3-10-2022

Robustness Analysis of Classification Using Recurrent Neural Networks with Perturbed Sequential Input

Guangyi Liu
Leigh University, PA, United States

Arash Amini
Leigh University, PA, United States

Martin Takac
Mohamed bin Zayed University of Artificial Intelligence

Nader Motee
Leigh University, PA, United States

Robustness Analysis of Classification Using Recurrent Neural Networks with Perturbed Sequential Input

Guangyi Liu, Arash Amini, Martin Takáč, and Nader Motee

***Abstract*— For a given stable recurrent neural network (RNN) that is trained to perform a classification task using sequential inputs, we quantify explicit robustness bounds as a function of trainable weight matrices. The sequential inputs can be perturbed in various ways, e.g., streaming images can be deformed due to robot motion or imperfect camera lens. Using the notion of the Voronoi diagram and Lipschitz properties of stable RNNs, we provide a thorough analysis and characterize the maximum allowable perturbations while guaranteeing the full accuracy of the classification task. We illustrate and validate our theoretical results using a map dataset with clouds as well as the MNIST dataset.**

I. Introduction

Real-time perception and classification have been among the most exciting topics in computer vision-based and machine learning-based robotic applications. However, in the most perception-based applications, the uncertainties and perturbations prevail in every section of a learned model from the input perturbation [6, 24] to the numerical error [28]. In order to ensure reliable performance for these applications, some inevitable questions need to be answered: (i) If input perturbation exists, does the learned framework still exhibit robust performance? (ii) What conditions does the learned model needs to satisfy to achieve robustness? (iii) Does there exist characteristics that can measure the robustness of the learned model? Finding the answer to these questions will significantly facilitate tackling perception-based problems since most learned models suffer from the fragility of the input perturbations [30].

In many applications involving area coverage, consensus, map classification, rendezvous [2], the robot can only sample the localized information of the environment [29], i.e., partially observation, at each time step. In order to acquire adequate observations, the robot could consider traversing the environment while collecting local information as sequential data and attempting to learn their inter-correlation with recurrent neural networks. The same type of framework that uses localized observations and recurrent neural networks to learn the image and map classification is illustrated in our previous works [22, 21, 16, 17], and shows promising performance in various scenarios.

In this paper, we consider the map (image) classification problem with the sampled sequential images as a motivational example, which is illustrated in Fig. 1. Instead of treating the learned classification model as a black box and solely focusing on its performance, we consider how it will perform with perturbed input and analyze its robustness in terms of its learned weights from the neural networks. The robustness analysis is achieved by considering the stability properties of the recurrent models that learn the interconnection of past observations and quantifying the classification criterion via the Voronoi partitioning to obtain the robustness conditions.

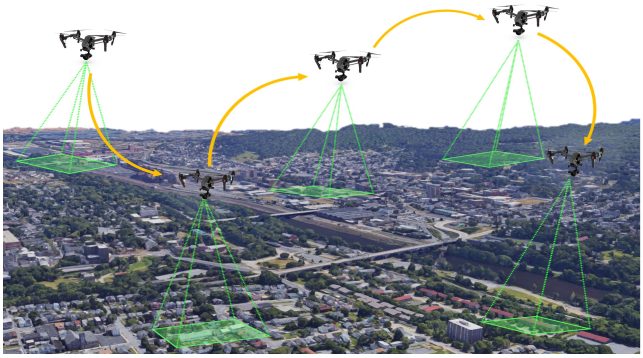

Fig. 1: The above figure depicts the motivational application of map classification with the aerial robot. The robot traverses the environment and collects the visual inputs as a sequence (order denoted by arrows).

The robustness of a classification model can be considered as not miss-classifying under the adversary attacks or perturbations [18], and recent research has made progress on investigating and ensuring the robustness of neural network models under the adversarial attack [31, 5]. Our work exhibits the novelty and differences to these works as: Instead of considering the Boolean classifier, we propose a robustness analysis of multi-class classifier [19], in which a novel representation with Voronoi diagram [4] is used to construct a quantifiable classification criterion. To ensure the robustness of a learned model, we implement the similar ideas of using expert demonstrations [25], for which we require the perturbed result to stay close to the nominal results, see §VI. To evaluate the deviation caused by the perturbation at the output stage, we seek the boundedness for every section of the classification model. For instance, inspired by the recent work on constructing a bound for neural network models, to name a few: [20, 12, 31], we seek similar boundedness for RNNs, which will provide an error estimate when the statistics of the input perturbation is available [1].

G.L., A.A., and N.M. are with the Department of Mechanical Engineering and Mechanics, Lehigh University, Bethlehem, PA 18015, USA {gliu,a.amini,motee}@lehigh.edu.

M.T. is with the Machine Learning Department at Mohamed bin Zayed University of Artificial Intelligence (MBZUAI), Masdar City, Abu Dhabi, United Arab Emirates {takac.mt}@gmail.com.

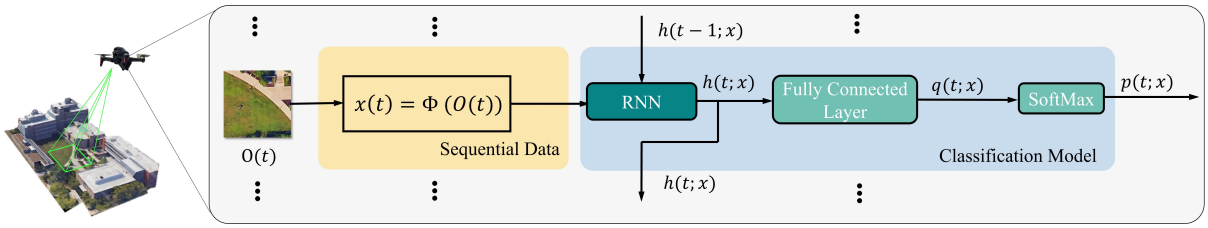

Fig. 2: This figure shows the image (map) classification model at time t.

