MLSS: MNIST project

Ema Borevković[1], Artyom Karpov[2], and Aashish Khimasia[3]

[1]CNN ensemble architecture and training (code and report), OOD detection (code and report), calibration of models
[2]Adversarial Robustness (code and report), Calibration (code and report), TeX report layout and editing, code review.
[3]Data Augmentation (code and report), Distribution Shift (code), Precision research, Report editing and review
*Authors are sorted alphabetically by last name.

I. Introduction

MNIST is a dataset for handwritten digit recognition that is often used in ML. It's available as part of the standard datasets of nearly every ML package. This project isn't to build an MNIST classifier, but an MNIST classifier that satisfies as many desirable safety properties as possible, such as robustness to adversarial examples, robustness to distributional shifts, use for anomaly detection, high accuracy, low calibration error and error detection.

II. Adversarial attacks

Results for adversarial attacks on two vanilla (no adversarial training) neural nets are as follows. The first network is four layers NN: two convolutional layers and two fully connected layers, with dropout, and 21.8K parameters in total. The second NN is an ensemble of three convolutional NN with 4546.3K parameters. We conveyed 6 attacks with ℓ_0, ℓ_2 and ℓ_∞ with three different epsilons (smaller, medium, larger). For all attacks the ensemble NN appeared to be more robust. The ensemble is expected to be more robust by about 10% (the mean of differences).

	CNN	Ensemble	Ensemble5	EnsembleRELU
L2 BIA, ε=0.3	75%	98%	98%	98%
L2 BIA, ε=1.5	42%	68%	65%	64%
L2 BIA, ε=10	0%	0%	0%	0%
L2AGNA, ε=0.3	81%	99%	99%	99%
L2AGNA, ε=1.5	81%	99%	99%	99%
L2AGNA, ε=10	80%	47%	39%	49%
L2DFA, ε=0.3	75%	98%	98%	98%
L2DFA, ε=1.5	42%	77%	70%	78%
L2DFA, ε=10	0%	0%	0%	0%
L2FGA, ε=0.3	76%	98%	98%	99%
L2FGA, ε=1.5	51%	88%	89%	91%
L2FGA, ε=10	0%	13%	4%	16%
LinfFGA, ε=0.3	5%	22%	10%	24%
LinfFGA, ε=1.5	0%	2%	1%	1%
LinfFGA, ε=10	0%	2%	1%	1%
LinfPGDA, ε=0.3	0%	3%	3%	5%
LinfPGDA, ε=1.5	0%	0%	0%	0%
LinfPGDA, ε=10	0%	0%	0%	0%
SAPNA, ε=0.3	81%	99%	99%	99%
SAPNA, ε=1.5	76%	98%	99%	98%
SAPNA, ε=10	26%	6%	6%	7%

Attacks: L2BIA - L2BasicIterativeAttack, L2AGNA - L2AdditiveGaussianNoiseAttack, L2DFA - L2DeepFoolAttack, L2FGA - L2FastGradientAttack, LinfFGA - LinfFastGradientAttack, LinfPGDA - LinfProjectedGradientDescentAttack, SAPNA - SaltAndPepperNoiseAttack.

Training the ensemble to be robust for LinfPGDA attack as per (Madry et al. 2019) [1]) wasn't successful, it resulted in worse scores. Also, for LinfFGA the training resulted in worse accuracy (-10%) and worse robustness to all attacks including the one that the model was trained against. Training: 1 epoch with 0.1 hyperparameter for adversarial examples loss, then 3 epochs with 0.5, then 3 epochs with 0.75. We didn't find the ensemble with 5 models (Ensemble5 in the table) to be more robust. The same for the ensemble with ReLU. This might be due we didn't train for long (we only did 1-5 epochs) which is a good task for further work. Also, further work might be to implement ViM (virtual-logit matching) (Wang et al. 2022) [2] for anomaly detection, but we didn't find a direct approach to implement it for all models under ensemble so that it won't effect adversarial training. We used the work by Schott et al. 2019 [3] to compare current adversarial robustness approaches.

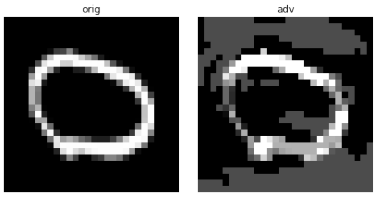
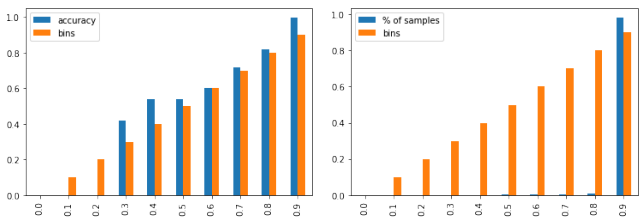


Fig. 1. An adversarial example for FGA attack, ℓ_∞, $\varepsilon = 1.5$.

III. Calibration

Calibration results for the vanilla NNs as described before (see Adversarial attacks section) confirm that the larger model has worse calibration compared to the smaller model. (Guo et al. 2017)[4] also showed that larger networks have worse Expected Calibration Error (ECE). Our small model showed about 13% less ECE than the larger model (200 times more parameters). Still the smaller model with about 15% ECE, 0.17 RMS error and 24% Maximum Calibraion Error can be considered not calibrated. For the ensemble it is 0.28 RMS error, 28% ECE and 49% MCE which is unsatisfactory because it says that model's confidence varies considerably. After applying temperature tuning, the ensemble performed much better. The best ECE is 0.2%, RMS is 0.007, still MCE is 24.5%. See the figure below for the confidence to accuracy bar graph. Still, as the ensemble is +99% accurate almost all predictions are in 90-100% bin.



