ECE 661 COMP ENG ML & DEEP NEURAL NETS

COURSE PROJECT INTRODUCTION

HAI "HELEN" LI, SPRING 2025

General information

- Please check the following files on Canvas and read them carefully.
  - "Project Topic List-25SP.pdf"
    - 8 topics for you to choose from
    - You are highly encouraged to propose your own topic.
  - "Project Guidelines-25SP.pdf"
    - Grading Rubrics, Submission Guidelines
- Important date
  - Proposal due by **March 12**
  - Poster session on **April 23 (Wednesday) 1:00 pm-4:00 pm**.

General Information

- Project Group
  - All projects **should** be done in a group of **3 students**. Please contact TA for difficulties finding a group.
  - We expect each student to have a similar workload in the project. Each team member should be assigned **balanced duty**, and you should state this clearly in your proposal/report.
  - For a team with 1-2 people, we will **NOT** lower our expectations in basic requirements. (**Teamwork is strongly encouraged**!)
  - Having 4 people in a group is **discouraged**. We will raise our expectations for a group of 4.
  - Having more than 4 people in a group is **Not Allowed**.
- Register your group with below link (Due on **March 5**)
  - **Only one submission each Group**
  - ECE661-25SP Project Group Registration

General information

- Deliverables of the project
  - **Proposal**: Due on **March 12 11:59pm** (hard deadline, no late days can be used.) We will return you the comments on Canvas.
  - **Midterm Checking-in (Optional):** Due on **April 2**, update your project progress and discuss with your TA.
  - **Presentation:** Poster session on **April 23 (1-4pm),** present in front of the whole class, followed by Q&A. Peer interaction and review will be included in the poster session
  - **Report**: Due on **April 25 11:59pm**, should be in a technical report format talking about what have you done and what results you get. Check "project format" for template.

General information: Ethics

- Your project **MUST** adhere to ethical standards for responsible research practice and due diligence in the conduct.

- If your project uses human-derived data, consider whether that data might:
  - Contain any personally identifiable information or sensitive personally identifiable information.
  - Contain information that could be deduced about individuals that they have not consented to share.
  - Encode, contain, or potentially exacerbate bias against people of a certain gender, race, sexuality, or who have other protected characteristics.
  - Have been discredited by the creators.

https://nips.cc/public/EthicsGuidelines

General information: Expectations

- In provided Topics there will be two requirements:
- **Basic requirements**
  - **Minimal** requirement to make the project “Acceptable”, for both presentation & report
  - Aligned and calibrated with a 3-person team
  - If you would like to change basic requirements, **discuss with TAs** and reflect the change in the proposal.

- **Optional requirements (MUST-Complete, at least One)**
  - At least one for 3-person team, at least 2 for 4 person team
  - Additional explorations on the same project topic. Can be either empirical or intuitive.
  - **DO NOT** burn resources for incremental explorations on optional requirements. For example, **DO NOT** run 100 sets of hyperparameters on the same model and justify gain contributions

Grading Criteria

- Final Grade
  - Proposal(10%)
  - Poster Session Poster (50%)
    - Poster (15%)
    - Presentation (15%)
    - Q&A (20%)
  - Final Report (40%)

- Detailed Rubric in “Project Guidelines” file
  - 1-3-6-8-10 Grading system
  - Borderline: (Meet all expectations)

- The grading depends on **what you complete** and **how you present**.