Our Contributions: In this paper, we construct a formal approach to analyze the robustness of the recurrent multi-label classification framework with localized sequential inputs when there exist input perturbations. Furthermore, we also quantify the multi-label classification criterion that uses the $\arg\max$ function. Our analysis shows that a robust classification with full accuracy is guaranteed when the RNNs are stable and the input perturbation is below the maximum allowable deviation. These results motivate us to turn our research efforts to explore further how the classification robustness can show its effect in a closed-loop model where the robot can select its sampling routine based on its observations.

The rest of the paper is organized as follows. In §III, we introduce the problem setting and the classification model. The possible origins of perturbations are illustrated in §IV. The error estimates and the boundedness of stable RNNs are presented in §V. Our main result is presented in §VI, where the quantifiable classification criterion and robust classification conditions are presented. The theoretical findings are validated in §VII by simulations in both MNIST dataset [14] and Campus Map dataset [17].

II. Mathematical Notations

The $n-$dimensional Euclidean space with elements $\boldsymbol{z} = [z_1, \ldots, z_n]^T$ is denoted by $\mathbb{R}^n$, where $\mathbb{R}_+$ will denote the positive orthant of $\mathbb{R}^n$. The set of standard Euclidean basis for $\mathbb{R}^n$ is represented by $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n\}$. We denote the $n \times n$ identity matrix as I and the vector of all ones as $\mathbf{1}_n$, respectively. The i'th element of a vector x is shown by x_i and the i'th row of a matrix A is represented by $(A)_i$. The induced matrix norm by vector norm $\|\,.\,\|$ is also shown by $\|A\|$ [26]. Let us define the collection of all feasible probability vectors p [15] as $\mathcal{P}_m = \{p \in \mathbb{R}_+^m \mid p^T \mathbf{1}_m = 1\}$. For a sequence of vectors $x = (x(1), x(2), \ldots, x(T))$, the ℓ_∞-norm of x is defined by

$$\|x\|_{\ell_\infty} = \max_{t \in \{1,\ldots,T\}} \|x(t)\|. \tag{1}$$

III. Problem Statement

Suppose there exist m unique pre-labeled environments for classification purposes. A robot is deployed into the environment to collect samples for classification. The sample, e.g., a vector that contains multiple states of the environment, from data sequence x. The data sequence x with length T can be represented in terms of its components as $x = (x(1), x(2), \ldots, x(T))$. Let us denote by $\mathcal{X}_k'$ the set which contains adequate sampled data sequences x from the k'th environment, i.e., the training set. Our proposed classifier, depicted in Fig. 2, is trained with $\mathcal{X}_k'$ for all $k = 1, ..., m$ to classify the label of x, which is sampled from an environment with unknown labels. In the classifier, each individual sample of x are fed recurrently to a RNN model [1], whose dynamics can be represented in a compact form by

$$h(t;x) = F\left(h(t-1;x), x(t)\right), \tag{2}$$

where $h(t;x) \in \mathbb{R}^b$ is the state and $x(t) \in \mathbb{R}^a$ is the input. The (history) state $h(t;x)$ memorizes all the past information about inputs x up to time t. The terminal state $h(T;x)$ is used for classification by passing it through a fully-connected layer

$$q(T;x) = W_c\, h(T;x) + b_c \tag{3}$$

and a `Softmax` function

$$p(T;x) = \texttt{Softmax}\left(q(T;x)\right), \tag{4}$$

where the weight matrix $W_c \in \mathbb{R}^{m \times b}$ and the bias vector $b_c \in \mathbb{R}^m$ are trainable. The belief vector $p(T;x)$ is utilized to represent the classification result from the sequence x.

Our *objective* is to provide a thorough robustness analysis of the classification model using stable RNNs and quantify their robustness bounds in terms of their trainable weight matrices.

Fig. 3: An input can be perturbed in various ways. This figures illustrate some possible cases for the map classification task.

IV. Origins of Perturbations

There exist several ways by which data can be collected as a sequence from the environment; for example, an agent can navigate in an environment and take localized observations [16] or a camera with a fixed location can capture images from a time-varying scene [9]. In most real-world applications, such raw observations are pre-processed, e.g., by neural networks, and the data sequence already carries

relevant features of the observed raw data. Let us assume that this pre-processing can be modeled by a nonlinear map

$$x(t) = \Phi\left(O(t)\right),$$

where $O(t) \in \mathcal{O}$ denotes the raw observation sampled at time t, and $\mathcal{O}$ is the space of all observables.

Perturbations can affect the data quality through several possible sources during the sampling process. The observer may deviate from its initially planned sampling routine due to dynamic noise in its motion planning [7] and take (slightly) deviated samples from nearby scenes in the environment. The raw observations may lose quality due to the environmental noise [3], e.g., change in light intensity, cloudiness, and blurriness. The raw observations may also experience various types of deformation, e.g., camera rotation or distortion [30]. Fig. 3 depicts some of these perturbations. The effect of uncertainty in all these cases can be modeled by

$$\tilde{x}(t) = \Phi\left(\tau\left(O(t)\right) + \xi(t)\right), \tag{5}$$

where $\tilde{x}(t)$ represents the perturbed data, $\xi : \mathbb{R} \to \mathcal{O}$ is an additive bounded stochastic noise or a bounded deterministic disturbance, and $\tau : \mathcal{O} \to \mathcal{O}$ is a deformation map that models sample deformation due to sensor movements (e.g., translation and rotation, and scaling).

