IADA: Iterative Adversarial Data Augmentation Using Formal Verification and Expert Guidance

Ruixuan Liu [1] **Changliu Liu** [1]

Abstract

Neural networks (NNs) are widely used for classification tasks for their remarkable performance. However, the robustness and accuracy of NNs heavily depend on the training data. In many applications, massive training data is usually not available. To address the challenge, this paper proposes an iterative adversarial data augmentation (IADA) framework to learn neural network models from insufficient amount of training data. The method uses formal verification to identify the most "confusing" input samples, and leverages human guidance to safely and iteratively augment the training data with these samples. The proposed framework is applied to an artificial 2D dataset, the MNIST dataset, and a human motion dataset. By applying IADA to fully-connected NN classifiers, we show that our training method can improve the robustness and accuracy of the learned model. By comparing to regular supervised training, on the MNIST dataset, the average perturbation bound improved 107.4%. The classification accuracy improved 1.77%, 3.76%, 10.85% on the 2D dataset, the MNIST dataset, and the human motion dataset respectively.

1. INTRODUCTION

The rapid development of neural networks (NNs) makes them widely used in many applications, such as image classification, intention prediction, pattern recognition, etc. Although NNs can approximate arbitrary nonlinear functions, their performance heavily depends on the quality of the training data.

Early works (Anand et al., 1993; Japkowicz & Stephen, 2002; He & Garcia, 2009; Krawczyk, 2016) have shown that a skewly distributed or insufficient amount of training data would make it difficult to train the NN model, as the performance of the learned model will deteriorate in testing. In many applications, due to the high cost of data collection, a well-distributed, sufficient training data is usually not available, which is the key hurdle to deploy learned NNs to real applications. One reason that leads to the high cost is the diversity of the data. For example, in a human motion dataset where the humans are doing assembly tasks, it is expensive (if not impossible) to collect data from all human subjects in all possible task situations. However, human subjects with different habits and task proficiencies may exhibit different motion patterns. Failure to include sufficient data to reflect these differences will lead to poor performance of the learned NN in real situations. Another reason for the high cost in data collection is due to the existence of exogenous input disturbances. These disturbances will not change the ground truth label, but can fool the NN to make a wrong prediction if the NN has not seen sufficiently many disturbed data during training. Such examples are shown in Fig. 2(a) for image classification (Szegedy et al., 2014; Carlini & Wagner, 2017) and in Fig. 2(b) for human intention prediction. These disturbances are usually inevitable when capturing the images or motion profiles due to the sensor noise. However, it is expensive (if not impossible) to generate a full distribution of these disturbances on each data point. To deal with data deficiency, some methods (Liu & Liu, 2021; Cheng et al., 2019) use online adaptation to incrementally update the NN using the data received online, which has been shown to improve prediction accuracy. However, these methods are post-deployment measures and there is no control over the incoming data. These may lead to safety hazards during the adaptation process.

This paper investigates how to efficiently learn NN models before deployment with limited data. The goal is to actively and cost-effectively augment the dataset (i.e., getting full control over the augmented data) so that the learned NN can perform optimally and robustly in real applications. To achieve the goal, this paper proposes an iterative adversarial data augmentation (IADA) framework to train general NN classifiers with limited data. This framework leverages formal verification and expert guidance for iterative data augmentation during training. The key idea of IADA is that

[1] Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA . Correspondence to: Ruixuan Liu <ruixuanl@andrew.cmu.edu>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021.

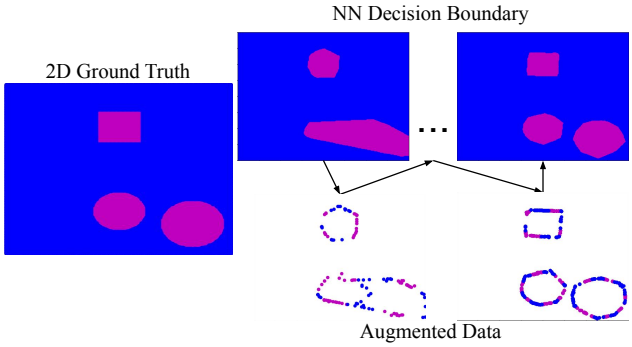


Figure 1: Example of 2D adversarial data augmented by IADA. Left: Ground truth decision boundary. Right first row: NN decision boundary. Right second row: adversarial data augmented by IADA given the current NN model.

