ECE 492-55 or ECE 592-106
Foundations of Generative AI for Systems

Instructor(s): Dr. Samira Mirbagher Ajorpaz, Department of Electrical & Computer Engineering

Email: smirbag@ncsu.edu

Objective or Description: This course provides a comprehensive exploration of AI techniques and their application in computer architecture and systems. Through hands-on projects, students will engage with foundational and advanced machine learning concepts to address real-world challenges in system design and optimization. The course emphasizes practical understanding and application, focusing on how generative AI can enhance performance in areas such as cache replacement policies, predictive load balancing, and hardware-based malware detection. Students will gain experience with generative models, large language models (LLMs), neural networks, reinforcement learning, and feature engineering. By the end of the course, they will be equipped with the knowledge and skills to apply machine learning for impactful solutions in system performance, optimization, and resource management.

Prerequisites: Computer Architecture (for 592),
Data Structures/Algorithms (for 592 & 492),
C++, Python (for 592 & 492).

Textbook & Resources: Selected readings from research papers (provided) and following textbooks:

Machine Learning:

1. *Machine Learning* by Tom Mitchell
2. Deep Learning, Ian Goodfellow and Yoshua Bengio and Aaron Courville
3. *Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play*
4. *Generative AI with LangChain: Build Large Language Model (LLM) Apps*
5. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*

Computer Architecture:

1. *Modern Processor Design: Fundamentals of Superscalar Processors* by John P. Shen and Mikko Lipasti
2. *Computer Architecture: A Quantitative Approach* (Morgan Kaufmann Series)

Topics: The course incorporates a wide range of machine learning models and system-level applications in computer architecture. Students are exposed to applications of ML in computer architecture through homeworks & projects, they learn to formulate architecture problems as an ML problem. Below is a comprehensive breakdown of the models and topics covered, along with the primary machine learning and system-related research papers targeted in homework and lectures.

Machine Learning Models and Techniques

1. **Foundational Models and Techniques:**
   - Perceptron & Single-Layer Neural Networks: Basic concepts, training rules, and applications for cache replacement.
   - Multi-Layer Perceptrons (MLP): Introduction to deeper networks, weight updates, activation functions, and supervised learning.

- Recurrent Neural Networks (RNN): Sequence modeling for tasks in cache replacement and resource prediction; includes backpropagation through time.
- Long Short-Term Memory Networks (LSTM): Memory-based sequence learning, specifically tailored for tasks requiring temporal dependencies in cache prediction.
- Self-Attention and Transformers: Mechanisms for learning dependencies across time in system tasks, including context vector computation and dot-product attention.
- Variational Autoencoders (VAEs): Probabilistic modeling for generating **representations, focusing on applications in dynamic hardware resource management.**

2. **Advanced Generative and Neural Models:**
    - Generative Adversarial Networks (GANs) & Conditional GANs (CGANs): Adversarial training and applications in system simulation and dynamic cache modeling.
    - Retrieval-Augmented Generation (RAG): Combines retrieval mechanisms with LLMs to enhance contextual understanding in cache replacement policies.
    - Large Language Models (LLMs): Applications of LLMs in structured reasoning, table-based tasks, and interpretability in cache replacement strategies.
    - Scaling for Large Models (e.g., BitNet): Efficient computation techniques like BitNet for reducing model complexity while maintaining interpretability in system tasks.
3. **Trustworthiness and Interpretability in ML for Systems:**
    - Attention Mechanisms in High-Level Program Analysis: Leveraging Glider's Section 4.3 to plot attention layers across varied embeddings for insight retrieval.
    - Emergent Abilities in LLMs: Zero-shot learning, in-context learning, and scaling impacts on emergent behavior.

ML and System Papers & Topics

Homework and class materials are centered on papers exploring machine learning models for practical system-level applications, including predictive caching, memory access optimization, and branching strategies. Below is a selection of system topics and target papers covered:

1. **Cache Replacement Policies and Predictive Modeling:**
    - **PARROT (Predictive Cache Replacement):** PARROT's use of neural architectures like LSTMs and embeddings to predict cache replacement more effectively than traditional policies.
    - **Multiperspective Reuse Prediction:** Explores how multiple perspectives and feature engineering impact predictive accuracy in reuse prediction.
    - **CHIRP (Control-Flow History Reuse Prediction):** Uses historical control flow data to predict cache and TLB management needs in CPU architectures.
    - **PerSpectron:** Investigates invariant memory access patterns for microarchitectural attacks, using perceptrons for footprint prediction.
2. **Branch Prediction and Instruction Prediction Policies:**
    - **TAGE Branch Predictor:** A state-of-the-art branch prediction model combining path-based and hashing techniques.
    - **Bit-Level Perceptron Prediction for Indirect Branches:** Applies perceptrons at the bit level for improved indirect branch prediction accuracy.
3. **Load Prediction and Memory Access Optimization:**
    - **Hermes:** Focuses on accelerating long-latency memory accesses using perceptron-based load prediction.
    - **Fast Path-Based Neural Branch Prediction:** Applies neural techniques to predict instruction flow in dynamic branching.