Project proposal

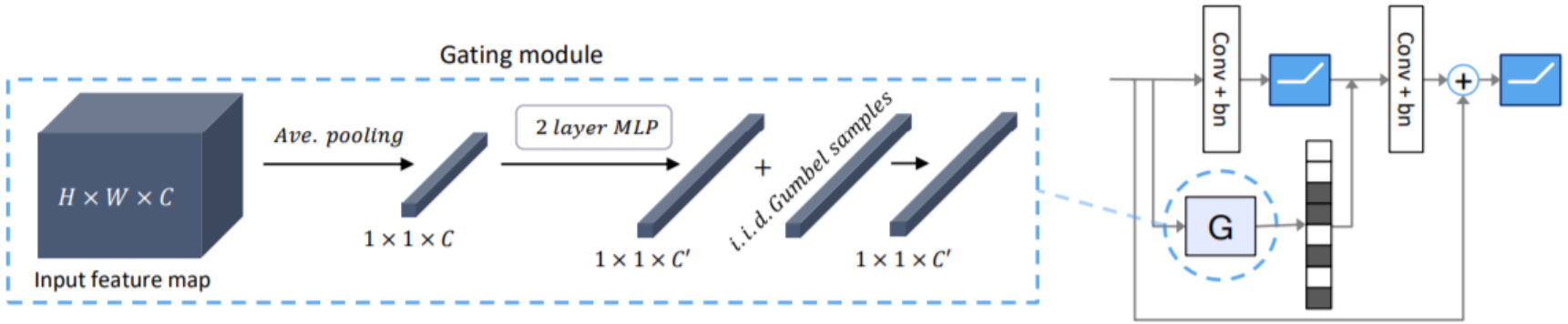
- Basic information of your project and group
- Heilmeier Catechism
  - What are you trying to do?
  - How is it done today?
  - Your approach and why do you think it will be successful?
  - What are the risks?
  - How long will it take?
  - What are the final “exams” to check for success?
- Use your own word to describe

- It’s OK to not follow the requirement exactly for planning you project, as long as the final workload is similar.

- Let’s use project topic 1 as an example

#1 Dynamic CNN model

- Dynamic channel gating selects the channel to use given input, save computation while keep model flexibility.
- Dynamic convolution combine multiple copies of the convolution kernel dynamically, more flexible while having same computation cost.



- **Objectives**
  - Implement training algorithm
  - Observe activation pattern of the dynamic model
  - Check performance under adversarial attack/training

An example proposal of project 1

- Examine Dynamic CNN models
- Heilmeier Catechism
  - What are you trying to do? Examine the effectiveness of input-dependent dynamic models for model compression.
  - How is it done today? Explain a little bit of dynamic convolution, gated channel/filter selection, and batch-shaping.
  - Your approach and why do you think it will be successful? Use ResNet-20 model on CIFAR-10, replacing all the convolutional layers by the dynamic convolution layer, further implementing as Bejnordi's paper. There's no need to call all features in the final learned model to deal with each individual input.
  - What are the risks? Will the accuracy drop dramatically?
  - How long will it take? Schedule for 5 weeks
  - What are the final "exams" to check for success? Code implementing the method, table showing comparison results

Expectation this semester

- As usual, we provide a few topics ("old").

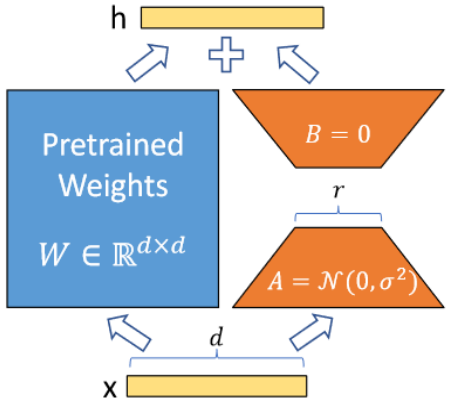
Project Topic List

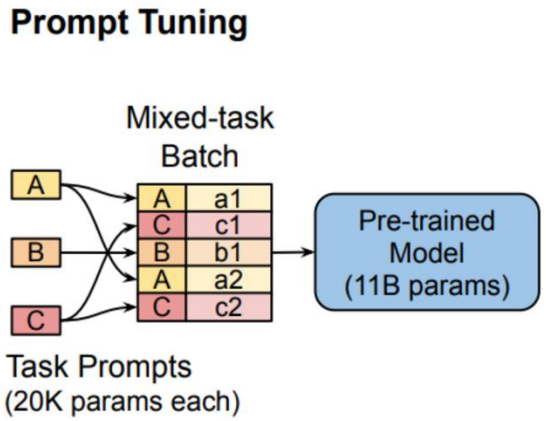
1. Input-dependent Dynamic CNN model
2. Robust and Non-robust Features
3. Improved Regularization of Convolutional Neural Networks
4. Self-Supervised / Representation Learning
5. Adversarial Patch Attack
6. Neural Network Distillation
7. Generative Adversarial Networks
8. Learned Non-linear Quantization
9. ***You own proposed project (talk to TA's to finalize the details)*** []
10. Appendix: More projects topics from previous semesters