V. Stable Recurrent Neural Networks and Their Error Estimates

In order to analyze the robustness of the classification model, we will evaluate how the input perturbation is affecting the classification result. Let us first identify what input sequences can generate correct classifications. Recall that the model is trained with sequences from $\mathcal{X}'_k$ for all $k = 1, ..., m$, and not every $x \in \mathcal{X}'_k$ will carry enough information to reveal the environment (e.g., a sequence consists of repeating scenes). Hence, only a subset of the sequences from the training set may generate the correct classification. We represent those data sequences that generate correct classifications by $\mathcal{X}^*_k$, where $\mathcal{X}^*_k \subseteq \mathcal{X}'_k$. This relation is depicted in Fig. 4.

To measure how the input perturbation will affect the classification, let us consider a nominal sequence $x \in \mathcal{X}^*_k$ and its corresponding perturbed sequence $\tilde{x} \in \mathcal{X}_k$ obtained from (5). The set $\mathcal{X}_k$ is the space of all possible sequences for the k'th class. In this section, we aim to evaluate the deviation generated by x and $\tilde{x}$ in terms of belief vectors, i.e., $p(T; x)$ and $p(T; \tilde{x})$. The first step is to consider the deviation generated at the output of the RNN, i.e., $h(t; x)$ and $h(t; \tilde{x})$.

In order to demonstrate our next result, let us introduce the concept of stable RNNs and its Lipschitz property. A stable RNN [20] provides the boundedness to its output when the input vectors are identical, i.e., $x(t) = \tilde{x}(t)$.

Definition 1. *A recurrent neural network model* (2) *is stable (contractive) if there exists a constant* $\lambda \in (0, 1)$ *such that for any* $h(t-1; x), h(t-1; \tilde{x}) \in \mathbb{R}^b$,

$$\|h(t; x) - h(t; \tilde{x})\| \leq \lambda \|h(t-1; x) - h(t-1; \tilde{x})\|, \tag{6}$$

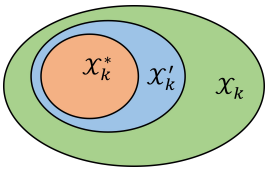

Fig. 4: This figure depicts the relation of the space of all possible sequences $\mathcal{X}_k$, the training set $\mathcal{X}'_k$, and the nominal set $\mathcal{X}^*_k$, i.e., $\mathcal{X}^*_k \subseteq \mathcal{X}'_k \subseteq \mathcal{X}_k$.

where $h(t; x) = F\big(h(t-1; x), x(t)\big)$ *and* $h(t; \tilde{x}) = F\big(h(t-1; \tilde{x}), x(t)\big)$.

It is also known that the RNN model is Lipschitz continuous with respect to its input [20, 12]. Then, for every two sequences x and $\tilde{x}$ with identical history state at $t-1$, i.e., $h(t-1; x) = h(t-1; \tilde{x})$, one has

$$\|h(t; x) - h(t; \tilde{x})\| \leq \kappa \, \|x(t) - \tilde{x}(t)\|, \tag{7}$$

where $\kappa \in \mathbb{R}_+$ is the real Lipschitz constant, $h(t; x) = F\big(h(t-1; x), x(t)\big)$, and $h(t; \tilde{x}) = F\big(h(t-1; x), \tilde{x}(t)\big)$. Then, the following result shows the upper bound for the deviation between the nominal belief $p(T; x)$ to the perturbed belief $p(T; \tilde{x})$ given a stable RNN model.

Theorem 1. *For the classification model with a stable RNN model, one has*

$$\|p(T; x) - p(T; \tilde{x})\| \leq \eta \, \|x - \tilde{x}\|_{\ell_\infty}$$

for every two sequences $x \in \mathcal{X}^*_k$ *and* $\tilde{x} \in \mathcal{X}_k$, *where*

$$\eta = \frac{\kappa \, \|W_c\|}{(1-\lambda)\sqrt{m}}. \tag{8}$$

Proof. The initial hidden states are set to $h(0; x) = 0$ and $h(0; \tilde{x}) = 0$ in both training and testing stage. Considering the Lipschitz continuity (7), the history states generated by sequences x and $\tilde{x}$ at $t = 1$ follows

$$\|h(1; x) - h(1; \tilde{x})\| \leq \kappa \|x(1) - \tilde{x}(1)\|. \tag{9}$$

For the next time step $t = 2$, we can obtain (10) by using the triangle inequality

$$\begin{aligned} \|h(2; x) - h(2; \tilde{x})\| &= \|h(2; x) - F\big(h(1; x), \tilde{x}(2)\big) \\ &\quad + F\big(h(1; x), \tilde{x}(2)\big) - h(2; \tilde{x})\| \\ &\leq \|F\big(h(1; x), x(2)\big) - F\big(h(1; x), \tilde{x}(2)\big)\| \\ &\quad + \|F\big(h(1; x), \tilde{x}(2)\big) - F\big(h(1; \tilde{x}), \tilde{x}(2)\big)\|. \end{aligned} \tag{10}$$

The first half of (10) can be bounded using (7),

$$\begin{aligned} \|F\big(h(1; x), x(2)\big) - F\big(h(1; x), \tilde{x}(2)\big)\| &\leq \kappa \, \|x(2) - \tilde{x}(2)\| \\ &\leq \kappa \, \|x - \tilde{x}\|_{\ell_\infty}. \end{aligned}$$