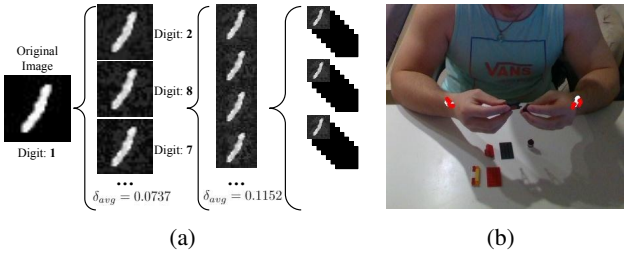


Figure 2: Adversarial data augmented by IADA. (a) MNIST dataset. Left: original training image. Second column: the first level adversaries found around the original image. Third column: the second level adversaries found around the first level adversarial images. Right: further expansions. (b) Human motion dataset. White: original wrist trajectory. Red: adversarial wrist trajectories.

we need to find the most "confusing" samples to the NN, e.g., the samples on the decision boundary of the current NN, and add it back to the dataset. We use formal verification to find these samples by computing the closet adversaries to existing data (called roots) in L_∞ norm. These samples are called "adversaries" since the current NN predicts that they have different labels from the labels of roots (hence they are on the decision boundary of the current NN). The IADA framework will seek expert guidance to label these samples in order to ensure the correctness of adversaries. The sample is a true adversary if its ground truth label is the same as the label of its root; otherwise, this sample is a false adversary. This paper incorporates human-in-the-loop verification as the expert guidance. The labeled samples will be added back to the training dataset no matter what labels they are. The true adversarial samples can improve the robustness of the network, while the false adversarial samples can help recover the ground truth decision boundary. The IADA framework iteratively expands the dataset and trains the NN model. To verify the effectiveness of IADA training, we applied it to three tasks, including a 2D artificial binary classification task, the MNIST digits classification task (Lecun et al., 1998), and a human intention prediction task (Liu & Liu, 2021). We compared our training framework against several other training methods. The results demonstrate that our training method can improve the robustness and accuracy of the learned model.

2. Related Work

Data Augmentation Learning from insufficient training data has been widely studied recently (Anand et al., 1993; Japkowicz & Stephen, 2002; He & Garcia, 2009; Krawczyk, 2016). Data augmentation (DA) is a widely-used approach. People use different approaches to generate additional data given the existing dataset (Shorten & Khoshgoftaar, 2019) to improve the generalizability of the classifiers. (Zhong et al., 2020; Cubuk et al., 2019) propose either to manually design a policy or search for an optimal policy among the pre-defined policies to generate new data. On the other hand, instead of explicitly design the policy, (Perez & Wang, 2017; Lemley et al., 2017; Antoniou et al., 2018; Bowles et al., 2018) propose to learn generative NN models to create new data. However, manually designing the augmentation policy requires a strong domain knowledge. In addition, the designed policy might only be suitable for a small range of related tasks. On the other hand, using NNs for DA is knowledge-free. However, it has poor explainability, which might be a potential concern for safety critical tasks.

Adversarial Training Adversarial training (Bai et al., 2021) has been widely studied since (Szegedy et al., 2014) first introduced the adversarial instability of NNs. Given the existing training data D_0, the adversarial training objective is formulated as a minimax problem

$$\min_{\theta} \mathbf{E}_{(x_i,y_i)\in D_0} \max_{||\delta_i||_\infty \le \varepsilon} L(f_\theta(x_i+\delta_i), y_i), \tag{1}$$

where δ_i is the adversarial perturbation and ε is the maximum allowable perturbation. Early works (Szegedy et al., 2014; Goodfellow et al., 2015) efficiently estimate the adversarial perturbations based on the gradient direction. However, (Moosavi-Dezfooli et al., 2016) showed that the estimation in the gradient direction might be inaccurate, and thus, makes the trained model sub-optimal. Recent work (Cheng et al., 2020) proposes to adaptively adjust the adversarial step size during the learning. Based on the idea of curriculum learning, (Zhang et al., 2020) proposes to replace the traditional inner maximization with a minimization, which finds the adversarial data that minimizes the loss but have different labels. The adversarial data generated via the minimization is also known as the *friendly adversarial data*. On the other hand, neural network verification provides a provably safe way to find the adversarial samples

with minimum perturbations, which are the *most friendly adversaries*. In fact, many works on neural network verification (Bastani et al., 2016) have shown that using the adversarial samples from verification is effective for adversarial training. However, these methods might incorrectly take false adversaries as true adversaries, which might harm the model training. Moreover, they only improve the local robustness of the trained model around the training data, and thus, have limited capacity to improve the generalizability of the network in real situations.

3. Problem Formulation

Given a training dataset $D_0 = \{(x_i, y_i)\}_{i=1}^n$ where x is the input and y is the output, regular supervised training learns the NN model by solving the following optimization

$$\min_\theta \frac{1}{n}\Sigma_{i=1}^n L(f_\theta(x_i), y_i) \left(=: \mathbf{E}_{(x_i,y_i)\in D_0} L(f_\theta(x_i), y_i)\right), \quad (2)$$

where L is the loss function, θ is the NN model parameter, and f_θ is the NN transfer function. Our goal is to learn a model that minimizes the expected loss when deployed in real applications:

$$\min_\theta \mathbf{E}_{(x,y)\in D}(L(f_\theta(x), y)), \quad (3)$$

where D represents the real input-output data distribution, which is unavailable during training. When D_0 has a similar distribution as D, we can obtain a f_θ that behaves similarly as the ground truth f. However, it is usually the case that D_0 is insufficient, and thus, leading to a poorly trained f_θ that behaves differently as f.