IV. Out of distribution detection

For OOD detection we experimented with two methods. The first being outlier exposure[5] and the second one being ODIN[6]. For outlier exposure we used *fashion MNIST* as an outlier dataset and *not MNIST* as an OOD dataset. The coefficient for outlier loss was 0.5 as suggested for vision tasks in Hendrycks et al. 2019[5]. First, we tested our model on fashion MNIST validation dataset and the AUROC score went up $\sim 3\%$. We used softmax max anomaly score[7]. However, when testing the model trained with outlier exposure on not MNIST the AUROC score lowered substantially suggesting that the outlier exposure didn't generalize well to the other OOD datasets in our case. For ODIN[6] we experimented with different temperatures and epsilons and got the best results for temp = 0.33 and eps = 0.1. With ODIN, the AUROC score went up $\sim 1\%$ for regular models and even more for the ones trained with OE. In the tabel below you can see the AUROC scores for different methods, models and anomaly scores for not MNIST dataset. We tested ensembles with 3 and 5 models, with and without outlier exposure[5] and four different anomaly scores: softmax max[7], max logit, cross entropy and ODIN[6] with softmax max[7].

	Ens OE	Ens	Ens 5	Ens 5 OE
max logit	0.882	0.977	0.976	0.942
softmax max	0.938	0.977	0.980	0.960
cross entropy	0.910	0.977	0.974	0.955
ODIN	0.969	0.990	0.986	0.985

V. Model architecture and training

Our model architecture was inspired by An et al. 2020[8]. The architecture in this paper is the current SOTA. It consists of three convolutional networks of kernel sizes 3, 5 and 7 respectively connected into an ensemble. An et al. 2020[8] also suggest using ReLU activations. The models we used had the exact same architecture as suggested but we also experimented with some changes. We experimented with GELU activations[9], training with and without outlier exposure and the number of models in the ensemble. The other four ensembles we experimented with had GELU activations[9]. The difference between the four was the number of models in the ensemble (3 and 5) and whether the models were trained with outlier exposure[5]. We decided to use GELU activations because we suspected it could have higher adversarial robustness accuracy. The additional two models were the same as the models with kernel sizes 3 and 5 but with less layers. All of the models trained separately had accuracy $> 98\%$ on the clean dataset, except for the models trained with outlier exposure which lowered the accuracy $\sim 3\%$. All of the ensembles, including the ones trained with outlier exposure had accuracy $> 99\%$. We have decided to add the ensemble with 5 models because we suspected it could yield smaller calibration error. To our surprise neither did the ensemble with five models yield smaller calibration error nor did the GELU activations improve adversarial attack accuracy. In fact, the calibration error of the two ensembles was similar and so was the adversarial attack accuracy of the ensembles with different activation functions. Both the best AUROC anomaly score and the smallest calibration error were attributed to the ensemble with three models with GELU activations and temperature tuning. This ensemble also yielded 99% accuracy on the clean dataset and its adversarial attack accuracy can be seen in the table under section 2.

VI. Data Augmentation

For data augmentation, we generated and experimented with MixUp [10], CutMix [11], random rotation and adding noise. We considered PixMix [12], however did not find this appropriate for the MNIST dataset. We augmented the training set with examples adding 10% to 30% of the data set. Each of the models outlined in section V yielded $> 99\%$ accuracy on noise-augmented data and rotation augmented data, except the model with kernel size 3 and fewer layers which yielded 98%. Both the 3 and 5 network ensembles also yielded $> 99\%$ on this data. Further, we investigated the impact of noise augmented training data upon adversarial robustness. We found that the testing accuracy increased as the proportion of noise augmented data was added to the training set, with proportions added 10% to 100%. However, realising this improved accuracy may be due to the increased number of training examples rather than data augmentation, we performed the same test using non-augmented duplicated and found the same improvement as with augmented data, meaning that this was due to the increased number of training examples. Further time would allow for testing with MixUp and CutMix, as well as the impact of these methods upon robustness to distributional shift using the MNIST-C data set.

References

[1] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2019.

[2] H. Wang, Z. Li, L. Feng, and W. Zhang, "Vim: Out-of-distribution with virtual-logit matching," 2022.

[3] L. Schott, J. Rauber, M. Bethge, and W. Brendel, "Towards the first adversarially robust neural network model on mnist," 2019.

[4] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," 2017.

[5] M. M. Dan Hendrycks and T. Dietterich, "Deep anomaly detection with outlier exposure," 2019.

[6] Y. L. Shiyu Liang and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," 2018.

[7] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," 2016.

[8] S. P. H. Y. Sanghyeon An, Minjun Lee and J. So, "An ensemble of simple convolutional neural network models for mnist digit recognition," 2020.

[9] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelu)," 2016.

[10] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," 2019.

[11] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," 2019.

[12] D. Hendrycks, A. Zou, M. Mazeika, L. Tang, B. Li, D. Song, and J. Steinhardt, "Pixmix: Dreamlike pictures comprehensively improve safety measures," 2022.