The second half of (10) can be bounded using (6) and then (9),

$$\begin{aligned} &\|F(h(1; x), \tilde{x}(2)) - F(h(1; \tilde{x}), \tilde{x}(2))\| \\ &\qquad \leq \lambda \, \|h(1; x) - h(1; \tilde{x})\| \\ &\qquad \leq \lambda\kappa \, \|x(1) - \tilde{x}(1)\| \leq \lambda\kappa \, \|x - \tilde{x}\|_{\ell_\infty}. \end{aligned}$$

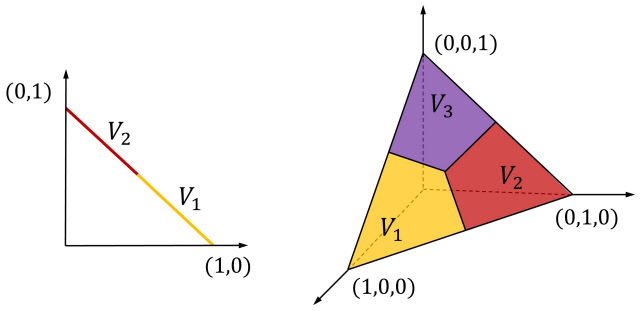

Fig. 5: This figure depicts the Voronoi partition of the classification sets $\mathcal{P}_m$ for both $m = 2$ (left) and $m = 3$ (right).

Summarizing the above inequalities, the deviation of history states at $t = 2$ obtains the following upper bound,

$$\|h(2;x) - h(2;\tilde{x})\| \leq (1+\lambda)\,\kappa\,\|x - \tilde{x}\|_{\ell_\infty}.$$

Repeating the above steps up to t, the deviation of $\|h(t;x) - h(t;\tilde{x})\|$ is upper bounded by

$$\begin{aligned}\|h(t;x) - h(t;\tilde{x})\| &\leq (1+\lambda+...+\lambda^{t-1})\kappa\ \|x-\tilde{x}\|_{\ell_\infty} \\ &\leq \frac{1-\lambda^t}{1-\lambda}\kappa\ \|x-\tilde{x}\|_{\ell_\infty}.\end{aligned}$$

Given that $0 < \lambda < 1$, the terminal deviation at T obtains the following boundedness,

$$\|h(T;x) - h(T;\tilde{x})\| \leq \frac{\kappa}{1-\lambda}\ \|x-\tilde{x}\|_{\ell_\infty}. \tag{11}$$

The similar bounds can be obtained for the fully connected layer (3),

$$\|q(T;x) - q(T;\tilde{x})\| \leq \|W_c\|\frac{\kappa}{1-\lambda}\ \|x-\tilde{x}\|_{\ell_\infty}.$$

Then, the results follows immediately by considering the fact that the `Softmax` function is $1/\sqrt{m}$ Lipshcitz continuous [13] with respect to its input $q \in \mathbb{R}^m$. □

The above theorem asserts that, in the classifier, if the RNN model is stable, the terminal belief difference is bounded by the maximum deviation along the input sequences. This result provides the knowledge of the classifier's robustness by identifying an upper bound for the error generated by the perturbed and the nominal data.

VI. Classification Criterion and Convergence Conditions

In this part, we aim to represent the $\arg\max$ classification criterion in a quantifiable manner and use the previously obtained deviation bounds in belief vectors to perform the robustness analysis.

The classification process with x is accomplished by identifying $\arg\max p(T;x)$ as the class label, i.e., it will conclude as the k'th class if and only if $p(T;x)_k > p(T;x)_j$ for all $j = 1,\ldots,m$ and $j \neq k$. This criterion can be explicitly represented via the Voronoi partitioning [11]. Let us denote by $V_1, V_2, \ldots, V_m$ the Voronoi partition of the probability vector space $\mathcal{P}_m$, i.e.,

$$V_k = \{p \in \mathcal{P}_m \mid p_k > p_j,\ \forall j \neq k\}. \tag{12}$$

Some examples of the Voronoi partitioning is shown in Fig. 5. The above partition is equivalent to the $\arg\max$ classification criterion, i.e., $\arg\max p(T;x) = p(T;x)_k$ if and only if $p(T;x) \in V_k$.

The next step is to convert the deviation $\|p(T;x) - p(T;\tilde{x})\|$ into a Boolean classification result, i.e., "true" or "false". To reveal our next result, let us introduce the distance function between vectors and sets.

Definition 2. *The distance between two vectors $p, p' \in \mathcal{P}_m$ is defined as*

$$d(p,p') = \|p - p'\|$$

and the distance between a vector and a set is defined as

$$d(p,V_j) = \inf_{p' \in V_j} d(p,p').$$

Let us express the $\arg\max$ classification criterion equivalently using the distance function.

Lemma 1. *A belief vector $p \in \mathcal{P}_m$ will be classified as the k'th class if and only if*

$$d(p,V_k) = 0, \tag{13}$$

or if and only if

$$d(p,V_j) > 0 \text{ for all } j \neq k. \tag{14}$$

Proof. Suppose there exists a $p' \in V_k$ such that $d(p,p') = 0$, then one has $p = p'$ and $p \in V_k$. On the other hand, if $p \in V_k$, then we have $d(p,V_k) \leq d(p,p) = 0$ and $d(p,V_k) = 0$ by definition.

For the equivalence, since $V_k \bigcap V_j = \emptyset$ for all $j \neq k$, one has if $d(p,V_k) = 0$, then $d(p,V_j) > 0$ for any V_j. On the other hand, consider the fact that

$$V_k \subset \mathcal{P}_m \setminus \bigcup_{j \neq k} V_j,$$

one has if $d(p,V_j) > 0$, then $p \notin V_j$ and $p \in V_k$ [1]. □

The above result introduces a quantifiable classification criterion, which can be used in all classification problems using the $\arg\max$ classifier. Given a belief vector $p \in \mathcal{P}_m$, one can obtain the classification result by checking if (13) or (14) are satisfied.

