Investigating the Robustness of Sequence Models in Deep Learning

u1617871

Department of Physics, University of Warwick, CV4 7AL, Coventry, UK

Contents

Abstract

The robustness of sequence models to adversarial attack is a vital area of research in Machine Learning that is yet to be extensively explored. In this paper, we extend the gaps in existing literature, providing a relationship between robustness and decision boundary curvature. In particular, we theoretically prove classifiers with less curved decision boundaries are more robust. This is in agreement with existing results for traditional feedforward Neural Networks, suggesting universality across deep learning architectures. We provide local bounds of robustness based on decision boundary curvature in a deterministic setting. For a general data point, we derive a probabilistic bound for the probability of misclassification on the ball of radius ρ as a function of perturbation radius ρ. Noting the relationship between loss function curvature and robustness [14], we extend the work of [18], approximating the spectral distribution of a single layer recurrent neural network (RNN) with cross-entropy loss, showing consistency with results for traditional NNs. Furthermore, we propose a Hessian regulariser based on the classifier Hessian Frobenius norm that we experimentally show to be an improvement on existing Jacobian regularisation techniques on an RNN trained on the MNIST dataset.

1 Introduction

While the accuracy of Machine Learning (ML) models has improved for a variety of applications, the reliability of these models has been put into question. A classifier that predicts the correct label of a data point with high confidence can be fooled into making a wrong classification with high confidence as well [14, 11]. Corruptions of a data set imperceptible to the human eye have been shown to repeatedly fool state-of-the-art image classifiers [11]. In addition, [14] show the existence of universal adversarial perturbations which result in misclassification when applied to every data point. This significant shortcoming brings question to the practicality of applying machine learning models to safety-critical systems. The consequence of a false negative in a medical diagnostic tool or a false prediction in a self-driving system could possibly result in the loss of a life.

The robustness of ML models to adversarial attack has been extensively studied in the context of image classification systems. This project aims to extend this work to sequence models trained on temporally variable or highly ordered data. Widely used deep learning

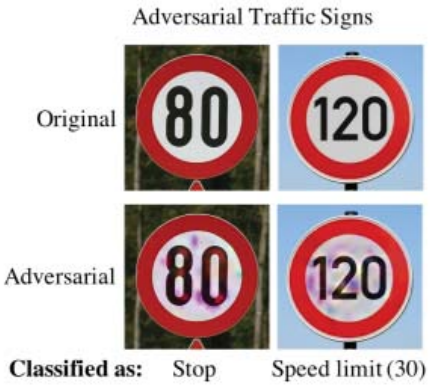

Figure 1: A Classifier can be fooled to identify an 80kph speed limit sign as a stop sign and a 120kph speed limit sign as a stop sign [20].

sequence models include recurrent neural networks (RNNs), Long Short Term Memory networks (LSTMs) and transformers. Applications of these models include making predictions of trends in financial time series data [5] and predicting exoplanets with light curve data from satellite missions [16]. The aims of this project include:

- Characterising the geometric properties of the decision boundary and loss surface of robust sequence models. It has been shown that decision boundaries which are less curved result in more robust models for image classification systems [14]. We investigate if this holds for sequence models, providing theoretical justification.

- Determining the robustness of sequence models in the probabilistic setting, identifying the probability of misclassification on the ball of radius ρ as a function of perturbation radius, ρ.

- Investigate the effect of curvature regularisation on the robustness of sequence models. If these regularisation techniques improve robustness, it would experimentally support the hypothesis that sequence models with less curved decision boundaries achieve greater robustness.

2 Theory

2.1 Preliminaries

Let $f : \mathbb{R}^d \to \mathbb{R}^L$ be an arbitrary multi-class classifier. We shall assume f is continuous and twice differentiable throughout. Given a data point $\mathbf{x} \in \mathbb{R}^d$, the class which f predicts for $\mathbf{x}$ is given by $\hat{k}(\mathbf{x}) = \arg\max_k f_k(\mathbf{x})$ where $f_k(\mathbf{x})$ is the k-th component of $f(\mathbf{x})$. Note $\hat{k}(\mathbf{x})$ need not be unique. In this case, w.l.o.g we take $\hat{k}(\mathbf{x})$ to be the first component where f achieves its maximum.