This paper tackles the problem of training an NN when D_0 is insufficient and potentially noisy. We aim to augment D_0 with adversarial data D_{adv} to robustly learn the true decision boundaries of the real but unknown data D. The goal is to ensure the learned model on $D_0 \cup D_{adv}$ is as close as possible to (3) with as few amount of augmented data in D_{adv} as possible. To achieve the goal, we formulate the training problem as a two-layer optimization

$$\begin{aligned}
&\min_\theta \ \mathbf{E}_{(x,y)\in D_0\cup D_{adv}}(L(f_\theta(x), y)), \\
&\quad D_{adv} = \bigcup_{x'\in\mathscr{X}'} \left(x', \text{label}(x')\right) \\
&\quad \mathscr{X}' = \{x' \mid \exists (x_0, y_0) \in D_0, \\
&\qquad x' = \arg\min_{||x_0 - x'||_\infty \le \varepsilon \wedge f_\theta(x') \neq y_0} L'(f_\theta(x'), y_0)\}.
\end{aligned} \quad (4)$$

The outer objective optimizes the NN model parameters θ to minimize the loss on the data from $D_0 \cup D_{adv}$. The inner objective constructs D_{adv} given θ, which essentially finds the most friendly adversarial data for all training data. The most friendly adversarial data for $(x_0, y_0) \in D_0$ is an input sample x' that is at most ε distance away from x_0 in the L_∞ norm but changes the network output from y_0 with smallest loss L'. The loss L' for the inner objective may or may not be the same as the original loss L. Ideally, we should choose a loss that guides us to the most "confusing" part of the input space. The label of these friendly adversarial data is decided by an additional labeling function, which ideally should match the ground truth. If $\text{label}(x') = y_0$, we call x' a true adversary; otherwise a false adversary. We will solve (4) in an iterative approach, to be discussed in the following section.

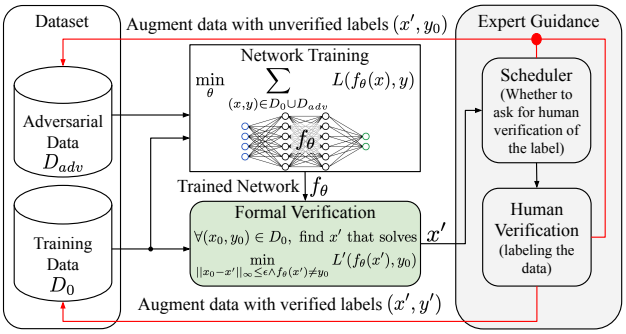


Figure 3: Iterative Adversarial Data Augmentation (IADA) training framework. For $(x,y) \in D_0 \cup D_{adv}$, the framework updates the model. If $(x_0, y_0) \in D_0$, the formal verification finds the most friendly adversarial sample x' around x_0 if exists. The scheduler determines if human verification is required for x'. If not, (x', y_0) is augmented to D_{adv}. Otherwise, if human is able to assign y', then (x', y') is added to D_0. If not, then (x', y_0) is appended to D_{adv}.

4. IADA: Iterative Adversarial Data Augmentation

This paper proposes an iterative adversarial data augmentation (IADA) training framework to robustly learn from insufficient training data. Figure 3 illustrates the proposed training framework that aims to solve the problem defined in (4). In particular, the inner objective will be solved using formal verification to be discussed in section 4.1, while the labeling will be performed under expert guidance to be discussed in section 4.2. The iterative approach to solve the two-layer optimization will be discussed in section 4.3. This paper focuses on fully-connected neural network (FCNN) classifiers with ReLU activations, mainly due to the limitation of the verification algorithms we use. But the framework can be easily extended to other network structures and activation functions by switching to more general verification algorithms.

4.1. Formal Verification to Find Adversaries

The two-layer optimization in (4) is similar to the formulation of adversarial training (1). The major distinction lies in

the inner objective. The minimax formulation in (1) might reduce the model accuracy and generalizability when maximizing the robustness (Tsipras et al., 2019), since the inner maximization could be too aggressive when generating adversaries. Similar to the approach in (Zhang et al., 2020), instead of finding the adversaries via the inner maximization, we take the adversarial label $f_\theta(x_0+\delta_0)$ into consideration when generating the adversarial sample $x' = x_0+\delta_0$. In particular, we design the inner loss to penalize the magnitude of δ_0. Then the inner optimization in (4) can be written as

$$\min_{x'} \|x_0 - x'\|_\infty, \text{ s.t. } \|x_0 - x'\|_\infty \le \varepsilon, f_\theta(x_0) \neq f_\theta(x'). \quad (5)$$

The reason why we use the distance metric in the input space as the loss L' instead of using the original loss or any other loss that penalizes the output is that this metric reflects the "confusing" level of samples. We generally expect that the learned model is regular at the training data for generalizability. The easier it is to change the label by perturbing the input data, the less regular the model is, and hence more "confusing". In addition, this formulation provides a quantitative metric δ_0 for evaluating the robustness online. Therefore, we can prioritize enhancing the weaker boundaries and obtain full control of the training process. Nevertheless, the optimization in (5) is nontrivial to solve due to the nonlinear and nonconvex constraints introduced by f_θ. To obtain a feasible and optimal solution, we use formal verification (Liu et al., 2020) to find the appropriate x'. The optimization in (5) essentially finds the minimum input adversarial bound, which can be solved by various neural verification algorithms. In particular, primal optimization-based methods such as MIPVerify (Tjeng et al., 2019) and NSVerify (Lomuscio & Maganti, 2017) solve the problem exactly by encoding the neural network into a mixed integer linear program; dual optimization-based methods (Wong & Kolter, 2018) can compute an upper bound of the problem by relaxing the nonlinear ReLU activation functions; reachability-based methods (Weng et al., 2018) can also compute an upper bound by over-approximating the reachable set and binary search for the optimal loss.

