UMC 203 2025 Term paper

March 13, 2025

Project Guidelines

Team Formation

We will have sixteen teams of four. Please fill out the attached Microsoft Form with your team's details and topic preferences.

Project Selection

Please carefully review all the project pitches provided below. Take the time to sift through them and identify the projects that resonate with your team. Consider reading the abstracts of the provided papers and reading about the respective areas online before finalizing your decision. Create a preference list, ranking all 16 projects in the order your team finds the most appealing.

The Project

- Teams are expected to read and understand the suggested papers and related references.
- Implement code for some of the experiments in the papers and compare different methods where possible.
- You will be pointed to the existing code where it is available. Some of them are already linked in the document.

Details for deliverables and timeline will be announced shortly. This may be your first time reading cutting edge research papers, here is a video from Andrew Ng that may help you: Lecture 8 - Career Advice / Reading Research Papers. Although you are free to find your own style.

1 NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis

NeRF (Neural Radiance Fields) is a deep learning model that generates high-quality 3D scenes from 2D images. It represents a scene as a continuous function, mapping 3D coordinates and viewing directions to color and density values. NeRF is trained using a set of posed images and can synthesize novel views with realistic lighting and details. It works by optimizing a volumetric rendering function using a neural network, making it particularly useful for applications like 3D reconstruction, virtual reality, and synthetic view generation.

References

- Papers: NeRF
- Code: Github
- Resources: NeRF Project Page, video,

2 Introduction to Robotic learning, Control & PPO Algorithms

Perception, Planning and Control are the three pillars of robot autonomy. The state of the art technologies fueling these tasks include environment mapping, robot localization, model predictive control, model based and model free reinforcement learning. Advancements in deep learning, probabilistic modeling and optimal control have significantly enhanced robotic capabilities, enabling them to operate efficiently in unstructured and dynamic environments. In this project you will analyse and apply the PPO algorithm for specific control tasks like quadruped locomotion, robotic arm control etc, using advanced simulation softwares.

References

- Papers: LocoTransformer (LT), Agile but Safe Locomotion of Quadruped Robots (ABS), Proximal Policy Optimization Algorithms (PPO)
- Code: LT, ABS
- Resources: OpenAI Gymnasium, IsaacGym, Google Deepmind - MuJoco Menagerie.

3 Analyzing Inverse Problems with Invertible Neural Networks

In natural sciences, determining hidden system parameters from measurements often involves a well-defined forward process but an ambiguous inverse problem, where one measurement can correspond to multiple parameter sets. Invertible Neural Networks (INNs) are particularly suited for this task, as they learn both the forward and inverse processes simultaneously, using latent variables to capture otherwise lost information. Unlike traditional neural networks, INNs provide a full parameter distribution for a given measurement and sampled latent variables. You can use this idea to simulate a simple inverse kinematics problem

References

- Papers: INNs, Local INN paper Real NVP
- Code: INNs, Local INN
- Resources: FrEIA, FrEIA code

4 Flow Based Models

Flow-based models are a type of generative model that learn to transform simple probability distributions (like a Gaussian) into complex ones, making them great for generating realistic data. They use a sequence of invertible transformations, meaning they can both generate new samples and compute exact probabilities, unlike GANs or VAEs. Since every transformation is reversible and has a known Jacobian determinant, training is efficient using maximum likelihood estimation. . These models are useful in image synthesis, density estimation, and more.

References

- NICE: `https://arxiv.org/abs/1410.8516`
- RealNVP: `https://arxiv.org/abs/1605.08803`
- GLOW: `https://arxiv.org/abs/1807.03039`
- Code: Glow, RealNVP

5 Kolmogorov Arnold Networks

In 2024, researchers developed Komogorov Arnold Networks (KANs) and claimed that "theoretically and empirically, KANs possess faster neural scaling laws than MLPs", "For interpretability, KANs can be intuitively visualized and can easily interact with human users" and "KANs are promising alternatives for MLPs". In this project, you will test these claims.

References

- Papers: Original Paper, A survey, Kolmogorov Arnold Transformers
- Code: Pykan, kat
- Resources: Kolmogorov Arnold Representation Theorem Wiki, Awesome KAN

6 Dimensionality Reduction

Much of modern machine learning deals with data (and consequently, models) whose feature dimensions tend to exceed the number of samples. This results in the curse of dimensionality, which manifests itself in sub-optimality of classifiers, exponential growth of sample complexity, and a breakdown of much of the intuition that we've built for three dimensions. For this project, you will study algorithms for dimensionality reduction, conducting explorations in linear, randomized, and non-linear methods, before progressing to neural network-based approaches for representation learning.

