ECE1512 2021 WINTER

ECE1512 – Project A: Visual Interpretation of Convoutional Neural Networks

Assigned:	**Monday**	**February 1**	**2021**
Due:	**Monday**	March 1	2021

Background

Machine learning (ML) has made a great breakthrough in various computer vision tasks in recent years. However, the solutions based on machine learning and deep learning (DL) mostly act like "black-boxes." Though promising, there is little insight into how and why they make the decisions they make. The lack of interpretability limits the utility of the solutions, especially in highly regulated areas such as medicine, finance, and law. It is unclear why consumers/users of the technology should trust these complicated solutions and accept the results and recommendations. Explainable Artificial Intelligence (XAI) attempts to address such issues by generating human-understandable explanations of the causes laying behind the ML-based decision-making models and "open" the black box of these cumbersome models.

Models can be inherently *explainable*, such as linear regression, naive Bayes models, or decision trees. However, more complicated models, such as the convolutional neural networks (CNNs) that are commonly used for image classification, tend to be too complex to be understood by humans and require either (1) an ad-hoc (or ante-hoc) approach (which integrates human-understandable assumptions into training) or (2) a post-hoc approach (which models the behavior of the target model after training has concluded).

Introduction

In this project, you will be exploring the problem of "Visual Explainable AI," which is a type of post-hoc XAI. This field aims to investigate local explanations for the models trained for array/image classification purposes in the evaluation phase. Through visualizing the input features that are the most responsible in the model's decision making, Visual Explainable AI can help the end-users to verify the trustworthiness of the predictions given by these models, diagnose their cases of misclassification, and reach an understanding of the evidence leading the model to make its decision.

Systematically looking at the solutions in this field, they take a trained machine learning-based model and a test array/image as input. The output of these solutions is a heat-map named "explanation map" that highlights each given input region, based on their importance for the model's decision-making procedure. Mainly, visual explanation methods, a.k.a. "*attribution methods*," can be categorized into the following groups:

a) **Approximation-based methods:** The methods that approximate the importance of the input features in the target model's decision making procedure (e.g., Local Interpretable Model-Agnostic Explanations (LIME), SHAP).
b) **Backpropagation-based methods:** The methods that operate by backpropagating the

signals from the output of the target model (e.g., Integrated Gradient, FullGrad, Layer-wise Relevance Propagation (LRP), etc.).

c) **Perturbation-based methods:** The methods that probe the behavior of the target model by feeding it with perturbed (masked) copies of the input (e.g., Randomized Input Sampling for Explanation (RISE), Semantic Input Sampling for Explanation (SISE), Extremal Perturbation, etc.).

d) **CAM-based methods:** The frameworks that visualize the perspective of the target model by performing combination of the abstract features extracted by the model. (These methods are specialized for explaining CNNs, and based on Class Activation Mapping (CAM) mehod, (e.g., Grad-CAM, Grad-CAM++, Ablation-CAM, etc.)

Objective

This project endeavors to generate explanation maps that interpret the behavior of two models trained on different datasets. The first model is a very shallow CNN trained on "**MNIST-1D**", a small-scale dataset prepared for generic array classification. The second model is a VGG-7 network trained on the "**HMT**" dataset utilized for histopathologic tissue classification. The detailed descriptions of the datasets are included in the next section. You may choose your favored algorithms to provide visual explanations for the model. You are supposed to discuss the methodology, advantages, and disadvantages of your selected method(s) in rich detail. You are also expected to evaluate and report the accuracy of your selected method(s) in explaining the given models.

Datasets

MNIST-1D:

- **Paper**: https://arxiv.org/abs/2011.14439
- **GitHub link**: https://github.com/greydanus/mnist1d
- **Objective**: Generic Image Classification
- **Description**: This dataset is a 1-Dimensional and low-memory analogue of the popular digit classification dataset, MNIST. In the same way as the **MNIST** dataset, the MNIST-1D data are divided into **10 classes**, each of which represents **a digit between 0-9**. Unlike MNIST, each example in MNIST-1D train/test data is **a one-dimensional sequence of points** generated by augmenting a 1-D template representing each of the digits by random padding, random translation, adding Gaussian noise, adding a constant linear signal analogous to shear in 2D images, and lastly, downsampling to 40 data points.
- **Availability:** Publically available (for academic purposes).
- **Resources needed:** CPU only
- **Data size**: 4000 train data + 1000 test data (partitioned by the dataset promoters).

