CS 6140: Machine Learning — Spring 2024— Paul Hand

Project Planning Document
Due: Monday March 15, 2024 at 11:59 PM Eastern time via Gradescope.

Names: Darshan Rajopadhye

For your final project, you will obtain a dataset, select multiple machine learning models, train the models, and evaluate the performance of the models. You may elect to reproduce some of the results from a scientific paper, but you must code up some aspect of dataset, model, or training yourself. You may use standard Deep Learning frameworks (e.g. PyTorch, TensorFlow, etc.). You may use code that is available on the internet as building blocks. You must train more than one machine learning model and compare the performance of those models. You are encouraged (but are not required to) to learn about and train models that we have not discussed in class.

THE FINAL PROJECT IS DUE at 11:59 PM on WEDNESDAY April 17, 2024 on Gradescope.

You will write up a short (at most 3 pages) report detailing: the dataset you are using and any data processing you have done, the models you are studying, the details of training the models, and the results of the evaluation. Please use the NeurIPS Style files for your report.

You may work in groups of up to 3 people. You may work alone.

If you want some ideas of projects, here are some ideas. You do not need to select one of these papers.

- Train several handwritten digit classifiers from the table at this website.
- Implement one of the chapters of the Mattmann book.
- Find a Kaggle dataset that you find interesting and train multiple models for it.
- Collect data from your own life and train two predictors that you could use.
- Train a neural network to remove additive noise from images. You can construct a dataset consisting of clean images and noisy images that you construct.
- Create a synthetic dataset and evaluate the k-means and k-means++ algorithms.
- Learn about random forrests and compare them to one or more other methods on real or synthetic data.
- Learn about bootstrapping and compare it to one or more other methods on real or synthetic data.
- Create a synthetic dataset and evaluate how successful cross-validation is at estimating test error.
- Reproduce aspects of Figure 1 of Understanding Deep Learning Requires Rethinking Generalization

Question 1. *Project Planning*

1. Provide a summary of the goal of your project. If you are replicating part of a paper, include a link to the paper here.

   Response:

   As computer networks expand rapidly and applications multiply, the imperative of network security intensifies. Vulnerabilities in systems persist, heightening the risk of detrimental attacks on the economy. Hence, the precise and immediate detection of system vulnerabilities within network packets becomes increasingly vital. This project undertakes a comprehensive investigation into the efficacy of Support Vector Machine models with diverse kernel functions, alongside advanced classification models such as Random Forest and XGBoost for the task of intrusion detection.

   By comparing different SVM kernels (linear, polynomial, RBF, sigmoid) and the advanced models, we aim to discern which approach offers the most effective detection of network intrusions. The project entails meticulous exploration of hyperparameter tuning, model complexity, and data intricacies, aiming to address challenges associated with training and evaluating classification models. Through a rigorous evaluation process encompassing various performance metrics and visualization techniques, we hope to unravel the strengths and limitations of each model, facilitating informed decision-making in real-world deployment scenarios.

2. What dataset will you use?

   Response:

   The NSL-KDD dataset serves as the foundation for the project, offering a comprehensive collection of network traffic data encompassing both normal and anomalous activities. This dataset is widely utilized in the field of intrusion detection for evaluating the performance of intrusion detection systems. It provides a diverse range of features, including protocol types, service types, source and destination IP addresses, duration of connections, and more. With labeled classification of network connections as either normal or anomalous, the NSL-KDD dataset enables supervised learning approaches for intrusion detection.

3. What models will you train? You need to have more than one.

   Response:

   Our project will involve training multiple classification models, including:

   - Support Vector Machine Models (From Scratch Implementations)
     - Linear kernel
     - Polynomial kernel
     - RBF kernel
     - Sigmoid kernel
   - Advanced CLassification Models (SKLearn Models)
     - XGBoost
     - Random Forest
     - Gradient Boosting Machines

4. What do you think will be most difficult about training the models?

   Response:

   - Hyperparameter Tuning :

     One of the primary challenges lies in optimizing the hyperparameters of each model, including the regularization parameter (C), degree (for polynomial kernel), gamma (for RBF kernel), and other kernel-specific parameters. Tuning these hyperparameters effectively requires careful experimentation and may involve extensive computational resources.
   - Data Imbalance and Complexity:

     The NSL-KDD dataset may exhibit class imbalance and contain complex relationships between features, which could impact the performance of the models. Thus thorough EDA must be performed before moving on to models to make sure the data is clean and ready to be modeled.

5. How will you evaluate the models?

Response:

- Performance Metrics :

  We will evaluate the models using a comprehensive set of performance metrics, including accuracy, precision, recall, F1-score, and ROC-AUC score. These metrics provide insights into different aspects of model performance, such as overall classification accuracy, ability to detect intrusions (sensitivity), and control over false alarms (specificity).

- Cross-Validation :

  To ensure reliable model evaluation, we will employ cross-validation techniques such as GridSearch CV and Random Search. This approach helps estimate the models' generalization performance and mitigates the risk of overfitting to the training data.

- Visualizations :

  In addition to numerical metrics, we will utilize visualization techniques such as ROC curves, precision-recall curves, and confusion matrices to gain deeper insights into the models' behavior and performance characteristics. Visualizations provide intuitive representations of the models' trade-offs between true positive and false positive rates, aiding in model selection and interpretation.