Definition 2.1.1 (Confidence of Classification)**.** *The confidence of classification, $F(\mathbf{x})$ made by the classsifier f at $\mathbf{x} \in \mathbb{R}^d$ is defined as:*

$$F(\mathbf{x}) = f_{\hat{k}(\mathbf{x})}(\mathbf{x}) - \max_{k \neq \hat{k}(\mathbf{x})} f_k(\mathbf{x}) \tag{1}$$

$F(\mathbf{x})$ describes the difference between the likelihood of classification for the most probable class and the second most probable class. Note that this second most probable class need not be unique. As before, w.l.o.g we take the first component for which f achieves its second maximum. The larger the value of $F(\mathbf{x})$, the more confident we are in the prediction $\hat{k}(\mathbf{x}) = \arg\max_k f_k(\mathbf{x})$ given by the classifier for point $\mathbf{x}$. Since f is continuous (by assumption), it can be shown that F is also continuous. This follows from the fact that the maximum of a set of continuous functions is continuous. A similar argument shows this is also the case for the second maximum.

Definition 2.1.2 (Decision Boundary)**.** *The decision boundary of a classifer, B is defined as:*

$$B = \left\{\mathbf{x} \in \mathbb{R}^d \mid F(\mathbf{x}) = 0\right\} \tag{2}$$

The set of points the classifier f is equally likely to classify into two distinct classes forms the decision boundary B. B splits the domain $\mathbb{R}^d$ into regions of similar classification, $C_k = \{\mathbf{x} \in \mathbb{R}^d \text{ s.t } \hat{k}(\mathbf{x}) = k, \quad 1 \leqslant k \leqslant L\}$. That is, $\mathbb{R}^d \backslash B = \bigcup_{k=1}^{L} C_k$ (we assume each C_k has a non-empty interior). Therefore, $\mathbf{x} \notin B \implies \mathbf{x} \in C_k$. For a perturbation $\boldsymbol{\delta}(\mathbf{x})$ s.t $\mathbf{x} + \boldsymbol{\delta}(\mathbf{x}) \in B$, we have that $\mathbf{x} + \boldsymbol{\delta}(\mathbf{x})$ is on the boundary of misclassification. Hence, when considering misclassification, we study properties of the decision boundary B.

In practice and throughout this paper, we consider the adversarial perturbation as a perturbation of minimal length to the decision boundary B. Similarly, the adversarial

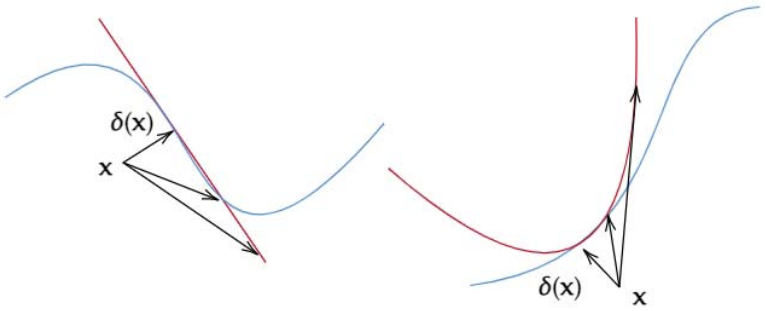

Figure 2: First order approximation (left) and second order approximation (right) to the decision boundary, B at a point. The true decision boundary is given by the curves in blue while the approximations are given by the curves in red. $\boldsymbol{\delta}(\mathbf{x})$ is the minimal perturbation to B.

distance is the minimal length of a perturbation to B. In general, B is intractable, which motivates approximations of the decision boundary, given below:

Definition 2.1.3 (Approximation of the decision boundary)**.** *Let* $\mathbf{x} \in \mathbb{R}^d$ *and* $\boldsymbol{\delta}(\mathbf{x}) \in \mathbb{R}^d$ *be a perturbation vector of minimal distance s.t* $\mathbf{z} = \mathbf{x} + \boldsymbol{\delta}(\mathbf{x}) \in B$.

The linear, first order approximation of the decision boundary at $\mathbf{z} = \mathbf{x} + \boldsymbol{\delta}(\mathbf{x})$ *is given by the set:*

$$\mathbf{x} + \{\mathbf{v} : \boldsymbol{\delta}(\mathbf{x})^T \mathbf{v} = ||\boldsymbol{\delta}(\mathbf{x})||_2^2\} \tag{3}$$

The quadratic, second order approximation of the decision boundary B*, is given by:*

$$\mathbf{x} + \{\mathbf{v} : \frac{1}{2}(\mathbf{v} - \boldsymbol{\delta}(\mathbf{x}))^T(\mathbf{H_z})(\mathbf{v} - \boldsymbol{\delta}(\mathbf{x})) + \alpha_{\mathbf{x}}(\boldsymbol{\delta}(\mathbf{x}))^T(\mathbf{v} - \boldsymbol{\delta}(\mathbf{x})) = 0\} \tag{4}$$

where $\alpha_{\mathbf{x}} = \|\nabla F(\mathbf{z})\|/\|\boldsymbol{\delta}(\mathbf{x})\|$