Using Lemma 1, we can establosh a connection between $\|p(T;x) - p(T;\tilde{x})\|$ and the robustness of the classification model. For a robust classification model, we expect $p(T;\tilde{x})$ to be located within V_k. This relation can be validated by comparing $d(p(T;x),V_j)$ with $\|p(T;x) - p(T;\tilde{x})\|$ for all $j = 1,...,m$ and $j \neq k$. In order to accomplish the analysis for an arbitrary perturbed sequence $\tilde{x}$, let us introduce the concept of the robustness radius.

Definition 3. *For all nominal sequences $x \in \mathcal{X}_k^*$, let us consider the robustness radius for the k'th class label as*

$$\varepsilon_k = \min_{x \in \mathcal{X}_k^*,\, j \neq k} d\big(p(T;x), V_j\big), \tag{15}$$

[1]The cases of p located on the boundary of the Voronoi partition is considered trivial since its probability of happening is 0.

where $j = 1, \ldots, m$.

The robust radius quantifies the minimal distance from a nominal belief vector $p(T; x)$ to the boundary of V_k, which enables us determine when the classification with $\tilde{x} \in \mathcal{X}_k$ is robust, i.e., the result is the same with the one generated by some $x \in \mathcal{X}_k^*$.

Theorem 2. *Suppose that the RNN model* (2) *is stable. The classification generated by a perturbed sequence* $\tilde{x} \in \mathcal{X}_k$ *is robust for the* k*'th class, if there exist some* $x \in \mathcal{X}_k^*$ *such that*

$$\|x - \tilde{x}\|_{\ell_\infty} < \varepsilon_k \eta^{-1}, \tag{16}$$

where $\|x - \tilde{x}\|_{\ell_\infty}$ *and* η *are defined in* (8) *and* (1).

Proof. In the view of the k'th class, the classification with $\tilde{x} \in X$ is robust if

$$d(\, p(T; \tilde{x}), V_j \,) = \inf_{p' \in V_j} d(p(T; \tilde{x}), p') > 0, \tag{17}$$

for all $j = 1, \ldots, m$ and $j \neq k$. The above quantity obtains a lower bound using the triangle inequality,

$$\inf_{p' \in V_j} d(p(T; \tilde{x}), p') \geq \inf_{p' \in V_j} d(p(T; x), p') - d(p(T; x), p(T; \tilde{x})),$$

for any $\tilde{x} \in \mathcal{X}_k$ and $x \in \mathcal{X}_k^*$. Then, the inequality (17) will be satisfied if

$$\inf_{p' \in V_j} d(\, p(T; x), p' \,) - d(\, p(T; x), p(T; \tilde{x}) \,) > 0,$$

which is equivalent to $\|p(T; x) - p(T; \tilde{x})\| < \varepsilon_k$. Then, we can conclude by applying Theorem 1 to the inequality above. □

The above theorem states that in the classification model with stable RNNs, a perturbed data sequence $\tilde{x}$ will generate a correct classification result if there exist some nominal sequences $x \in \mathcal{X}_k^*$ that satisfy (16). In other words, the constant $\varepsilon_k \eta^{-1}$ measures maximal allowed deviation for the input to perform robust classification on the k'th class, i.e., for any perturbed sequence $\tilde{x}$ that satisfies (16), the classification result is guaranteed to be correct. In Fig. 6, we present the idea of this relation with a simplified example. We highlight that the above result provides a sufficient condition, so $\tilde{x}$ may still generate a correct classification if (16) is not satisfied.

VII. Case Study and Simulations

We use the map and image classification [16] as examples to demonstrate and validate our theoretical results. In the case study, it is assumed that an aerial robot with a downward-facing camera aims to classify the underlying image (or map). However, due to the limited sensing capability, the robot can only observe a localized portion of the image at each time step, and it is allowed to traverse the environment to collect partial images as a time-indexed sequence.

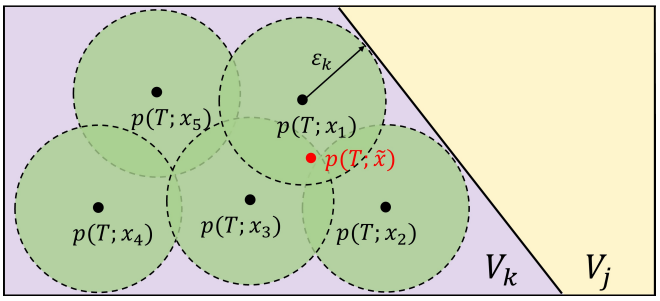

Fig. 6: This figure depicts nominal belief vectors in the k'th class $p(T; x_1), ..., p(T; x_5)$, and the perturbed belief vector $p(T; \tilde{x})$. The model will perform a robust classification if $p(T; \tilde{x})$ is within the distance of ε_k to some $p(T; x_i)$, where $x_i \in \mathcal{X}_k^*$.

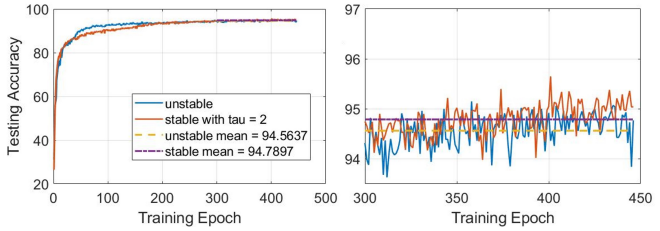

Fig. 7: This figure depicts the training progress with and without the stability constraint (18) on the MNIST dataset.