HMT:

- **Paper**: Multi-class texture analysis in colorectal cancer histology https://www.nature.com/articles/srep27988
- **Description**: This dataset was formed to elevate the performance of ML-based solutions in "histopathological tissue classification." HMT is an equally-balanced dataset that contains images extracted from 10 independent samples of colorectal cancer (CRC) primary tumors and divided into one of the following 8 classes: (a) tumor epithelium, (b) simple stroma, (c) complex stroma, (d)

immune cell conglomerates, (e) debris and mucus, (f) mucosal glands, (g) adipose tissue, (h) background.

- **Availability:** Publically available (for academic purposes).
- **Resources needed:** CPU only
- **Data size**: 4504 train images + 496 test images.

Experimental Setup

1. Prerequisites:
   1.1. Python3
   1.2. keras (TensorFlow backend)
   1.3. sci-kit-learn (suggested)
   1.4. sklearn (suggested)
   1.5. numpy (suggested)
   1.6. matplotlib (suggested)
2. Download project_a_supp.zip from Quercus
3. Two CNN models trained on each of the mentioned datasets are provided. (alternatively, you may train your own CNN for classification)
   3.1. The first model, is a shallow CNN that is trained on MNIST 1-D dataset models/MNIST1D.h5.
   3.2. The second model, is a shallow version of the family of VGG networks (VGG-7), that is trained of HMT dataset models/HMT.h5.
4. mnist1d_utils.py is a Python file to re-create the MNIST-1D dataset. (Based on: https://github.com/greydanus/mnist1d)
5. MNIST1D.pkl is a "pickle file" containing all data in the MNIST-1D dataset. The data in this file is save as a dictionary that can be also created using the function make_dataset() in the library mnist1d_utils.py. More information to read the dictionary is provided in the notebook MNIST1D.ipynb.
6. hmt_dataset is a folder containing two subfolders, including the train and test set for the HMT dataset.
7. xai_utils.py is a Python file including utility functions needed for three state-of-the-art solutions in the field of visual XAI, Grad-CAM, RISE (Randomized Input Sampling for Explanation), and SISE (Semantic Input Sampling for Explanation).
8. HMT.ipynb and MNIST1D.ipynb are two Python notebooks showing 1) how the models are trained with each of the two described datasets, and 2) how the explanation algorithms included in xai_utils.py are applied on each of the described models.
9. In this project, you will not need to use any advanced device such as GPU in order to run your codes.

Part 1: 1-D digit classification

Task #1: 1-Dimensional digit classification [5 Marks]

1. Load the CNN trained on the MNIST 1-D dataset models/MNIST1D.h5, and the test data MNIST1D.pkl. Evaluate the performance of the trained model on the test data, assuming that the most-confident class is predicted for each image. Report the following [2.5 Marks]:
   a. Overall classification accuracy on the test set [0.5 Marks].
   b. Class-wise classification accuracy for all classes [0.5 Marks].

c. Plot the classification ROC and AUC curves for each class (More details regarding these metrics are available in the resources) [0.5 Marks].
d. Plot the normalized confusion matrix [0.5 Marks].
e. Precision, Recall, and F-1 score on the test set [0.5 Marks].

2. Show some qualitative examples of the success/failure cases of the model. Among which two classes misclassification happens the most? Provide your insights and support your answers with analytic reasons [2.5 Marks].

Task #2: CNN interpretation [10 Marks]

1. Among the attribution methods provided in xai_utils.py, or the other available options, select two methods (one, if you do the project in a group of one). Read the paper of the two existing methods, and answer the following questions [5 Marks].
   Note: at least one of the methods you select should be outside the three ones for whom the codes are provided. You can find a shortlist containing the references for some state-of-the-are attribution methods in the last page. Furthermore, considering the categorization provided in the introduction paragraph, you should not select two methods from same category.
   1. What research gap did your two chosen attribution methods fill? [0.5 Marks] What novelty did they contribute compared to their prior methods? [0.5 Marks]
   2. Explain in full detail the methodologies of your selected method(s). [2 Marks]
   3. Discuss the main advantages and disadvantages of your selected method(s). Do you think these methods can concretely interpret the target model in difficult scenarios (e.g., when the target model is a deep CNN or the input contains a high amount of texture or noise)? Do you think your selected method(s) can analyze and inspect the cases of misclassification by the target model? Why? [2 Marks]