- **We highly encourage you to propose your own ideas**
  - A new topic is more likely to get a better score.
  - Your own topic can fit your interest better.
  - Discuss with TA before proceed

Potential Topics

- Implement and compare across various parameter-efficient fine tuning (PEFT) methods.
  - Prompt tuning, LoRA, …
  - Compare best performance, performance vs epoch, latency per epoch, peak GPU memory, ...
  - Develop an application/scenario utilizing PEFT methods.
  - https://huggingface.co/docs/peft/en/





Potential Topics

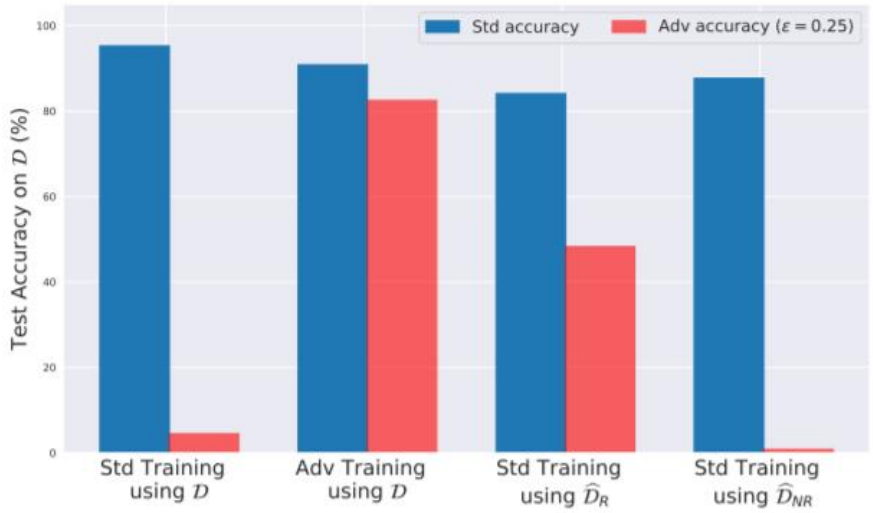
- Stable diffusion is now a popular technique for image generation.
  - Implement and compare U-net backbone diffusion, Diffusion Models with Transformers (DiT), and GAN-based methods.
  - Example applications: transfer photos of Duke campus into different genres, automatically draw comics for short stories, ...
  - https://github.com/CompVis/stable-diffusion
  - https://github.com/facebookresearch/DiT

- More introductions for the provided topics, check by yourselves later.

- Time for QA

#2 Robust and Non-Robust Feature

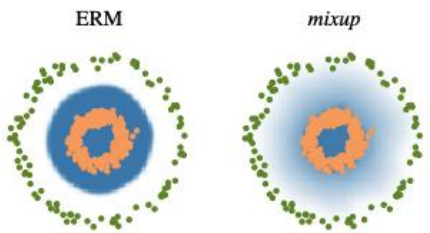
- Datasets contain both robust and non-robust features. Standard model mainly captures non-robust features while adversarial training learns robust features.