A. Training for Classification

We consider the sampling routine for the robot is given and fixed for both training and testing stage [2]. Data sequences are generated with a fixed length T. In the case study, a VGG-19 [27] model is adopted to process the raw observations $O(t) \in \mathcal{O}$, such that $x(t) = VGG\big(O(t)\big)$. The robot also uses a stable LSTM cell [8], which is a special case of RNN, to recurrently process the input. A stable (contractive) LSTM can be learned by introducing the following constraints in the training stage [3]

$$\max\left\{\|W_u\|_\infty, \|W_o\|_\infty, 4\|W_z\|_\infty, \sqrt{\|W_f\|_\infty}\right\} < 1 - \|f\|_\infty. \tag{18}$$

When the above constraint is not satisfied during the training, each row of the LSTM weight matrices will be divided by a constant scaling factor $\tau > 1$ after each gradient step until (18) is satisfied.

In the training stage, we evaluate the classification reward with the data sequence $x \in \mathcal{X}_k'$ by a log-sum-exp (LSE) loss as $r = -LSE\big(e_k, p(T; x)\big)$, in which k denotes the ground truth label.

B. MNIST Dataset

In the first case study, we consider the robot is traveling over the image from the MNIST dataset [14, 22]. The training and testing is performed in PyTorch [23] with

[2]For each map (or image), we generate five unique paths for a robot to traverse, and they are fixed through the training and testing. However, robots are also capable of planning the path based on their observation, see [17, 21]

[3]In our notation, $\|f\|_\infty = \sup_t \|f_{t \geq 0}\|_\infty$ and f_t is the output of the forget gate of the LSTM cell. We refer to [20] for the details.

Dataset	Observation Size (%)	Stability Constraint	Scaling Factor τ	Classification Accuracy (%)	Benchmark (VGG-19)
Map Dataset with clouds	$\leq$7.8	No	-	77.62	99.43
		Yes	1.01	77.23	
MNIST	$\leq$ 68.88	No	-	94.56	99.33
		Yes	1.05	94.34	
		Yes	1.1	94.90	
		Yes	2	95.05	
		Yes	4	94.20	
		Yes	8	94.04	

TABLE I: The performance measured with different observation size, stability condition and dataset.

ADAM [10] and a learning rate $l_r = 0.0001$. The testing result is presented in Table I, which is validated over five random seeds.

To establish a benchmark, we train an independent VGG-19 model with the entire image (map) as the input, shown in the last column of Table I. In the second column, the observation size denotes the maximum possible coverage of the entire image or map. It is shown that a single robot can classify by only revealing a small portion of the environment. It should be emphasized that the performance on the MNIST will reach 98% by using multiple communicating robots [22].

C. Robustness Analysis on the MNIST Dataset

To reveal how a stable RNN model will affect the performance of the classifier, we test both unstable and stable models with various scaling factors τ in the MNIST dataset. As shown in both Fig. 7 and Table I, a stable model has a comparable performance and sometimes it even outperforms the unstable model for certain values of τ.

1) Constant deviation: To investigate the robustness of the classifier, we first obtain the nominal sequences[4] from the training set $\mathcal{X}'_k$ and measure the performance of the model by adding a constant deviation $\xi \in [0, 10]$ to the nominal sequences to get

$$\tilde{x}(t) = x(t) + \xi \mathbf{1}_a,$$

where $\mathbf{1}_a$ is the vector of all ones. This implies that $\|x - \tilde{x}\|_{\ell_\infty} = \xi$. The test is performed on the 5'th class of the MNIST dataset for the unstable and stable models with $\tau = 1.05$. The result is shown in Fig. 8, and it implies that, for the stable model, if the deviation of the input $\|x - \tilde{x}\|_{\ell_\infty}$ is less than $\varepsilon_5/\eta = 0.0886$, the classification with $\tilde{x}$ is guaranteed to be robust, i.e., with a 100% accuracy. This agrees with our theoretical result. However, the unstable model starts to generate wrong predictions before the quantity $\|x - \tilde{x}\|_{\ell_\infty}$ reaches 0.0886.

We highlight that for the stable model the accuracy remains 100% even for some $\|x - \tilde{x}\|_{\ell_\infty} > \varepsilon_5/\eta$. This is because our theoretical result provides a sufficient condition, which is usually conservative. As we observe from the simulations, there is still a chance to get robust classification when (16) is not satisfied. Furthermore, it is interesting to notice that the unstable model starts to outperform the stable model when $\|x - \tilde{x}\|_{\ell_\infty} \geq 0.2$. The reason is that by imposing the stability constraint (18) one only require the stable model to *confidently* exhibit the robust classification with the input perturbation less than ε_k/η, instead of concerning the performance with $\|x - \tilde{x}\|_{\ell_\infty} \geq \varepsilon_k/\eta$. The stable model provides the confidence of 100% accuracy with all deviations $\|x - \tilde{x}\|_{\ell_\infty} < 0.0886$, which is not guaranteed for the unstable models, as shown in the shaded area in Fig. 8. This difference is crucial when the robot is performing high-precision tasks, in which any level of mistake is not acceptable. On the other hand, the stable model that focuses on improving the performance with the Gaussian input noise has been proposed and validated in our previous work [1].

[4]A sequence is called nominal if its resulting classification accuracy is 100%.

Stability Constraint	τ	Unperturbed Sequence	Perturbed Sequence	Performance Loss
No	-	95.88	94.56	**1.3163**
Yes	1.05	94.68	94.34	0.3396
Yes	1.1	95.24	**94.90**	0.3416
Yes	2	95.50	**95.05**	0.4467
Yes	4	94.40	94.20	0.1978
Yes	8	94.14	94.04	0.1062

TABLE II: The performance loss under perturbation with various scaling factor τ on the MNIST dataset.