For the quadratic approximation, we consider the first terms of the Taylor expansion of $F(\mathbf{x})$ around $\mathbf{z} = \mathbf{x} + \boldsymbol{\delta}(\mathbf{x})$.

$$F(\mathbf{x} + \mathbf{v}) \approx F(\mathbf{z}) + (\mathbf{v} - \boldsymbol{\delta}(\mathbf{x}))^T \nabla F(\mathbf{z}) + \frac{1}{2}(\mathbf{v} - \boldsymbol{\delta}(\mathbf{x}))^T(\mathbf{H_z})(\mathbf{v} - \boldsymbol{\delta}(\mathbf{x})) \tag{5}$$

Note $\nabla F(\mathbf{z})/\|\nabla F(\mathbf{z})\| = \boldsymbol{\delta}(\mathbf{x})/\|\boldsymbol{\delta}(\mathbf{x})\|$ and $\mathbf{H_z}$ denotes the Hessian (w.r.t inputs) at $\mathbf{z}$.

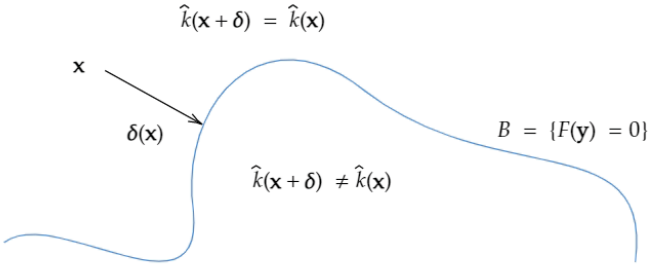

Figure 3: Decision boundary, B and minimal perturbation to B, $\boldsymbol{\delta}$. Note that on opposite sides of B, the classifier makes a different prediction. This is our motivation for reformulating adversarial perturbation and distance in terms of the decision boundary

Definition 2.1.4 (Decision Boundary with Confidence)**.** *Given a confidence margin $t > 0$, the decision boundary with thresholding or confidence margin t is given by:*

$$B'(t) = \left\{\mathbf{x} \in \mathbb{R}^d \mid |F(\mathbf{x})| < t\right\} \tag{6}$$

In other words, $\mathbf{x} \notin B'(t)$ implies $\mathbf{x}$ is assigned class $\hat{k}(\mathbf{x})$ by the classifier f, with confidence $t > 0$. All points $\mathbf{x} \in B'(t)$ are not classified by confidence at least t.

Definition 2.1.5 (Adversarial Perturbation)**.** *An adversarial perturbation, $\boldsymbol{\delta}_{\boldsymbol{adv}}(\mathbf{x}; f)$ is defined by the following optimisation problem:*

$$\boldsymbol{\delta}_{\boldsymbol{adv}}(\mathbf{x}; f) = \arg\min_{\boldsymbol{\delta} \in \mathbb{R}^d} \|\boldsymbol{\delta}\|_2 \ s.t \ F(\mathbf{x} + \boldsymbol{\delta}) = 0 \tag{7}$$

In other words, $\boldsymbol{\delta}_{\boldsymbol{adv}}(\mathbf{x}; f)$ is a perturbation vector of minimal length to the decision boundary. Note $\boldsymbol{\delta}_{\boldsymbol{adv}}(\mathbf{x}; f)$ need not be unique. In this case, w.l.o.g we select a valid perturbation vector at random and assign it to $\boldsymbol{\delta}_{\boldsymbol{adv}}(\mathbf{x}; f)$. Point $\mathbf{x} + \boldsymbol{\delta}_{\boldsymbol{adv}}(\mathbf{x}; f)$ will then be on the boundary of misclassification. Note an adversarial perturbation is typically defined as the minimal distance to mislcassification, $\min_{\boldsymbol{\delta} \in \mathbb{R}^d} \|\boldsymbol{\delta}\|_2 \ s.t \ \hat{k}(\mathbf{x} + \boldsymbol{\delta}) \neq \hat{k}(\mathbf{x})$. However, it is easier to consider a perturbation to the decision boundary and we shall consider these two frameworks as identical throughout this paper. This is because once on the decision boundary, an infinitesimal perturbation will result in misclassification.