4. **Reinforcement Learning for Cache and Load Management:**
    - **Reinforcement Learning with Human Feedback (RLHF):** Framework to improve RL algorithms with human feedback, focusing on dynamic load and cache management.
    - **Exploring Predictive Replacement Policies for Instruction Cache and Branch Target Buffer (ISCA 2018):** Detailed comparison of predictive caching techniques using RL to enhance instruction cache efficiency.
5. **Dynamic Power Management and Prefetching:**
    - **Merging Path and Gshare Indexing in Perceptron-Based Prefetch Filtering:** Uses perceptron models to filter prefetch requests based on history patterns and current load characteristics.
    - **Perceptron-Based Prefetch Filtering:** Leveraging perceptron's decision-making for refined prefetch accuracy to optimize memory usage.
6. **Analysis of ML Interpretability Techniques in Systems:**
    - **Attention Layer and Program Semantics Analysis (Glider Paper Section 5.5):** Uses attention mechanisms to capture program-level semantics in ML models for system analysis.
    - **EVAX:** Examines proactive adaptive architecture techniques for secure high-performance systems using ML-driven predictions.

Grading: The grading for this course is based on a mix of homework assignments, projects, and presentations. Undergraduates have an option to replace the project with a final exam.

	Percentage of Final Grade
Homework 1	**8**
Homework 2	**12**
Homework 3	**15**
Class Presentation	**10**
Class Participation + Quizzes	**10**
Final Project	**25**
Final Exam	**20**

(*492 can select to be graded based on homework, exam, project or all three whichever makes their grade higher.)

Cross-listing in other departments: Cross listed with CSC

Fall 2024 Schedule:

Aug 19	Class Introduction	Overview of course structure, grading, and ML and architecture topics.	
Aug 21	Processor Design Background	Top-down method for performance analysis and counters.	
Aug 23	Foundations in ML	Dot Product, Matrix Multiplication, and Linear Layers.	
Aug 26	McCulloch-Pitts Model, Perceptron & Gradient Descent	Building Boolean logic, Perceptron model, XOR problem, and backpropagation. Perceptron training rule, batch and stochastic modes, and gradient calculation for sigmoid.	
Aug 28	Activation Functions, Softmax Function	Overview of ReLU, Sigmoid, Tanh, and Softmax functions for neural networks. Application in multi-class classification, translation invariance, and practical exercises.	
Sep 4	Multi-Layer Perceptrons (MLP), Gradient Computation, Discussion of mathematical foundation in MLP	Layer-by-layer exercise on MLPs with a focus on weight updates and activation functions. Derivative calculation, backpropagation examples, and gradient exercises with ReLU.	
Sep 6	Introduction to RNNs	Transition from feedforward to recurrent networks, and RNN limitations.	
Sep 9	Advanced RNNs	BPTT, and RNN limitations.	
Sep 11	LSTM Networks	Detailed LSTM cell architecture, forward pass, cell state updates, and practical applications.	Long Short Term Memory
Sep 13	Self-Attention Mechanism	Transformers, dot-product attention, scaling, softmax normalization, and context vectors.	Attention Is All You Need
Sep 16	Predictive Policies for Cache Replacement Deep Learning for Cache Replacement	Dead block prediction, LRU policies, feature engineering, and metadata management in cache. Applying deep learning models to optimize cache policies, with exercises. Presentation of baseline policy reproduction, introduction to advanced replacement algorithms.	Applying Deep Learning to the Cache Replacement Problem