2. Apply your selected method(s) on the CNN trained on the MNIST 1-D dataset, taking different inputs. [5 Marks]
   a. For each given input, your output should be a 1-dimensional explanation map that scores the input features based on their contribution to the model's prediction. [3 Marks]
   b. Qualitatively report the explanation maps you achieved, and compare them with the templates presented for each of the digits. Discuss your results. Do you think the highlighted region is similar to the template corresponding to the digit predicted by your model? Do you think the explanation map shows the local behavior of the model well? [2 Marks]

Part 2: Histopathological tissue classification

Task #3: Biomedical image classification and interpretation [5 Marks]

1. Load the CNN trained on models/HMT.h5, and the test set hmt_dataset/HMT_test. Evaluate the performance of the trained model on the test data, using the same metrics that are mentioned in **Task#1** [2.5 Marks].
2. Repeat the subsection 2.a of the **Task#2** on the HMT dataset, using the two XAI methods you previously selected [2.5 Marks].

Task #4: Quantitative evaluation of the attribution methods [10 Marks]

Different from so-called "**ground truth-based**" evaluation metrics such as mean Intersection over Union (mIoU) that compare the output of such algorithms with ground-truth masks, the concreteness and faithfulness of the attribution methods should be mainly assessed by another group of metrics named as "**model truth-based**," that verify the correctness of the explanations provided by an attribution method with the model's behavior. Model truth-based metrics (e.g., Insertion/Deletion, Drop/Increase) measure the relationship between the explanation maps generated by attribution methods and the target model's outputs.

In this task, you will evaluate the performance of your selected method, using a pair of model truth-based metrics, "**Drop%**" and "**Increase%**." This pair of metrics measure the decrease/incensement in the target model's confidence score when the target model is fed only with the most highlighted features by an explanation method, compared to when the whole input is given to the image. The intuition for these metrics are as follows:

- **Drop rate:** If we remove unimportant features from the input, the model's confidence score should not drop considerably.
- **Increase rate:** If we remove misleading features from the input, the model's confidence score may increase.

According to these metrics, low drop% and high increase% denote that the features assigned with a high score by the evaluated explanation method are highly considered in the model's decision making procedure. More details regarding these metrics can be found in the resources, the last subsection in the notebooks HMT.ipynb and MNIST1D.ipynb, and the figure 1.

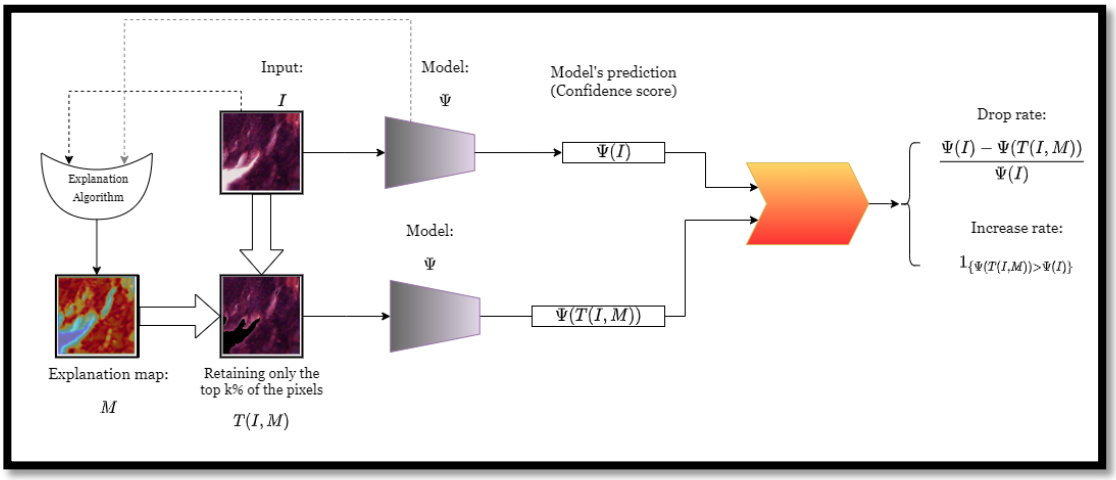

Figure 1: This figure depicts how the metrics "Drop" and "Increase" evaluate the correctness of the explanations generated by an explanation algorithm.

1. Apply the "Drop%" and "Increase%" metrics to evaluate the performance of your selected attribution method(s) when applied to the CNN trained on both datasets. For the MNIST-1D dataset, take the parameter "*k*" shown in *fig.1* as *30%*. This parameter can be tuned using the input argument `frac` in the function `calculate_drop_increase()`. Take this parameter for the HMT dataset as *90%*. For both datasets, calculate the average drop% and average increase% on the whole test set. [5 Marks].