2) Variable deviation: It is also interesting to see how the model will perform with the variable deviations instead of the constant ones. The differences from the testing dataset to the training dataset can be naturally considered variable deviations since the variation of handwritten digits from the training to the testing set can not be modeled as the constant deviation. Hence, we compare the performances of different models on both training and testing sets, see Fig. 9 and Table II, in which we denote the training set as "unperturbed" and the testing set as "perturbed."

The results of variable deviation show that the performance loss from unperturbed data to the perturbed data is significantly reduced by using a stable model. In addition, the stable model sometimes outperforms the unstable model if a proper scaling factor is selected, e.g., $\tau = 1.1$ and $\tau = 2$. We also highlight that there exists an potential trade-off between the performance loss and the overall performance: As shown in Table II, a higher scaling factor τ usually implies a more stable RNN and less performance loss, but also a weaker overall performance since the stability constraint will potentially drive the model away from the optimal classifier.

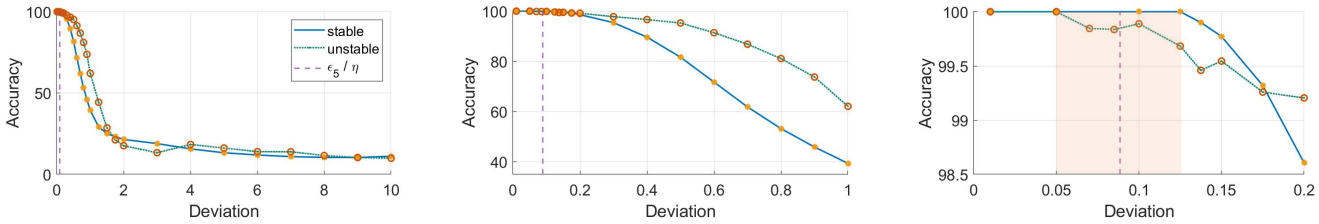

Fig. 8: This figure depicts the classification accuracy when the constant perturbation has been added to the nominal sequences in the 5'th class in the MNIST dataset with $\varepsilon_5/\eta = 0.0886$. From left to right are magnified details. The accuracy eventually converges to 10% since the total number of labels is $m = 10$, and a random guess has $1/10$ of the chance to get the correct label.

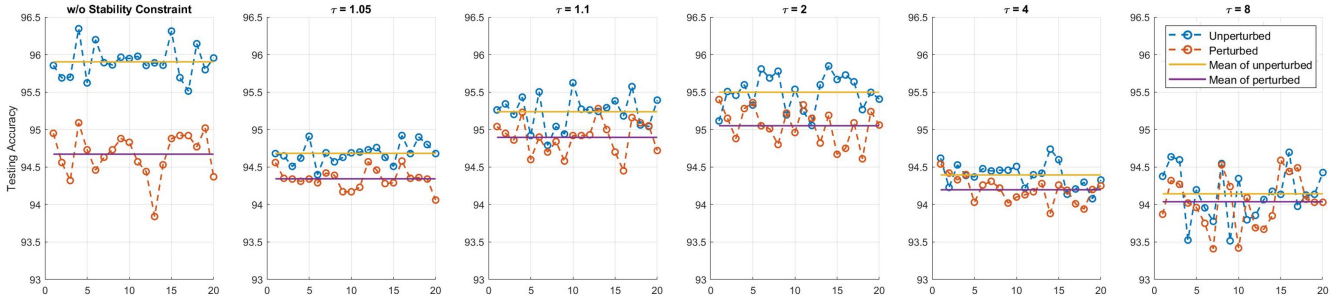

Fig. 9: The above figures depict performance with both unperturbed and perturbed data for all models trained with various scaling factor τ on the MNIST dataset. Each data point represents the average result from 20000 sampled data sequences.

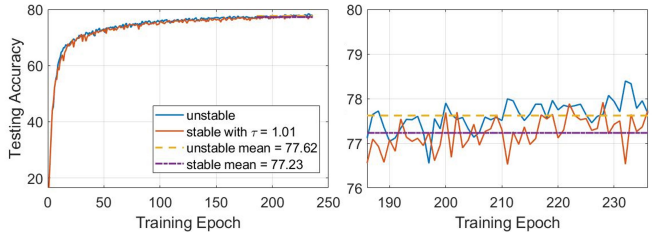

Fig. 10: This figure depicts the training progress with and without the stability constraint (18) on the Campus map dataset.

D. Campus Map Dataset

In the second case study, we test our model on the Campus Map dataset [17]. The training and testing are performed in the same platform with the first case study with a learning rate $l_r = 0.00001$. The testing result is presented in Table I and Fig. 10, which is validated over five random seeds. It is shown that the stable model obtains a comparable performance as the unstable model on the Campus map dataset and enjoys the robustness guarantee when the input perturbation satisfies certain conditions. It should also be emphasized that the performance of the map classification will reach 97% by using a team of communicating robots [16].

We also successfully performed a real-world experiment of map classification with aerial robots. Some snapshots taken from the experiments are shown in Fig. 11, and a full experiment video can be found at `https://youtu.be/nsnPFAvJLoY`.

VIII. Conclusion

We present a framework to analyze the robustness properties of stable RNNs with sequential inputs for classification purposes. It is shown that every trained RNN exhibits robust classification with respect to some bounded perturbations. We quantify robustness bounds in terms of trainable weight matrices. Our results are significant as they reveal interplay among various design (trainable) parameters. Our extensive simulations and one real-world experiment support and validate the usefulness of our theoretical findings.

References

[1] A. Amini, G. Liu, and N. Motee. "Robust Learning of Recurrent Neural Networks in Presence of Exogenous Noise". In: *2021 60th IEEE Conference on Decision and Control (CDC)*. 2021, pp. 783–788.