Sep 18	Reinforcement Learning & RLHF	Exploration of reward systems, Q-Learning, RL with Human Feedback, and real-world applications.	Q-Learning,
Sep 19	Homework 1 Due: Sep 19, 2024, 11:59 PM	Summary: The PARROT Approach for Cache Replacement - Objective: Establish a working environment, set up simulation, understand the codebase, and reproduce baseline results of the PARROT approach. - Learn to formulate cache replacement policy as a Imitation Learning problem: Understanding cache replacement policies through imitation learning frameworks. - Key Tasks: Simulation setup, workload configuration, mathematical explanation of ranking loss, and system diagrams. - Submission: LaTeX report with code snippets, diagrams, results, interpretation, and a GitHub contribution report.	
Sep 21	CNN	Backpropagation in CNNs applied to digit classification, feature extraction, convolutional layers, pooling.	Backpropagation Applied to Handwritten Zip Code Recognition [CNNs]
Sep 23	GANs and Conditional GANs	Overview of GAN structures, adversarial loss, and applications in system modeling.	General Adversarial Nets, **Conditional Generative AI (https://arxiv.org/abs/1411.1784)**
Sep 25	Hands on exercise GAN		
Sep 27	Autoencoding Variational Bayes	Introduction to VAEs, latent space, and optimization for probabilistic reasoning.	
Oct 2	Hands on exercise AVE		
Oct 4	Emergent Abilities of LLMs	Exploration of emergent behaviors in LLMs and their applications.	
Oct 7	LLMs for Table Reasoning	Few-shot reasoning with LLMs for structured data processing.	Large Language Models are few(1)-shot Table Reasoners
Oct 9	Chain of Thought (CoT) & Graph of Thoughts (GoT)	Prompting techniques in LLMs to solve complex problems in structured reasoning.	Chain-of-Thought Prompting Elicits Reasoning in Large Language Models Graph of Thoughts: Solving Elaborate Problems with Large Language Models,
Oct 11	Advanced LLM Scaling Techniques	Exploring pathway scaling, the impact on performance, and model accuracy.	

Oct 16	LLM Reasoning		**Towards Reasoning in Large Language Models: A Survey**
Oct 18	LLM Reasoning		**Towards Reasoning in Large Language Models: A Survey**
Oct 21	Retrieval-Augmented Generation (RAG)	RAG in LLM tasks, using vector databases for knowledge augmentation. Hands-on exploration of RAG methods.	**Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks Excercise by hand https://aibyhand.substack.com/p/vector-database-spreadsheet**
Oct 23	Homework 2 Due: Oct 23, 2024, 11:59 PM	Homework 2: Space Exploration and ML/Non-ML Replacement Policies - Objective: Implement and test single perceptron, MLP, and RNN-based cache replacement policies. Retrieval-Augmented Generation tasks for improved system responses. - Tasks: Workload selection, MPKI comparisons, RAG development, and analysis of embeddings and attention layers. Hands-On with RAGs & LangChain. - Submission: LaTeX report with results analysis, bar charts, and comparisons with baseline methods. Data Compilation for LLMs: Organizing datasets, adding metadata, and preparing RAG files for analysis.	
Oct 25	Residual Networks and CLIP Models	Residual connections in CNNs, large language models for image-text applications.	**Residual network , Hands-on Excercise (https://aibyhand.substack.com/p/16-can-you-calculate-resnet-by-hand)**
Oct 28	BitNet Scaling for LLMs	BitNet transformation, efficient LLM computation, and 1-bit scaling.	**BitNet: Scaling 1-bit Transformers for Large Language Models (hands on exercise) The Era of 1-bit LLMs: All Large Language Models are in 1.58 Bits**
Nov 1	ML application Paper		**Ithemal: Accurate, Portable and Fast Basic Block Throughput Estimation using Deep Neural Networks**
Nov 4	ML application Paper		**DiffTune: Optimizing CPU Simulator Parameters with Learned Differentiable Surrogates (Do not change this paper please, highly recommended)**
Nov 6	Reinforcement Learning for Hardware Optimization	Implementing RL models for dynamic hardware resource management.	Practical Online Reinforcement Learning for Microprocessors With Micro-Armed Bandit

Nov 8	Advance RNN, LSTM and Attention mechanism Review		
Nov 11	Homework 3 Due: Nov 11, 2024, 11:59 PM	Homework 3: LLM Applications in System Cache, Trustworthiness, Interpretability, and Insight Retrieval in AI for Systems - Objective: Exploring LLMs as cache replacement agents and benchmarks for effectiveness. Examine trustworthiness and insight retrieval through attention layers and embedding comparisons. - Tasks: Review section 4.3 of the Glider paper for attention plotting, explore varied embeddings, compare with prior cache replacement policies, and analyze feature engineering. - Submission: Detailed report with findings, system diagrams, attention layer plots, and comparison charts.	
Nov 13	Project Proposal Due by email	All final project proposals must be submitted, with detailed outlines and initial research.	
Nov 15	Genetic Algorithms in System Design	Introduction to Genetic Algorithms.	
Nov 20	Continuation: Genetic Algorithms	Genetic algorithms for system optimization, evolutionary search for parameter tuning.	
Nov 29	Review and Discussion	Final project reviews, feedback sessions, and wrap-up discussions.	
Dec 11	Final Project Presentation In Person, Teams.	Project presentations with prize announcements for top projects.	
Dec 12	Final Project Report Due by email	Complete project report with code, RAG, analysis.	