CS6316 Project

Code and report due: December 12, 2023, 11:59pm

General Introduction:

There are two separate directions for the project: (1) application project and (2) theoretical project. In the application project, you are given a detailed design and are asked to implement various classification algorithms and analyze the results. In the theoretical project, please select one of the listed topics and write a comprehensive review article.

(1) Application Project

Dataset Description:

Three datasets (*project_dataset1*, *project_dataset2*, and *MNIST*) can be found on Piazza.

Here is a short description of the first two datasets:

Each line represents one data sample.
The last column of each line is class label, either 0 or 1.
The rest columns are feature values, each of them can be a real-value (continuous type) or a string (nominal type).
project_dataset1: 569 observations, 31 attributes
project_dataset2: 462 observations, 10 attributes

Complete the following tasks (on *project_dataset1*, *project_dataset2*) (45%):

- Implement four classification algorithms by yourself: **Logistic Regression**, K **Nearest Neighbor, Decision Tree,** and **SVM**. (Normalize the data to avoid scaling issue, and/or apply regularization to avoid overfitting if needed.)
- Implement **Random Forests** based on your own implementation of Decision Tree.
- Implement **Boosting** based on your own implementation of Decision Tree.
- Adopt 10-fold **Cross Validation** to evaluate the performance of all methods on the provided two datasets in terms of **Accuracy, Precision, Recall**, **F-1 measure, and AUC (area under the curve)**.
- Conduct analysis on Bias-variance tradeoff and overfitting vs. underfitting for all methods, i.e., test different specifications of hyperparameters in these algorithms, e.g., the value you used for K in K Nearest Neighbor, switch from linear kernel to nonlinear kernel in SVM, your parameters for random forests, such as depth and number of trees, with/without regularization in each model, width and depth in neural network with MNIST etc. Basically, observing the training error and testing error with simple and complex model structures and explain the overfitting/underfitting pattern.
- Discuss which algorithm works best in each scenario/validation.

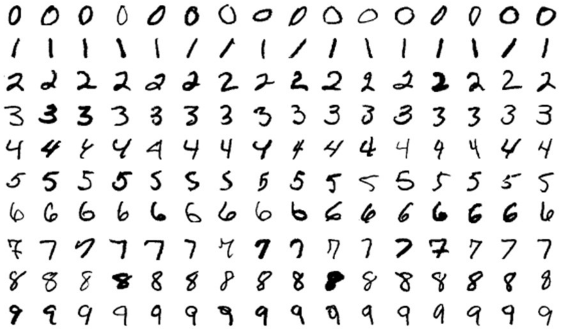
Implement Neural Network code on the MNIST dataset (45%):

Implement neural network (two hidden layers, sigmoid activation function, softmax output layer, and cross entropy loss). You may implement a neural network from scratch or use pytorch (or tensorflow).

The MNIST dataset description:

The MNIST dataset (Modified National Institute of Standards and Technology dataset) is a dataset of handwritten digits which used to serve as a benchmark dataset for various image processing tasks. Here is the visualization of some examples within this dataset. Current dataset consists of 50k training images, 10k validation images, and 10k testing images. Each image has 28 by 28 pixels (equivalently, 784 features).

Your task is to train a neural network with 50k training samples to classify 10 digits (0-9) and report its classification results on 10k testing images (ignore validation images for now).



We have uploaded a piece of code (mnist_loader.py) on Piazza. You could use the following lines of code to import the mnist dataset with mnist_loader.py.

```
import mnist_loader
training_data, validation_data, test_data = mnist_loader.load_data_wrapper()
training_data = list(training_data)
test_data = list(test_data)
```

Try different number of hidden units, different weight/bias initializations, different learning rates, and discuss whether they affect the training/testing performance.

Presentation (10%)

Summarize the key results and possible improvement on any algorithms.

Bonus: 1-15 points bonus on a new improved algorithm with empirical justification and class presentation (Tentative dates: Nov. 29 and Dec 4).

(2) Theoretical Project

Please select one of the following topics and write a comprehensive review article. These ten topics have solid theoretical foundations but have very wide real-world applications.