Definition 2.1.6 (Adversarial Distance)**.** *An adversarial distance, $\delta_{adv}(\mathbf{x}; f)$ is defined as:*

$$\delta_{adv}(\mathbf{x}; f) = \|\boldsymbol{\delta}_{\boldsymbol{adv}}(\mathbf{x}; f)\| = \min_{\boldsymbol{\delta} \in \mathbb{R}^d} \|\boldsymbol{\delta}\|_2 \; s.t \; F(\mathbf{x} + \boldsymbol{\delta}) = 0 \tag{8}$$

In other words, $\delta_{adv}(\mathbf{x}; f)$ is the minimal length of a perturbation to the decision boundary. Given $\alpha \in \mathbb{R}$, if $\alpha \geqslant \delta_{adv}(\mathbf{x}; f)$, then $\exists \boldsymbol{\delta} \in \mathbb{S} = \left\{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x}\| = 1\right\}$ such that $F(\mathbf{x} + \alpha\boldsymbol{\delta}) = 0$. On the other hand, if $\alpha < \delta_{adv}(\mathbf{x}; f), \forall \boldsymbol{\delta} \in \mathbb{S}$ we have that $F(\mathbf{x} + \alpha\boldsymbol{\delta}) \neq 0$. All perturbations of magnitude less than α cannot reach the decision boundary B. The larger the value of $\delta_{adv}(\mathbf{x}; f)$, the more robust we say the point $\mathbf{x} \in \mathbb{R}^d$ is to adversarial perturbations.

2.2 Multilayer Neural Networks and Recurrent Neural Networks

Definition 2.2.1 (Multilayer Neural Network (NN))**.** *Let $f : \mathbb{R}^d \to \mathbb{R}^L$ be a fully connected neural network. Denote the number of layers by K and the number of neurons in the I^{th} layer by N_I $(I \in \{0, 1, \dots K\})$. Given $\mathbf{x} \in \mathbb{R}^d$, let $\mathbf{z}^{(I)}(\mathbf{x})$ and $\mathbf{a}^{(I)}(\mathbf{x})$ denote the input and activations of the I^{th} layer of the network. Denote the weights and bias of the I^{th} layer by $\mathbf{W}^{(I)} \in R^{N_I \times N_{I-1}}$, and $\mathbf{b}^{(I)} \in R^{N_I}$ respectively. The activation function is denoted by $\sigma()$. Define $\mathbf{a}^{(0)}(\mathbf{x}) = \mathbf{x}$ and $N_0 = D$. $\mathbf{z}^{(I)}$ and $\mathbf{a}^{(I)}$ are defined as follows:*

$$\mathbf{z}^{(I)} = \mathbf{W}^{(I)}\mathbf{a}^{(I-1)} + \mathbf{b}^{(I)}, \quad \mathbf{a}^{(I)} = \sigma\left(\mathbf{z}^{(I)}\right) \tag{9}$$

The output is given by $f(\mathbf{x}) = \mathbf{a}^{(L)} = \sigma\left(\mathbf{z}^{(L)}\right)$

For sequential data, it is important to consider an architecture that takes into account the order of data. Therefore, we consider the simplest deep learning sequence model which takes into consideration previous parts of a sequence of data, defined below

Definition 2.2.2 (Recurrent Neural Network (RNN))**.** *Let $f : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^L$ be an RNN. For each time-step $t \in \mathbb{R}$ and corresponding input $\mathbf{x}^{\langle t \rangle} \in \mathbb{R}^d$, the activation, $\mathbf{a}^{\langle t \rangle}$ and output, $\mathbf{y}^{\langle t \rangle} \in \mathbb{R}^L$ are obtained from the weight matrices $\mathbf{W}_{aa}$, $\mathbf{W}_{ax} \in \mathbb{R}^{n \times m}$, $\mathbf{W}_{ya} \in \mathbb{R}^{L \times n}$ and the activation function $\sigma()$ as follows:*

$$\mathbf{a}^{\langle t \rangle} = \sigma\left(\mathbf{W_{aa}}\mathbf{a}^{\langle t-1 \rangle} + \mathbf{W_{ax}}\mathbf{x}^{\langle t \rangle} + \mathbf{b}_a\right), \quad \mathbf{y}^{\langle t \rangle} = \sigma\left(\mathbf{W_{ya}}\mathbf{a}^{\langle t \rangle} + \mathbf{b}_y\right) \tag{10}$$