- **Objectives**
  - Distill robust/non-robust CIFAR-10 versions
  - Use qualitative and quantitative evaluation techniques to reproduce some of the findings in [Ilyas et al. 2019]
  - Test robust and non-robust models against adversarial attacks

#3 Improved Regularization of CNNs

- CNNs trained with standard cross-entropy often overfit to the training distribution.
- In this project, you will investigate and implement 3 regularization techniques to improve model generalization



- **Objectives**
  - Implement (1) cutout regularization, (2) mixup regularization, and (3) self-supervised rotation predictor.
  - Tune the hyperparameters and observe their effect on a ResNet architecture trained on CIFAR-10
  - Explore the effect of these regularizers when test data is corrupted

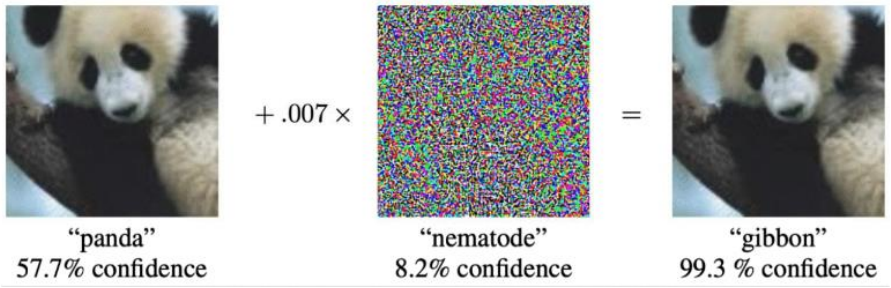
#4 Self-supervised learning

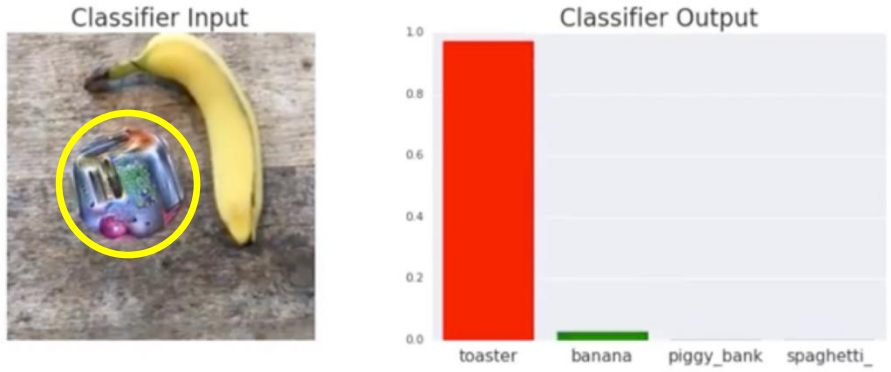
- CNN models can be good feature extractor after it is trained to perform classification (with labels). But what if there are no labels? Self-supervised learning is about learning useful representation from **unlabeled** data.

- **Objectives**
  - Implement and reproduce the results of a self-supervised learning framework (SimCLR) using ResNet-20.
  - Evaluate the effect of having limited labeled data. See how self-supervised learning compare with supervised learning.
  - Implement another algorithm (RotNet) and compare/contrast its performance to SimCLR.

#5 Adversarial patch attack

- Fooling neural networks with a carefully crafted patch/sticker.





- **Objectives**
  - Implement adversarial patch attack to mislead CIFAR-10 classifiers in both untargeted and targeted fashion.
  - Evaluate the effect of the patch size.
  - Evaluate the transferability of adversarial patch.

#6 Neural Network Distillation

- The main idea of distillation is to let a small model, usually referred to as student model, mimic the predictions of a large model, usually referred to as teacher model.

- **Objectives**
  - Implement the distillation and investigate several properties of it
  - Reverse the student model and the teacher model to verify if the distillation still works
  - Implement self distillation to prove that model can also learn from itself

#7 GANs

- “Generative” networks are trained to model a data distribution as to be capable of generating new samples from it. In this project you will create and train a conditional Generative Adversarial Network called the Auxiliary Classifier GAN (AC-GAN) to generate novel samples from the CIFAR-10 distribution and investigate a method for making GAN training process more stable
- **Objectives**
  - Implement the AC-GAN following instructions directly from the paper
  - Evaluate the quality of the generated samples both quantitatively and qualitatively
  - Implement the WGAN following instructions directly from the paper and compare its training process with original GAN
  - Evaluate the performance of all your GANs through quantitative metrices