4.2. Expert Guidance for Labeling

With the adversaries obtained from formal verification, we need to label them before augmentation. The proposed IADA framework uses expert guidance to guarantee the safety and accuracy of the training. The framework requires human expert to verify the newly added adversaries. It uses a scheduler to balance the required human effort and the training accuracy. We use an ensemble NN and the L_∞ distance check to construct the scheduler. The ensemble NN indicates whether the adversary is meaningful. We say a data is meaningful if humans can properly interpret the data label. For example, the disturbance to human trajectory that violates human kinematic cosntraints or to MNIST image that have two digits shown in one image are not meaningful. The scheduler requires human verification if $\delta_0 > d$ or the ensemble model agrees that the adversary is meaningful, where d is a pre-defined threshold. Based on the application, a smaller d can be chosen to require more frequent human verification to improve the training accuracy, while a larger d alleviates the amount of human effort.

4.3. Iterative Adversarial Data Augmentation

We use an iterative approach to solve the two-layer optimization in (4) by incrementally augmenting the data with the outer NN training loop. In one iteration, we find the adversaries generated in section 4.1, which are the most "confusing" points for the current NN. The adversaries will be labeled by expert and augmented to the dataset, either D_0 or D_{adv}, as shown in Fig. 3. In the next iteration, the framework will further verify, label, and expand the dataset based on the augmented dataset. The framework maintains two datasets D_0 and D_{adv}, where D_{adv} is initially empty and D_0 starts with the original training data. The adversaries are only augmented to D_0 if they are verified by human expert, otherwise, they are pushed into D_{adv}. Note that the adversaries generated in section 4.1 can either be true or false adversaries, but both are informative and useful for improving the NN learning. However, incorrectly mixing these two types of adversaries can greatly harm the NN learning. Therefore, we maintain D_0 and D_{adv} to distinguish the safe data and potentially incorrect data. The framework iteratively expands only from the data $(x_0, y_0) \in D_0$. Therefore, the framework can further expand the available training data safely. D_{adv} will be refreshed before verification as shown in algorithm 1, since the framework only wants temporary effect from D_{adv} but permanent effect from D_0 for safety and correctness. Also note that the IADA framework is reduced to standard adversarial training, except that we have a different inner minimization loss objective, if the scheduler requires no human verification. On the other hand, it is equivalent to standard online data augmentation when the scheduler always requires human verification (similar to data aggregation in imitation learning (Ross et al., 2011)). Based on the applications, the scheduler can be tuned to either require more or less expert knowledge to balance the training accuracy and the human effort.

The IADA framework is summarized in algorithm 1. The verification rate r_v indicates the frequency of online formal verification and data augmentation. The verification number C indicates a maximum number of points to verify at each iteration. Q_v is a priority queue based on the level of expansion and the perturbation bound δ. The level is defined as 0 for the original training data. The adversaries generated from original data have the level 1. Q_v determines the weakest points and prioritizes those points during verification and expansion. On line 12, the system gets the weakest point

Algorithm 1 Iterative Adversarial Data Augmentation (IADA)

1: **Input:** Original dataset $(x_0, y_0) \in D_0$.
2: **Input:** Robustness bound ε, verify rate r_v, verification number C, learning rate α.
3: **Output:** NN parameter θ.
4: **Initialize:** $\theta = \theta_0$, $Q_v = \{D_0\}$.
5: **repeat**
6: **if** VerificationRound(r_v) **then**
7: // Refresh D_{adv} since it might contains data with incorrect label.
8: Clear D_{adv}.
9: **for** $i = 1, \ldots, C$ **do**
10: Breaks if Q_v is empty.
11: // Q_v prioritizes the point with 1) the lowest level of expansion and 2) the smallest perturbation bound.
12: $(x,\ y) = Q_v.pop$.
13: // Find an adversary by solving (5).
14: $x' = Verify(\theta, x, y, \varepsilon)$.
15: Skip if no adversary x' is found.
16: **if** Scheduler && Human Verified **then**
17: Obtain verified label y'.
18: Push (x,y) and (x',y') to Q_v and append (x',y') to D_0.
19: **else**
20: Append (x',y) to D_{adv}.
21: **end if**
22: **end for**
23: **end if**
24: **for** minibatch $\{X,Y\}$ in $D_0 \cup D_{adv}$ **do**
25: $\theta \longleftarrow \theta - \alpha L(f_\theta(X), Y)$.
26: **end for**
27: **until** *max_epoch* reached.
28: **return** θ.

and verifies it on line 14. If the robustness is not satisfied, the system either asks for expert knowledge to assign a label or assumes it to have the same label as the root. If the label is assigned by humans, the adversarial sample is added to D_0 for further expansion. The NN model is updated using the training data and the adversarial data at line 24-26.

