs felt like a payoff for sticking with it, showing me how training refines a model over time. Those moments made abstract ideas real.

Challenges and Struggles

Each lab threw curveballs my way. Lab 1's stemming versus lemmatization debate tripped me up—stemming's blunt cuts (like "involv" from "involve") seemed sloppy next to lemmatization's nuance, but I couldn't nail down when to pick one over the other. Lab 2's BoW methods introduced sparse matrices, and decoding CountVectorizer outputs felt like deciphering a secret code at first. Lab 3's GloVe embeddings brought 50-dimensional tensors—wrapping my head around "cat" as a long list of numbers was tough until cosine similarity clicked. Lab 4's RNN was the beast—PyTorch syntax, sequence padding, and the model's layers had me scrambling to keep up. I tackled these hurdles by slowing down and digging in. For stemming and lemmatization, I pored over examples, realizing stemming's speed suits big datasets while lemmatization's accuracy fits semantic tasks. In Lab 2, I traced BoW arrays back to the vocabulary by hand, piecing together the puzzle. Lab 3's tensors made sense once I sketched cosine similarity as angles between vectors, grounding the math in something visual. For Lab 4, I dissected the RNN code line by line, and searched up PyTorch terms like nn.Embedding, and tested snippets to get comfortable.

My problem-solving toolkit grew into a blend of trial and error plus big-picture thinking. The "Try it Yourself" sections became my playground—I'd tweak and run code to see what broke or worked. Drawing diagrams, like RNN layers, helped me link code to concepts. When stuck, I'd rephrase the issue in plain English, which either solved it or pointed me to the right question to chase online.

Personal Growth

These labs reshaped how I see machine learning—from a fuzzy "computers learn stuff" idea to a clear grasp of data prep, representation, and modeling. I used to view NLP as magic; now, it's a pipeline I can tweak: clean the text, encode it, build a model, check the results. This shift turned me from a passive learner into someone who questions tools and their trade-offs—like why TF-IDF balances frequency and rarity, or how RNNs mimic memory.

The slow burn of learning caught me off guard. I wanted instant "aha!" moments, but understanding crept up—Lab 1's word clouds seemed basic until Lab 3 tied them to frequency analysis, and Lab 4's RNN clicked only after wrestling with embeddings. That pace taught me grit and patience, traits I didn't expect to lean on so much in tech.

These skills feel like gold for what's ahead. In school, I could analyze texts or social media for projects. In a job, preprocessing text and training models could power chatbots or sentiment tools—real-world stuff like improving customer service or spotting market vibes. Even debugging tensor shapes feels like a universal coding win I can carry anywhere.

Critical Reflection

Looking back, I'd rerun these labs with more gusto. In Lab 1, I breezed past word clouds—next time, I'd mess with custom stop words to see how the picture changes. In Lab 4, I stuck to default settings; I'd now play with hidden_size or learning_rate to feel out their effects, really owning the model. I'd also jot down questions as I went, building a cheat sheet to lock in what I learned.

New curiosities popped up: How do GloVe embeddings stack up against Word2Vec or FastText? Could an LSTM in Lab 4 handle longer reviews better than the RNN? And how do you scale this to millions of documents without everything crashing? The labs teased transformers like BERT, and now I'm itching to dig into those.

In the grand scheme of machine learning, these labs are a stepping stone. Lab 1's preprocessing is the unsung hero of any text system. Lab 2's BoW shows off early, readable methods, while Lab 3's embeddings and Lab 4's RNNs nudge toward deep learning's rise. They're not the bleeding edge—transformers rule now—but they're the practical roots, hands-on lessons that peel back the layers of a field that's equal parts science and art.

Conclusion

These four labs wove together a tapestry of technical growth and personal discovery, threading NLP's essentials—from text cleanup to neural networks—into a narrative I can call my own. They taught me not just how to process words or train models, but how to wrestle with complexity and emerge sharper. The journey from confusion over stemming to confidence in tuning an RNN mirrors a broader shift: I'm no longer just a bystander to machine learning—I'm a participant, armed with tools and questions to push further. What lingers is a hunger to explore more—be it transformers or real-world applications—knowing that each challenge is a chance to grow.