#8 Learned Non-linear Quantization

- A non-linear quantization scheme can reach a higher performance comparing to a linear quantization

- **Objectives**
  - Read the LQ-Net paper and implement a ResNet-20 with the quantizer and network training algorithm from the paper
  - Explore accuracy and weight distribution under different hyperparameter setting
  - Try induce sparsity in the quantizer function to induce mixed-precision quantization, or apply sparsity as another objective to train jointly with LQ-Net

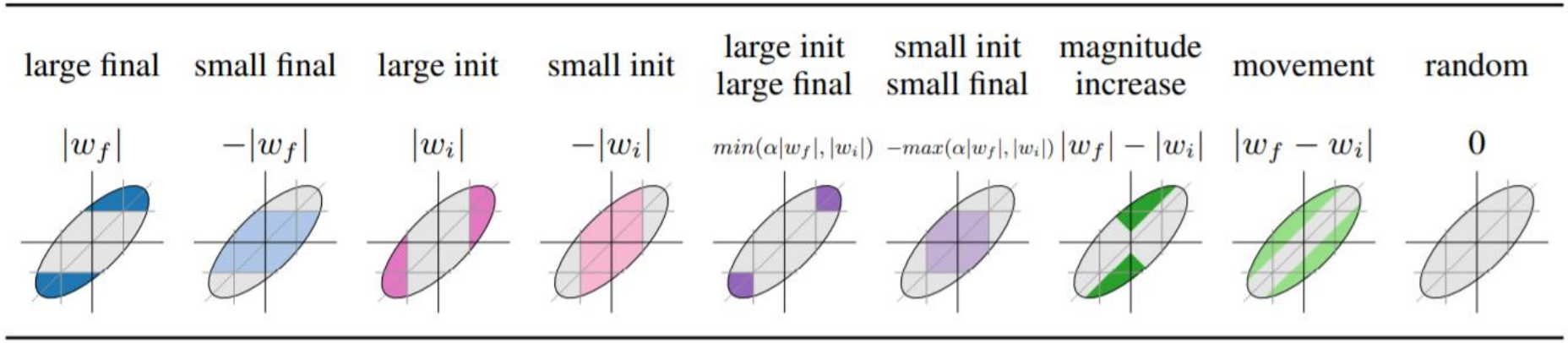
More possible topics

- You are **encouraged** to propose your own topics
  - You will get Higher Grade related to Innovation Part.

- We also append some other topics from previous semesters. (We may unable to provide enough guidance for these topics.)

#A Lottery ticket hypothesis

- Dense, randomly-initialized, feed-forward networks contain subnetworks (winning tickets) that—when trained in isolation—reach test accuracy comparable to the original network in a similar number of iterations.



- **Objectives**
  - Compare criteria for finding winning ticket models
  - Explore lottery ticket hypothesis in adversarial training
  - Finding lottery ticket with Continuous Sparsification

#B Noise-aware DNN training

- DNN performance under weight performance implies model generalizability, robustness to quantization and ability to run on certain hardware.
- Train DNN with adversarial or random noise presents in the weight.

Input: Training set $\mathcal{S} \triangleq \cup_{i=1}^{n}\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}$, Loss function $l : \mathcal{W} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$, Batch size b, Step size $\eta > 0$, Neighborhood size $\rho > 0$.
Output: Model trained with SAM
Initialize weights $\boldsymbol{w}_0$, $t = 0$;
while *not converged* **do**
  Sample batch $\mathcal{B} = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), ...(\boldsymbol{x}_b, \boldsymbol{y}_b)\}$;
  Compute gradient $\nabla_{\boldsymbol{w}} L_{\mathcal{B}}(\boldsymbol{w})$ of the batch's training loss;
  Compute $\hat{\boldsymbol{\epsilon}}(\boldsymbol{w})$ per equation 2;
  Compute gradient approximation for the SAM objective (equation 3): $\boldsymbol{g} = \nabla_w L_{\mathcal{B}}(\boldsymbol{w})|_{\boldsymbol{w}+\hat{\boldsymbol{\epsilon}}(\boldsymbol{w})}$;
  Update weights: $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \boldsymbol{g}$;
  $t = t + 1$;
end
return $\boldsymbol{w}_t$