4.4. IADA Analysis

The IADA framework is unique in several ways. First, it is well-known that there exists a trade-off between accuracy and robustness in general adversarial training (Tsipras et al., 2019). This is mainly due to incorrectly mixing the true and false adversaries. However, by introducing expert guidance, the true adversaries can improve the NN robustness and enlarges the decision area, while the false adversaries can enhance the ground truth decision boundary for better accuracy. Second, although the augmented data may not recover the ground truth data distribution, the augmented data will converge to and fully cover the ground truth decision boundary in the limit (proof left for future work). As a result, the learned NN will converge to the ground truth with the correct decision boundary. Hence this data augmentation is most effective.

Based on these features, we argue that IADA is most suitable for tasks that have non-trivial distribution of the data around the decision boundary, such as human intention prediction in section 5.3. Such tasks generally have false adversaries close to the training data and can easily have "confused" decision boundaries. On the other hand, image-related tasks, such as MNIST, are suitable for using IADA, but not neccessarily required. It is mainly due to that image data is generally easier to collect, thus, unlikely to have insufficient data. In addition, it is rare to have false adversaries close to the original images.

5. Experiment Results

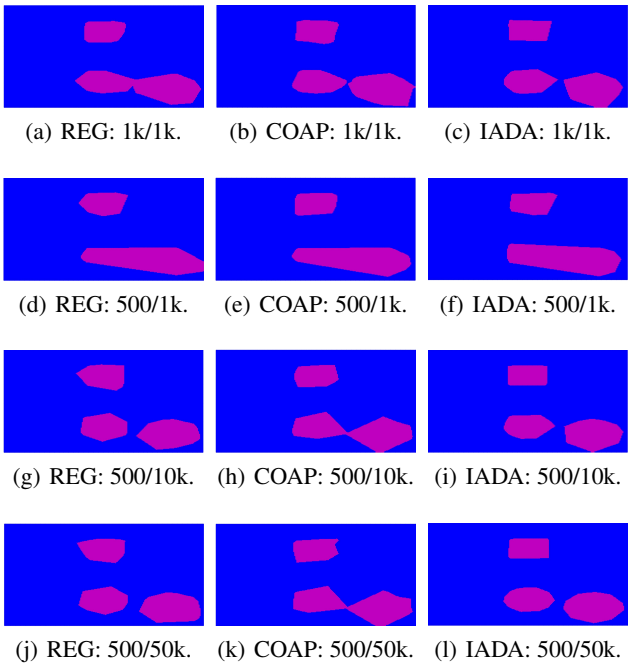
We evaluate the proposed IADA framework by training NN classifiers in three different problems, including a 2D artificial binary classification problem shown in Fig. 1, the MNIST digits classification problem (Lecun et al., 1998), and the intention prediction problem using the human motion dataset in (Liu & Liu, 2021). We use a single-layer fully-connected neural network (FCNN) with the hidden neuron size being 32 and the ReLU activation function. The formal verification is implemented using the MIPVerify in NeuralVerification.jl toolbox (Liu et al., 2019). We compared three training methods on each problem, including the regular supervised training (REG), the robust training via the convex outer adversarial polytope (COAP) (Wong & Kolter, 2018), and our IADA training. The regular and IADA training use the Cross-Entropy loss with the Adam optimizer (Kingma & Ba, 2017) implemented in PyTorch (Paszke et al., 2019). The learning rate is set to 0.01 for all learning methods. The ε values are set to 0.1, 0.1, and 0.05 for the 2D, MNIST, and intention problem respectively. For IADA, the verify rate is set to be $r_v = 500$, the verification number C is set to be 5000, 2000, and 5000 for the 2D, MNIST, and intention problem respectively. All experiments were run in Windows 10 with AMD Ryzen 3700X 8-Core processor, 16GB RAM and an RTX 2070 Super GPU.

5.1. 2D Binary Classification

We include the 2D example mainly to provide a better visualization and understanding of the training process. Figure 1 shows the ground truth of the problem, where we have two classes. Given an available training dataset, we want the FCNN to recover the true decision boundary. The scheduler is implemented only using the distance check since the 2D data point does not have semantic meaning. During the human verification, ideally the human assigns the ground truth label. But in this experiment, we assume the human has the ground truth knowledge and directly use the ground truth to automatically assign the labels. We evaluate the training

Table 1: Comparison of the FCNN classification accuracy on 2D binary classification. The numbers in () indicates the number of augmented adversaries by IADA to D_0. REG+DA and COAP+DA are the training methods with uniform data augmentation with the same amount of data augmented by IADA.