Your review article should find and analyze related work in your chosen topic that could explain the history and recent advances to a beginner in this topic. You should cover both theoretical foundations and empirical applications of the topic. We provide a list of papers for each topic for you to start with. Your review article should go much beyond the listed papers. You should investigate the strengths and weaknesses of the paper you include in the review and discuss how later papers improve over it. The page limit for the review article is 5-page with the given template. The page limit does not apply to your list of references.

Survey paper (90%), Presentation (10%), and the same bonus as above.

Topic:

(1) Stochastic Gradient Descent and Variants
(2) Generative Adversarial Networks
(3) Bayesian Neural Network
(4) Deep Reinforcement Learning
(5) Extreme Multi-label Classification
(6) Meta-Learning
(7) Interpretability of Deep Learning
(8) Machine Unlearning
(9) Federated learning
(10) Concept-based Learning
(11) Shortcut Learning

Papers to start with:

(1) Stochastic Gradient Descent
[1] Optimization Methods for Large-Scale Machine Learning, Leon Bottou, Frank Curtis, Jorge Nocedal
[2] Adam: A Method for Stochastic Optimization, Diederik P. Kingma, Jimmy Ba
[3] Adaptive Subgradient Methods for Online Learning and Stochastic Optimization, Duchi, J., Hazan, E., Singer, Y.
[4] Quasi-hyperbolic Momentum and Adam for Deep Learning, Jerry Ma, Denis Yarats
[5] On the Convergence of Nesterov's Accelerated Gradient Method in Stochastic Settings, Mahmoud Assran, Michael Rabbat
[6] Don't Decay the Learning Rate, Increase the Batch Size, Samuel Smith, Pieter-Jan Kindermans, Chris Ying, Quoc V. Le

(2) Generative Adversarial Networks
[1] Generative Adversarial Nets, Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio
[2] Unsupervised Representation Learning with Deep Convolution Generative Adversarial Networks, Alec Radford, Luke Metz, Soumith Chintala
[3] Conditional Generative Adversarial Nets, Mehdi Mirza, Simon Osindero
[4] Wasserstein GAN, Martin Arjovsky, Soumith Chintala, Leon Bottou
[5] Generalization and Equilibrium in Generative Adversarial Nets, Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, Yi Zhang

(3) Bayesian Neural Network
[1] Weight Uncertainty in Neural Networks, Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, Daan Wierstra
[2] Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning
[3] Deep Neural Networks as Gaussian Processes, Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel Schoenholz, Jeffrey Pennington, Jascha Sohl-Dickstein
[4] Bayesian Optimization with Robust Bayesian Neural Networks, Jost Tobias Springenberg, Aaron Klein, Stefan Falkner, Frank Hutter
[5] Bayesian GAN, Yunus Saatchi, Andrew Gordon Wilson

(4) Deep Reinforcement Learning

[1] Playing Atari with Deep Reinforcement Learning, Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, Martin Riedmiller
[2] Trust Region Policy Optimization, John Schulman, Sergey Levine, Philipp Moritz, Michael Jordan, Pieter Abbeel
[3] Visualizing and Understanding Atari Agents, Sam Greydanus, Anurag Koul, Jonathan Dodge, Alan Fern
[4] A Theoretical Analysis of Deep Q-Learning, Jianqing Fan, Zhaoran Wang, Yuchen Xie, Zhuoran Yang
[5] Recurrent Models of Visual Attention, Volodymyr Mnih, Nicolas Heess, Alex Graves, Koray Kavukcuoglu
[6] Quantifying Generalization in Reinforcement Learning, Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, John Schulman

(5) Extreme Multi-label Classification

[1] X-BERT: eXtreme Multi-label Text Classification with BERT, Wei-Cheng Chang, Hsiang-Fu Yu, Kai, Zhong, Yiming Yang, Inderjit Dhillon
[2] Deep learning for extreme multi-label text classification, Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, Yiming Yang
[3] Attentionxml: Extreme multi-label text classification with multi-label attention based recurrent neural networks, Ronghui You, Suyang Dai, Zihan Zhang, Hiroshi Mamitsuka, Shanfeng Zhu
[4] Parabel: Partitioned Label Trees for Extreme Classification with Application to Dynamic Search Advertising, Yashoteja Prabhu, Anil Kag, Shrutendra Harsola, Rahul Agrawal, Manik Varma
[5] Consistent Multilabel Classification, Oluwasanmi Koyejo, Nagarajan Natarajan, Pradeep Ravikumar, Inderjit S. Dhillon
[6] MeSHProbeNet: a self-attentive probe net for MeSH indexing, Guangxu Xun, Kishlay Jha, Ye Yuan, Yaqing Wang, Aidong Zhang
[7] Correlation Networks for Extreme Multi-label Text Classification, Guangxu Xun, Kishlay Jha, Jianhui Sun, Aidong Zhang