Algorithm 1: SAM algorithm

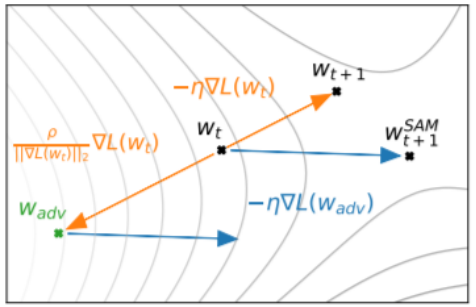


Figure 2: Schematic of the SAM parameter update.

- **Objectives**
  - Train DNN model under different weight perturbation
  - Evaluate model performance on generalization, under random weight perturbation and under quantization

#C Out-of-distribution detection

- What if the classifier encounter a sample that does not belong to any of the categories?

- **Objectives**
  - Implement several OOD detection algorithms on CIFAR-10 and evaluate their performance against various OOD datasets.
  - Implement a training algorithm and see how it can enable better detection.
  - See the effect of training data diversity on OOD detection for ResNet-20.

#D Transfer Learning for Domain Adaptation

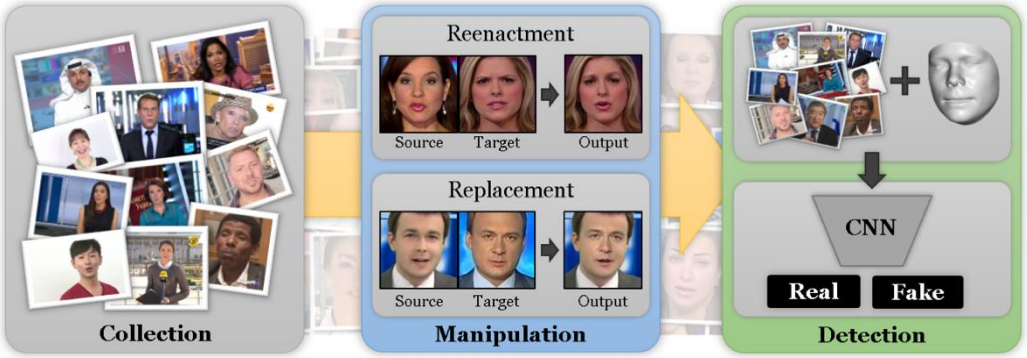
- Semantic segmentation refers to pixel-wise classification in order to identify specific objects of interest from images capturing holistic view of the scene. Unlike typical classification tasks where one can build their own models and train from scratch, image segmentation would require rather different model architectures (such as fully convolutional network), and would usually need transfer learning to jump-start the model training.
- **Objectives**
  - Work with a real-world medical image dataset and clean/pre-process them properly.
  - Play with mm-segmentation, a library that provides you access to many pre-trained segmentation models
  - Choose the best fine-tuning method for domain-adaptation on different model architectures, losses, etc.

#E Dynamic BlockDrop

- Not all images are equally difficult to recognize. It is intuitive that some simple images can be correctly classified with a smaller model, but the difficult ones have to be classified by a larger model. Some observations show ResNets behave like ensembles and dropping certain blocks of ResNets will not impact their performance.
- **Objectives**
  - Prove the ensembles like property of ResNets
  - Implement a per-instance network routing policy that will adaptively drop certain blocks in a neural network for each image

#F Deepfake Generation

- Fake face images can be generated by either face synthesizing or face manipulation.
- Face manipulation can be further categorized into face replacement (face swap) and face reenactment.
- Existing deepfake detection mainly focuses on passive detection.



- **Objectives**
  - Prepare datasets for Deepfake generation
  - Implement face replacement and face reenactment
  - Train a classifier to detect deepfake images
  - Discuss potential ways to defense against deepfake