2. Discuss in full details the qualitative/quantitative results you have achieved. Were your selected method(s) successful in interpreting the target model trained on the HMT and MNIST-1D dataset correctly? In what cases they fail to explain the target model's predictions? If you select two attribution methods, in what cases each one of them works better than the other one? Support all of your answers with detailed reasons [5 Marks].

Notes

1. Coding is expected for this assignment, and your code must be included in your submitted report – use the Python programming language
2. No machine learning models need to be trained for this assignment
3. External code may be used only if properly cited and if it were initially introduced as part of non-XAI research.
4. Include as many data visualization results as possible – a good visual is worth more than a thousand words (as per your ECE1512 Lecture 1 handout – "One picture is worth more than ten thousand words" (anonymous))

Resources

Certain concepts and methods used in this assignment may be unfamiliar to you. Refer to these online resources for more details (cite if code is used):

- Keras Installation: https://keras.io/#installation
- Training and test sets: https://en.wikipedia.org/wiki/Training,_validation,_and_test_sets
- Evaluating a pre-trained CNN in Keras: https://medium.com/@vijayabhaskar96/tutorial-image-classification-with-keras-flow-from-directory-and-generators-95f75ebe5720

- ROC curves: https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/
- Confusion matrix: https://www.geeksforgeeks.org/confusion-matrix-machine-learning/

Background on Explainable AI:

- https://christophm.github.io/interpretable-ml-book/shap.html
- https://explainer.ai/
- Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI https://www.sciencedirect.com/science/article/pii/S1566253519308103
- Explainable Artificial Intelligence: a Systematic Review https://arxiv.org/pdf/2006.00093.pdf
- The Mythos of Model Interpretability https://arxiv.org/pdf/1606.03490.pdf

Papers for visual explanation methods (you can browse for more papers):

- [LIME] "Why Should I Trust You?": Explaining the Predictions of Any Classifier: https://arxiv.org/abs/1602.04938
- [CAM]: https://arxiv.org/pdf/1512.04150.pdf
- [Grad-CAM] Gradient-weighted Class Activation Mapping: https://arxiv.org/abs/1610.02391
- [Grad-CAM++] Improved Visual Explanations for Deep Convolutional Networks: https://arxiv.org/abs/1710.11063
- [Abation-CAM] Visual Explanations for Deep Convolutional Network via Gradient-free Localization: https://openaccess.thecvf.com/content_WACV_2020/html/Desai_Ablation-CAM_Visual_Explanations_for_Deep_Convolutional_Network_via_Gradient-free_Localization_WACV_2020_paper.html
- [RISE] Randomized Input Sampling for Explanation of Black-box Models: https://arxiv.org/abs/1806.07421
- [SISE] Explaining Convolutional Neural Networks through Attribution-based Input Sampling and Block-wise Feature Aggregation: https://arxiv.org/abs/2010.00672
- [Integrated Gradient] Axiomatic Attribution for Deep Networks: https://arxiv.org/abs/1703.01365
- [Extremal Perturbation] Understanding Deep Networks via Extremal Perturbations and Smooth Masks: https://arxiv.org/abs/1910.08485
- [FullGrad] Full-Gradient Representation for Neural Network Visualization: https://arxiv.org/abs/1905.00780
- [LRP] Layer-wise Relevance Propagation: https://arxiv.org/abs/1604.00825
- [SHAP]: https://shap.readthedocs.io/en/latest/

Report Format and Grading [10 Marks]

Your report should be approximately 20 pages, list any additional references you may have used for your answers, and provide the codes you utilized/implemented in your project. The page limit is a general guideline only, and includes any figures and code that you might include in the report. Links to external web pages and code repositories, listing your results and code implementations, such as "GitHub" are permissible.

Kindly provide answers to the above questions in complete sentences and in paragraph form. You will be marked on the correctness, preciseness and comprehensiveness of your answers and quality of your presentation.

Turnitin scores will be visible upon submission, so make sure that you are not penalized for plagiarism in uncited text and code portions.

Front Page Matter:

Page 1. Cover Page. Typed:

- Project title
- Course number
- Student's name (or student names in a group of two students setting)
- Student ID (student IDs in a group of two students setting)
- Name/ID of submitting student (if applicable) – Note: Only one report per groups should be submitted
- Date due
- Date handed in

Submission

The report should be submitted as PDF using the ECE1512 Q-page's assignment facility.