	REG	REG + DA	COAP	COAP + DA	IADA
Data Size: 1000, Epochs: 1000	97.16%	95.62%	97.22%	98.01%	**97.62%** ($\sim$ 200)
Data Size: 500, Epochs: 1000	94.97%	94.97%	**95.89%**	95.69%	95.53% ($\sim$ 200)
Data Size: 500, Epochs: 10000	97.93%	98.46%	97.61%	98.55%	**99.01%** ($\sim$ 1000)
Data Size: 500, Epochs: 50000	97.78%	98.99%	97.31%	99.39%	**99.51%** ($\sim$ 5000)



(a) REG: 1k/1k. (b) COAP: 1k/1k. (c) IADA: 1k/1k.

(d) REG: 500/1k. (e) COAP: 500/1k. (f) IADA: 500/1k.

(g) REG: 500/10k. (h) COAP: 500/10k. (i) IADA: 500/10k.

(j) REG: 500/50k. (k) COAP: 500/50k. (l) IADA: 500/50k.

Figure 4: Visualizations of the decision boundaries of the trained FCNN. The numbers are in the format **data size/epochs**.

quality by visualizing the decision boundaries of the classifier (shown in Fig. 4) and testing the classification accuracy (shown in table 1). Note that the result in this example for COAP has $\varepsilon = 0.01$. Since the adversarial bound computed in COAP is over-approximated, enlarging the robustness to $\varepsilon = 0.1$ will result in too many false adversaries incorrectly taken as true adversaries, which will fail the training.

We started from 1000 training samples. From table 1, we can see all trained classifiers have high classification accuracy, which is above 97%. IADA achieves the highest accuracy among all. We then decreased the training data size to 500 to simulate the situation with insufficient training data. As shown in table 1, with limited training epochs, all trained models have worse accuracy. When we increase the training epochs, we observe that the trained models have higher accuracy, where IADA has the highest among all. However, as shown in Fig. 4(g) and Fig. 4(h), the recovered regions skew and overfit the training data, whereas IADA recovers similar regions as the ground truth as shown in Fig. 4(i). As we further increase the number of training epochs to 50000, we can see there is more overfitting by regular training and COAP as the accuracy starts to drop. On the other hand, IADA further refines the decision boundary and improves the classification accuracy.

Figure 1 shows the data augmentation process of IADA. The augmented data converges to the true decision boundary as mentioned in section 4.4. The numbers in the brackets of the last column in table 1 indicates the number of adversaries being augmented. To demonstrate the effectiveness of using adversaries for data augmentation, we added the same amount of data, by uniform sampling, to the original training dataset for REG and COAP. The model accuracy is shown in the second and fourth columns of table 1. We can see that although the performance improves for both methods, IADA still achieves the highest accuracy. This demonstrates that the adversaries are more informative than uniformly sampled data. IADA is more effective and requires fewer additional data for training.

5.2. MNIST

MNIST is a more realistic classification problem in higher dimension. It is rare to have adversarial images that humans cannot interpret the ground truth label. Thus, the scheduler is implemented only using the distance check. During the human verification, the human is given the adversarial image and asked to assign the ground truth label. Table 2 shows the performance of different training methods on the MNIST dataset. Each entry includes the classification accuracy using the MNIST testing dataset and the average perturbation bound of the trained model. Given the testing data $D_T = \{x_t, y_t\}_{t=1}^n$, the average perturbation bound is calculated as $p_b = \frac{1}{n}\Sigma_{t=1}^n \delta_t$, where $\delta_t = 0$ if $f_\theta(x_t) \neq y_t$. If $f_\theta(x_t) = y_t$, δ_t is the solution to (5) on the point x_t. Figure 2(a) shows a visualization of the adversarial images found by IADA and the iterative expansion. As we expand for more iterations, the adversarial images have more perturbations added relative to the original image. From table 2, we can see that IADA, in general, achieves the highest accuracy, which indicates that the model learned a better decision boundary. However, COAP achieves larger average perturbation bound in general. This is mainly due to the different

Table 2: Comparison of the FCNN accuracy on MNIST. Each entry is in the format of **Accuracy** (p_b).

	REG	COAP	IADA
Data Size: 3000, Epochs: 1000	87.60% (0.00719)	88.30% (0.01393)	**89.60% (0.01491)**
Data Size: 2000, Epochs: 1000	84.85% (0.00689)	85.30% **(0.01346)**	**87.05%** (0.01309)
Data Size: 1000, Epochs: 1000	82.45% (0.00821)	83.95% **(0.01495)**	**85.55%** (0.01361)
Data Size: 500, Epochs: 1000	80.80% (0.00975)	80.90% **(0.01599)**	**82.55%** (0.01301)
Data Size: 500, Epochs: 2000	80.80% (0.01006)	80.75% **(0.01521)**	**82.85%** (0.01094)
Data Size: 500, Epochs: 3000	80.70% (0.00972)	80.65% (0.01492)	**83.10% (0.01641)**

Table 3: Comparison of the FCNN accuracy on human intention prediction.