References

- Papers: Random projection in dimensionality reduction: applications to image and text data; Visualizing Data using t-SNE; Gaussian Mixture Variational Autoencoder with Contrastive Learning for Multi-Label Classification
- Code: scikit-learn's user guide for classical DR; Reference repository for VAE implementations; The official CGM-VAE repository
- Resources: Larry Wasserman's notes; C.J.C. Burges's Guided Tour

7 Decision Transformers

Decision Transformers (DT) are a type of transformer model used for reinforcement learning (RL) and sequential decision-making tasks. Instead of learning a policy in the traditional way using reward maximization, Decision Transformers treat decision-making as a sequence modeling problem, similar to how transformers process natural language. The model is trained in a supervised learning fashion using offline RL data (datasets of past trajectories collected from other policies). These sequences are fed into a causal transformer, which learns to output the next action given the current return-to-go and state.

References

- Papers: Decision Transformer: Reinforcement Learning via Sequence Modeling
- Code: Decision-Transformer, min-Decision-Transformer
- Resources: Medium Blog Post, Huggingface Blog Post

8 Conformal Prediction

Neural networks do not need to be black boxes. Conformal prediction is a theoretical framework for quantifying uncertainty in the predictions made by arbitrary prediction algorithms by converting an algorithm's predictions into prediction sets with provable guarantees. In this project, you will study conformal prediction with rigour and apply them to real world models.

References

- Papers: A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification, Uncertainity Sets for Image Classifiers using Conformal Prediction, Class-Conditional Conformal Prediction with Many Classes
- Code: Uncertainity Sets for Image Classifiers using Conformal Prediction, Class Conditional Conformal Prediction
- Resources: Conformal Prediction Wiki, Ryan Tibshirani's notes, A tutorial on Conformal Prediction, Awesome Conformal Prediction

9 Learning Image Restoration without clean data

A image denoising approach that introduces a deep earning-based denoising method that demonstrates how neural networks can be trained to remove noise from images without requiring clean (noise-free) images as ground truth. Instead of using pairs of noisy and clean images, the method relies on pairs of noisy images, making it highly useful in real-world scenarios where obtaining clean images is difficult. The Noise2Noise paper does not introduce a new architecture but leverages well-known convolutional neural network (CNN) architectures for denoising. One of the most effective architectures used in the paper is U-Net, which is widely used in image restoration, segmentation, and denoising tasks

- Papers: Noise2Noise: Learning Image Restoration without Clean Data, Zero-Shot Noise2Noise: Efficient Image Denoising without any Data
- code: Noise2Noise
- Resources: Nvidia page

10 Introduction to State Space Models

SSM's are sequential models inspired from control theory that generalized the time series models RNN's,CNN's, Transformers, Kalman filters etc (view them as a special case of SSM). Also unlike other DNN architectures some of these SSM's with Structured and Selective mechanisms have better explainability and also perform well in various applications like NLP, Speech, CV and Robotics.

Motivation

Structured and selective parameterisation of SSM's can inherit the advantages of various time series models. For example RNN's are very fast in inference but struggles with slow training due to recurrence structure, exploding and vanishing gradients problem and also they can't capture long time dependencies. On the other hand Transformers can be trained fastly but they are slow at inference. Mamba-SSM combines the advantages of both models with fast inference and training along with additional benefit of capturing long time dependencies. similarly there are lots of SSM's like LSSL, S4, S5, S6 (Mamba), DSS, etc. To understand these SSM's better with strong mathematical background it will be very beneficial to start with the following literature attached in reference (I strongly recommend you to read and understand the mathematics fueling the SSM's in LSSL and HIPPO and then choose an architecture S4/ S5/ S6 to get handson using code repositories).

References

- Papers: LSSL, HIPPO, S4, S5, S6/ Mamba,
- Code: S4 & S6/mamba, S5
- Resources: Introduction to SMM's, Hippo & S4 , A visual guide to Mamba.