(6) Meta-Learning

[1] Model-agnostic meta-learning for fast adaptation of deep networks, Chelsea Finn, Pieter Abbeel, and Sergey Levine
[2] Prototypical networks for few-shot learning, Jake Snell, Kevin Swersky, and Richard Zemel
[3] Matching networks for one shot learning, Oriol Vinyals, Charles Blundell, Timothy Lillicrap, and Daan Wierstra
[4] Meta-learning with memory-augmented neural networks, Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap
[5 Meta-learning with latent embedding optimization, Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell
[6] Probabilistic model-agnostic meta-learning, Chelsea Finn, Kelvin Xu, and Sergey Levine
[7] Learning to adapt in dynamic, real-world environments through meta-reinforcement learning, Ignasi Clavera, Anusha Nagabandi, Simin Liu, Ronald S. Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn

(7) Interpretability of Deep Learning

[1] ”Why should I trust you?” Explaining the predictions of any classifier, Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin
[2] Understanding neural networks through representation erasure, Jiwei Li, Will Monroe, and Dan Jurafsky
[3] A unified approach to interpreting model predictions, Scott Lundberg and Su-In Lee
[4] Axiomatic attribution for deep networks, Mukund Sundararajan, Ankur Taly, and Qiqi Yan

[5] Grad-cam: Visual explanations from deep networks via gradient-based localization, Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra
[6] Is attention interpretable? Sofia Serrano and Noah A. Smith
[7] e-SNLI: Natural language inference with natural language explanations, Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom
[8] Towards robust interpretability with self-explaining neural networks, Alvarez Melis, David, and Tommi Jaakkola

(8) Machine Unlearning

[1] Towards Making Systems Forget with Machine Unlearning, Yinzhi Cao and Junfeng Yang
[2] Machine unlearning, Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, Nicolas Papernot
[3] Certified Data Removal from Machine Learning Models, Chuan Guo, Tom Goldstein, Awni Hannun, Laurens Van Der Maaten
[4] Machine Unlearning: Linear Filtration for Logit-based Classifiers, Thomas Baumhauer, Pascal Schöttle, M. Zeppelzauer
[5] When Machine Unlearning Jeopardizes Privacy, Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang
[6] Adaptive Machine Unlearning, Varun Gupta, Christopher Jung, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Chris Waites
[7] Analyzing Information Leakage of Updates to Natural Language Models, Santiago Zanella-Béguelin, Lukas Wutschitz, Shruti Tople, Victor Rühle, Andrew Paverd, Olga Ohrimenko, Boris Köpf, Marc Brockschmidt
[8] Unrolling SGD: Understanding Factors Influencing Machine Unlearning, Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, Nicolas Papernot

(9) Federated Learning

[1] Communication-Efficient Learning of Deep Networks from Decentralized Data, H. Brendan McMahan Eider Moore Daniel Ramage Seth Hampson Blaise Aguera y Arcas
[2] Federated Learning with Non-IID Data, Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, Vikas Chandra
[3] Personalized Federated Learning using Hypernetworks, Aviv Shamsian, Aviv Navon, Ethan Fetaya, Gal Chechik
[4] Adaptive Federated Optimization, Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konecný, Sanjiv Kumar, H. Brendan McMahan
[5] SCAFFOLD: Stochastic Controlled Averaging for Federated Learning, Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, Ananda Theertha Suresh
[6] FedCD: Improving Performance in non-IID Federated Learning, Kavya Kopparapu, Eric Lin, Jessica Zhao
[7] Personalized Federated Learning: A Meta-Learning Approach, Alireza Fallah, Aryan Mokhtari, Asuman Ozdaglar
[8] An Efficient Framework for Clustered Federated Learning, Avishek Ghosh, Jichan Chung, Dong Yin, Kannan Ramchandran