Epochs	REG	COAP	IADA
500	81.99%	77.94%	**82.53%**
1000	**85.92%**	77.95%	85.48%
2000	85.26%	81.66%	**88.75%**
3000	82.97%	83.95%	**89.52%**
4000	82.86%	83.95%	**90.83%**
5000	81.55%	83.95%	**92.03%**

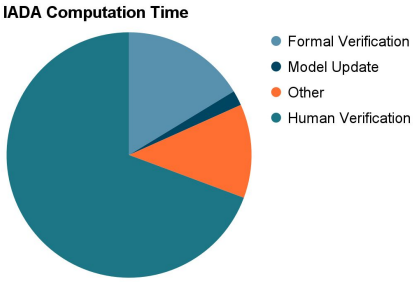


Figure 5: Time cost decomposition for IADA training. Model update refers to training using $D_0 \cup D_{adv}$. Other refers to data operation (*i.e.*, Q_v operation).

objectives for the two training methods. The objective for COAP is to enlarge the perturbation bound, whereas IADA focuses on recovering the true decision boundary.

5.3. Intention Prediction

Intention prediction is widely studied in human-robot collaboration (HRC). The model outputs an intention label given the observed trajectory. We use parts of the dataset from (Liu & Liu, 2021). The FCNN is trained using the first two trials of human subject 1 doing task 1 and tested using the third trial. The input to the FCNN is the previous 10-step (0.3s) historical trajectories for both right and left wrists and the output is a class label: assembling, retrieving, reaching, and abnormal. Figure 2(b) shows the adversarial wrist trajectories being added to the dataset by IADA. The model validation accuracy is shown in table 3. We can see that the accuracy for regular and COAP increase initially. But soon the accuracy starts to drop in regular training due to the overfitting while COAP stays constant. On the other hand, by using expert knowledge and iterative expansion, IADA is able to continuously improve the model accuracy.

5.4. Discussion

Time Efficiency Figure 6 shows the training time for each method in each problem. We can see that COAP and REG have very comparable time costs. IADA is significantly more expansive in terms of computation time. On the 2D problem, it is around 5 to 7× the time of REG and COAP, but around 20× and 100× on larger problems, *i.e.*, MNIST and intention prediction. We can expect the time cost to be higher when scaling to more complex and larger NN structures. Figure 5 shows the time decomposition of the IADA training on the 2D example. We can see the most expensive part is the human verification, since the problem requires more expert guidance. Due to low dimensionality of the problem, the formal verification takes smaller ratio in the total computation time. Similarly for MNIST and intention prediction, human verification and formal verification takes more than 85% of the total training time, but formal verification has significantly larger ratio. This is mainly due to the problem requires less human verification for MNIST, and the formal verification is much more computationally expensive on higher dimensional problems. However, note that the comparison established here is slightly unfair since both REG and COAP are implemented solely in Python, whereas IADA has the formal verification implemented in Julia. The formal verification time accounts for the time exchanging between platforms.

Human Effort It is interesting to observe that COAP fails on the 2D problem when $\varepsilon = 0.1$. But it works properly when $\varepsilon = 0.01$. For the 2D task and the intention prediction task, the data is more likely to be on the decision boundary, thus, more likely to find false adversaries online. Therefore, expert guidance is critical to make the training safe and robust. Figure 7 shows the percentage of the true adversaries generated online during training. It is obvious that MNIST has significantly more true adversaries generated online. Therefore, the required human effort is significantly lower for image-related tasks, *i.e.*, MNIST, compared to other tasks, *i.e.*, intention prediction. By introducing expert guidance, IADA is most suitable for tasks with "confusing"

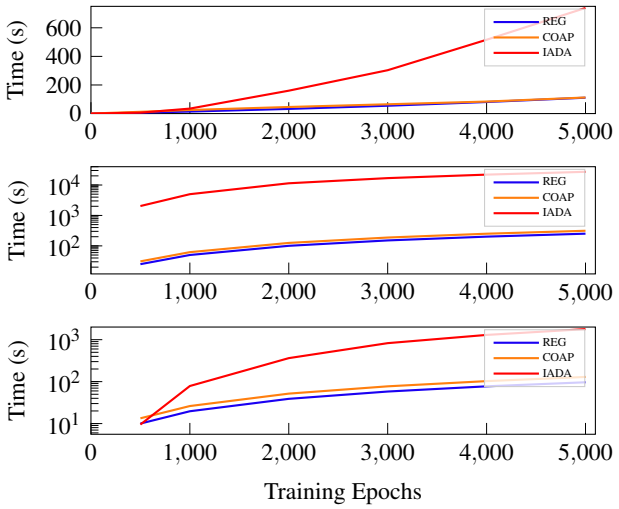


Figure 6: Training Time Comparisons. Top: 2D binary classification. Middle: MNIST. Bottom: Intention Prediction.

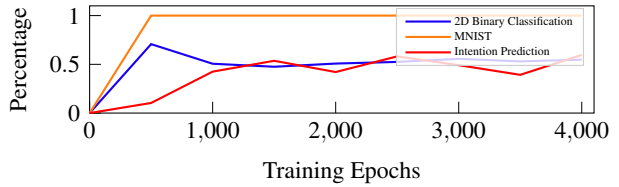


Figure 7: Percentage of true adversarial data found by formal verification online.

decision boundaries in which case it is easy to mix true and false adversaries, as discussed in section 4.4.