[2] M. Bock, J. Böhner, O. Conrad, R. Köthe, and A. Ringeler. "XV. Methods for creating Functional Soil Databases and applying Digital Soil Mapping with SAGA GIS". In: *JRC Scientific and technical Reports, Office for Official Publications of the European Communities, Luxemburg* (2007).

[3] R. A. Boie and I. J. Cox. "An analysis of camera noise". In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 14.06 (1992), pp. 671–674.

[4] A. Breitenmoser, M. Schwager, J.-C. Metzger, R. Siegwart, and D. Rus. "Voronoi coverage of non-convex environments with a group of networked robots". In: *2010 IEEE international conference on robotics and automation*. IEEE. 2010.

[5] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, and A. Kurakin. "On evaluating adversarial robustness". In: *arXiv preprint arXiv:1902.06705* (2019).

[6] F. De la Torre and M. J. Black. "Robust principal component analysis for computer vision". In: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*. Vol. 1. IEEE. 2001, pp. 362–369.

Fig. 11: The above figures are snapshots taken while the aerial robot is flying over the map of the Lehigh University campus. The robot's location and observation are denoted with the red and the yellow box. The bar plot shows the real-time belief vector. Eventually, the robot correctly classifies the map as the Lehigh University (gray bar). For more detail, please see the full experiment video at `https://youtu.be/nsnPFAvJLoY`.

[7] N. E. Du Toit and J. W. Burdick. "Robot motion planning in dynamic, uncertain environments". In: *IEEE Transactions on Robotics* 28.1 (2011), pp. 101–115.

[8] S. Hochreiter and J. Schmidhuber. "Long Short-term Memory". In: *Neural computation* 9 (Dec. 1997), pp. 1735–80.

[9] T. S. Huang and R. Tsai. "Image sequence analysis: Motion estimation". In: *Image sequence analysis*. Springer, 1981.

[10] D. P. Kingma and J. Ba. *Adam: A Method for Stochastic Optimization*. 2014. arXiv: `1412.6980 [cs.LG]`.

[11] R. Klein. "Abstract Voronoi diagrams and their applications". In: *Workshop on Computational Geometry*. Springer. 1988.

[12] C.-Y. Ko, Z. Lyu, L. Weng, L. Daniel, N. Wong, and D. Lin. "POPQORN: Quantifying robustness of recurrent neural networks". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 3468–3477.

[13] D. Kohli. *Machine Learning: Is the softmax function Lipschitz with Lipschitz constant 1?* Mathematics Stack Exchange. URL:https://math.stackexchange.com/q/2021011.

[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998).

[15] C. Li and S. Zhang. "Stationary probability vectors of higher-order Markov chains". In: *Linear Algebra and Its Applications* 473 (2015), pp. 114–125.

[16] G. Liu, A. Amini, M. Takáč, H. Muñoz-Avila, and N. Motee. *Distributed Map Classification using Local Observations*. 2020. arXiv: `2012.10480 [cs.RO]`.

[17] G. Liu, A. Amini, M. Takáč, and N. Motee. "Classification-Aware Path Planning of Network of Robots". In: *International Symposium Distributed Autonomous Robotic Systems*. Springer. 2021, pp. 294–305.

[18] D. Lowd and C. Meek. "Adversarial learning". In: *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. 2005, pp. 641–647.

[19] A. Matyasko and L.-P. Chau. "Improved network robustness with adversary critic". In: *Advances in Neural Information Processing Systems* 31 (2018).

[20] J. Miller and M. Hardt. "Stable Recurrent Models". In: *International Conference on Learning Representations*. 2018.

[21] H. K. Mousavi, G. Liu, W. Yuan, M. Takáč, H. Muñoz-Avila, and N. Motee. *A Layered Architecture for Active Perception: Image Classification using Deep Reinforcement Learning*. 2019. arXiv: `1909.09705 [cs.LG]`.

[22] H. K. Mousavi, M. Nazari, M. Takáč, and N. Motee. "Multi-Agent Image Classification via Reinforcement Learning". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019.

[23] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. "Automatic differentiation in pytorch". In: (2017).

[24] V. Ramesh and R. M. Haralick. "Random perturbation models and performance characterization in computer vision". In: *Proceedings 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society. 1992, pp. 521–522.

[25] A. Robey, H. Hu, L. Lindemann, H. Zhang, D. V. Dimarogonas, S. Tu, and N. Matni. "Learning control barrier functions from expert demonstrations". In: *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE. 2020.

[26] W. Rudin. *Real and Complex Analysis*. McGraw-Hill, 1974.

[27] K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *3rd International Conference on Learning Representations* (2015).

[28] J. Solomon. *Numerical algorithms: methods for computer vision, machine learning, and graphics*. CRC press, 2015.

[29] N. Sünderhauf, O. Brock, W. Scheirer, R. Hadsell, D. Fox, J. Leitner, B. Upcroft, P. Abbeel, W. Burgard, M. Milford, et al. "The limits and potentials of deep learning for robotics". In: *The International Journal of Robotics Research* 37.4-5 (2018), pp. 405–420.

[30] Z. Tang, R. G. von Gioi, P. Monasse, and J.-M. Morel. "A precision analysis of camera distortion models". In: *IEEE Transactions on Image Processing* 26.6 (2017).

[31] K. Xu, S. Liu, P. Zhao, P.-Y. Chen, H. Zhang, Q. Fan, D. Erdogmus, Y. Wang, and X. Lin. "Structured adversarial attack: Towards general implementation and better interpretability". In: *arXiv preprint arXiv:1808.01664* (2018).