(10) Concept-based Learning

[1] Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV), Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, and Fernanda Viegas

[2] Interpretable Basis Decomposition for Visual Explanation, Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba
[3] Towards Automatic Concept-based Explanations, Amirata Ghorbani, James Wexler, James Y. Zou, and Been Kim.
[4] Concept bottleneck models, Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang
[5] Explaining Neural Networks Semantically and Quantitatively, Runjin Chen, Hao Chen, Jie Ren, Ge Huang, and Quanshi Zhang
[6] Towards robust interpretability with self-explaining neural networks, David Alvarez Melis, and Tommi Jaakkola
[7] Explaining classifiers with causal concept effect (cace), Yash Goyal, Amir Feder, Uri Shalit, and Been Kim
[8] Towards Global Explanations of Convolutional Neural Networks With Concept Attribution, Weibin Wu, Yuxin Su, Xixian Chen, Shenglin Zhao, Irwin King, Michael R. Lyu, and Yu-Wing Tai.

(11) Shortcut Learning

[1] Geirhos, R.; Jacobsen, J. H.; Michaelis, C.; Zemel, R.; Brendel, W.; Bethge, M.; and Wichmann, F. A. 2020. Shortcut learning in deep neural networks. Nature Machine Intelligence, 2(11): 665–673.
[2] Sagawa, S.; Koh, P. W.; Hashimoto, T. B.; and Liang, P. 2019. Distributionally Robust Neural Networks. In International Conference on Learning Representations.
[3] Geirhos, R.; Rubisch, P.; Michaelis, C.; Bethge, M.; Wichmann, F. A.; and Brendel, W. 2019. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In International Conference on Learning Representations.
[4] Izmailov, P.; Kirichenko, P.; Gruver, N.; and Wilson, A. G. 2022. On feature learning in the presence of spurious correlations. Advances in Neural Information Processing Systems, 35: 38516–38532.
[5] Kirichenko, P.; Izmailov, P.; and Wilson, A. G. 2022. Last Layer Re-Training is Sufficient for Robustness to Spurious Correlations. In The Eleventh International Conference on Learning Representations.
[6] Lee, Y.; Yao, H.; and Finn, C. 2022. Diversify and disambiguate: Out-of-distribution robustness via disagreement. In The Eleventh International Conference on Learning Representations.
[7] Liu, E. Z.; Haghgoo, B.; Chen, A. S.; Raghunathan, A.; Koh, P. W.; Sagawa, S.; Liang, P.; and Finn, C. 2021. Just train twice: Improving group robustness without training group information. In International Conference on Machine Learning, 6781–6792. PMLR.
[8] Nam, J.; Cha, H.; Ahn, S.; Lee, J.; and Shin, J. 2020. Learning from failure: De-biasing classifier from biased classifier. Advances in Neural Information Processing Systems, 33: 20673–20684.
[9] Nam, J.; Kim, J.; Lee, J.; and Shin, J. 2022. Spread Spurious Attribute: Improving Worst-group Accuracy with Spurious Attribute Estimation. In International Conference on Learning Representations.
[10] Neuhaus, Y.; Augustin, M.; Boreiko, V.; and Hein, M. 2022. Spurious Features Everywhere–Large-Scale Detection of Harmful Spurious Features in ImageNet. arXiv preprint arXiv:2212.04871.

Project Submission:

● Prepare your submission. Make a zipped folder named "*CompID[-CompID]-Project.zip*", where "CompID[-CompID]" refers to the list of your group members' computing IDs. In the folder, you should include:

1. Report: A pdf file named *Classification_report.pdf*. Describe the flow of all the implemented methods, and briefly describe the choice you make (such as parameter setting, pre-processing, post-processing, how to deal with over-fitting, etc.). Compare their performance, and state their pros and cons based on your findings.

2. Presentation PPT file.

3. Code: A zipped folder named *code.zip*, which contains all codes used in this part (preferably, each algorithm has a separate .py file with informative file name). Inside the folder, please also provide a README file which describes how to run your code.

4. A canvas assignment page has been created for Project. Please submit your zipped folder there. One team only needs to provide one submission on canvas.

Note that copying code/results/report from another group or source is not allowed and may result in an F in the grades of all the team members.

If you select the theoretical project, you only need to submit the review paper in pdf format.