6. Conclusion and Future Work

This paper proposes an iterative adversarial data augmentation (IADA) framework to train general NN classifiers. It uses formal verification online to find most vulnerable part of the network such as samples on the decision boundary of the network (adversaries). By acquiring human expert knowledge, the framework augments the training data using the adversaries verified by humans and iteratively expands the available data. The experiments demonstrate that our training method can improve the robustness and accuracy of the learned model. The IADA framework has several advantages. First, it is composable since it allows easy switch of individual modules (*i.e.*, the training algorithm, the formal verification method, and the method for expert guidance). Second, it is safe and explainable due to the inclusion of expert guidance, unlike other adversarial training methods. Lastly, it is generic and appliable to general NNs, unlike existing data augmentation methods that require strong domain knowledge.

There are many future directions that we would like to pursue, including formally proving that the IADA framework will ensure convergence of the learned NN model to the ground truth; exploring efficient online verification algorithms (both implementation-wise and algorithm-wise) to improve the scalability of the IADA framework; designing a better scheduler in order to balance the required human effort and the learning accuracy; extending the IADA framework to general regression NN where we will augment new data at places with larger gradients.

References

Anand, R., Mehrotra, K., Mohan, C., and Ranka, S. An improved algorithm for neural network classification of imbalanced training sets. *IEEE Transactions on Neural Networks*, 4(6):962–969, 1993.

Antoniou, A., Storkey, A., and Edwards, H. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2018.

Bai, T., Luo, J., Zhao, J., Wen, B., and Wang, Q. Recent advances in adversarial training for adversarial robustness. *arXiv preprint arXiv:2102.01356*, 2021.

Bastani, O., Ioannou, Y., Lampropoulos, L., Vytiniotis, D., Nori, A., and Criminisi, A. Measuring neural net robustness with constraints. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Bowles, C., Chen, L., Guerrero, R., Bentley, P., Gunn, R., Hammers, A., Dickie, D. A., Hernández, M. V., Wardlaw, J., and Rueckert, D. Gan augmentation: Augmenting training data using generative adversarial networks. *arXiv preprint arXiv:1810.10863*, 2018.

Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. *arXiv preprint arXiv:1608.04644*, 2017.

Cheng, M., Lei, Q., Chen, P.-Y., Dhillon, I., and Hsieh, C.-J. Cat: Customized adversarial training for improved robustness. *arXiv preprint arXiv:2002.06789*, 2020.

Cheng, Y., Zhao, W., Liu, C., and Tomizuka, M. Human motion prediction using semi-adaptable neural networks. In *2019 American Control Conference (ACC)*, pp. 4884–4890, 2019.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2019.

Goodfellow, I., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

He, H. and Garcia, E. A. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

Japkowicz, N. and Stephen, S. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, pp. 429–449, 2002.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2017.

Krawczyk, B. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lemley, J., Bazrafkan, S., and Corcoran, P. Smart augmentation learning an optimal data augmentation strategy. *IEEE Access*, 5:5858–5869, 2017.

Liu, C., Arnon, T., Lazarus, C., and Kochenderfer, M. J. NeuralVerification.jl: Algorithms for verifying deep neural networks. In *Workshop on Debugging Machine Learning,*, 2019.

Liu, C., Arnon, T., Lazarus, C., Strong, C., Barrett, C., and Kochenderfer, M. J. Algorithms for verifying deep neural networks. *arXiv preprint arXiv:1903.06758*, 2020.

Liu, R. and Liu, C. Human motion prediction using adaptable recurrent neural networks and inverse kinematics. *IEEE Control Systems Letters*, 5(5):1651–1656, 2021.

Lomuscio, A. and Maganti, L. An approach to reachability analysis for feed-forward relu neural networks. *arXiv preprint arXiv:1706.07351*, 2017.

Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. Deepfool: a simple and accurate method to fool deep neural networks. *arXiv preprint arXiv:1511.04599*, 2016.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. 2019.

Perez, L. and Wang, J. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

Ross, S., Gordon, G., and Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635. PMLR, 11–13 Apr 2011.

Shorten, C. and Khoshgoftaar, T. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6: 1–48, 2019.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

Tjeng, V., Xiao, K., and Tedrake, R. Evaluating robustness of neural networks with mixed integer programming. *arXiv preprint arXiv:1711.07356*, 2019.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2019.

Weng, T.-W., Zhang, H., Chen, H., Song, Z., Hsieh, C.-J., Boning, D., Dhillon, I. S., and Daniel, L. Towards fast computation of certified robustness for relu networks. *arXiv preprint arXiv:1804.09699*, 2018.

Wong, E. and Kolter, Z. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5286–5295. PMLR, 10–15 Jul 2018.

Zhang, J., Xu, X., Han, B., Niu, G., Cui, L., Sugiyama, M., and Kankanhalli, M. Attacks which do not kill training make adversarial learning stronger. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 11278–11287. PMLR, 13–18 Jul 2020.

Zhong, Z., Zheng, L., Kang, G., Li, S., and Yang, Y. Random erasing data augmentation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):13001–13008, Apr. 2020.