11 Ensemble Methods

Ensemble methods are approaches to constructing classifiers through combining a set of "smaller" or "simpler" learners. Ensemble models come with a bunch of benefits, since they have the potential to mitigate the probability of a learner selecting the incorrect hypothesis from your hypothesis space. In this project, you will study ensemble learning, tracing its progress through the early statistical approaches of the late 20$^{\text{th}}$ century, all the way to modern Mixture-of-Experts approaches for today's deep architectures.

References

- Papers: Random Forests; XGBoost: A Scalable Tree Boosting System; Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity
- Code: scikit-learn's ensembles API; the first of many lucidrains MoE PyTorch implementations
- Resources: A Survey on Mixture of Experts; A Survey of Ensemble Learning; Explaining AdaBoost; Zhi-Hua Zhou's book on Ensemble Methods

12 Variational Autoencoders

Variational Autoencoders (VAEs) are a type of generative model that learn to encode data into a lower-dimensional latent space and then decode it back to its original form. Unlike standard autoencoders, VAEs introduce randomness by mapping inputs to a probability distribution, allowing them to generate new, realistic samples. They achieve this using a probabilistic encoder-decoder setup trained with a loss function that balances reconstruction accuracy and latent space regularization. They are widely used in image generation, data compression, and unsupervised learning.

References

- VAE: https://arxiv.org/abs/1312.6114
- VQ-VAE: https://arxiv.org/abs/1711.00937
- Info-VAE: https://arxiv.org/abs/1706.02262
- Code: VAEs

13 Navigation Model for Robots

For this project we will use a robot navigation model (NoMaD) designed to handle both goal-directed navigation and goal-agnostic exploration.
NoMaD is a goal-conditioned robotic navigation framework that combines EfficientNet + Transformer for perception and a Conditional diffusion model for motion planning. In this project we will primarly focus on the perception part.

References

- Papers: NoMaD, An Image is Worth 16x16 Words, EfficientNet
- Code: NoMaD,
- Resources: ,

14 Diffusion Models

Diffusion models are generative models that learn to create data by gradually denoising a sample, reversing a process that adds noise step by step. Inspired by thermodynamics, they start with pure noise and iteratively refine it into a meaningful structure using a learned model. Diffusion models excel at generating high-quality images and have outperformed GANs in realism and diversity. Popular examples include DALL·E 2 and Stable Diffusion, which power modern AI-generated art and image synthesis.

References

- DDPM: https://arxiv.org/abs/2006.11239
- DDIM: https://arxiv.org/abs/2010.02502
- Survey: https://arxiv.org/abs/2403.18103
- Code: DDIM

15 Vision Transformers

In this project we will focus on EfficientFormer v2 a lightweight Vision Transformer (ViT). EfficientFormer v2 was developed to address the limitations of Vision Transformers in real-time applications, particularly on resource-constrained devices like mobile phones. While ViTs have demonstrated impressive performance in various computer vision tasks, their substantial parameter counts and computational demands often result in slower inference speeds compared to lightweight convolutional neural networks (CNNs). EfficientFormer aims to combine the strengths of transformers with the efficiency of models like MobileNet, achieving high accuracy while maintaining low latency suitable for deployment on devices with limited computational resources.

References

- Papers: EfficientFormer v2, EfficientFormer v1, An Image is Worth 16x16 Words, MobileNets
- Code: EfficientFormer,
- Resources: ,

16 Unsupervised Domain Adaptation

Unsupervised domain adaptation (UDA) is a type of domain adaptation in machine learning where a model is trained on a source domain with labelled data, and then adapted to a target domain with unlabelled data. In UDA, the source domain and target domain have different distributions, but the goal is to leverage the labelled data in the source domain to improve performance on the target domain. In supervised learning, the model is trained on labelled data in the same domain as the test data. However, in real-world scenarios, it is often the case that the labelled data is not available in the target domain. UDA attempts to address this issue by learning a model that can generalize to the target domain using only unlabelled data. You will be working with image datasets to address the distribution shift among dataset domains.

References

- Domain-Adversarial Training of Neural Networks: `https://arxiv.org/pdf/1505.07818.pdf`
- A theory of learning from different domains: `https://link.springer.com/article/10.1007/s10994-009-5152-4`
- A Survey of Unsupervised Deep Domain Adaptation: `https://dl.acm.org/doi/pdf/10.1145/3400066`