Example 2.2.1 (Two Layer RNN). *We note a relationship between a general single layer RNN and a general two layer NN. In particular:*

$$\begin{aligned}
\mathbf{x} &\leftrightarrow \mathbf{x}^{\langle 1\rangle} \\
\mathbf{W}^{(1)} &\leftrightarrow \mathbf{W}_{ax} \\
\mathbf{z}^{(1)} &\leftrightarrow \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)} = \mathbf{W}_{ax}\mathbf{x}^{\langle 1\rangle} + \mathbf{W}_{aa}\mathbf{a}^{\langle 0\rangle} + \mathbf{b}_a \\
\mathbf{a}^{(1)} &= \sigma(\mathbf{z}^{(1)}) \\
\mathbf{W}^{(2)} &\leftrightarrow \mathbf{W}_{aa} \\
\mathbf{b}^{(2)} &\leftrightarrow \mathbf{b}_y \\
\mathbf{z}^{(2)} &\leftrightarrow \mathbf{W}^{(2)}\mathbf{a}^{(1)} + \mathbf{b}^{(2)} = \mathbf{W}_{aa}\mathbf{a}^{(1)} + \mathbf{b}_y \\
\mathbf{a}^{(2)} &= \sigma(\mathbf{z}^{(2)}) \leftrightarrow \mathbf{y}^{\langle 1\rangle}
\end{aligned} \tag{11}$$

Therefore, single layer RNNs inherit the properties of NNs. Properties of NNs can be applied to a single layer RNN (or a general RNN considered at any given timestep with known variable bias $\mathbf{a}^{\langle t-1\rangle}$).

2.3 Lipschitz Continuity and Robustness

Definition 2.3.1 (Global Lipschitz Continuity). *The function $f : \mathbb{R}^d \to \mathbb{R}^L$ is globally Lipschitz continuous if there exists an $\mathcal{L} > 0$ s.t:*

$$\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d, \quad \|f(\mathbf{x}) - f(\mathbf{y})\| \leqslant \mathcal{L}\|\mathbf{x} - \mathbf{y}\| \tag{12}$$

The Lipschitz constant of f, $\mathcal{L}(f)$ is defined as the smallest $\mathcal{L} > 0$ such that (12) holds. In other words,

$$\mathcal{L}(f) \coloneqq \inf_{\mathcal{L}>0} \{\mathcal{L} \geqslant \frac{\|f(\mathbf{x}) - f(\mathbf{y})\|}{\|\mathbf{x} - \mathbf{y}\|}\} \tag{13}$$

Definition 2.3.2 (Local Lipschitz Continuity). *The function $f : \mathbb{R}^d \to \mathbb{R}^L$ is locally Lipschitz continuous if given any neighbourhood $U \subseteq \mathbb{R}^d$, the restriction of f to the neighbourhood U, $f_U : U \to \mathbb{R}^L$ is Lipschitz continuous.*

The Lipschitz constant, $\mathcal{L}$ is intricately linked to the derivative. The Lipschitz constant allows us to view derivatives (local property) in a more global setting.

Lemma 2.3.1. *The function $f : \mathbb{R}^d \to \mathbb{R}^L$, $f \in C^1\left(\mathbb{R}^d\right)$ is Lipschitz continuous with Lipschitz constant $\mathcal{L}$ if and only if for every $\mathbf{x} \in \mathbb{R}^d$, $\|\nabla f(\mathbf{x})\| \leqslant \mathcal{L}$*

Proof. Assume f is Lipschitz continuous and let $\mathbf{x} \in \mathbb{R}^d$ be a point with $\nabla f(\mathbf{x}) \neq 0$. Choose $\boldsymbol{h} \in \mathbb{R}^d$ with $\|\boldsymbol{h}\| = 1$ such that

$$\frac{|f(\mathbf{x} + t\boldsymbol{h}) - f(\mathbf{x})|}{t} \leqslant \mathcal{L} \tag{14}$$

holds for t sufficiently small. In particular, this holds in the limit as $t \to 0$, and hence

$$|\langle \nabla f(\mathbf{x}), \boldsymbol{h} \rangle| = \lim_{t \to 0} \frac{|f(\mathbf{x} + t\boldsymbol{h}) - f(\mathbf{x})|}{t} \leqslant \mathcal{L} \tag{15}$$

This holds, in particular, with the choice $\boldsymbol{h} = \nabla f(\mathbf{x})/\|\nabla f(\mathbf{x})\|$. Therefore, the norm of the gradient is bounded. Conversely, assume that $\|\nabla f(\mathbf{x},)\| \leqslant \mathcal{L}$. Then

$$|f(\mathbf{x}+\boldsymbol{h}) - f(\mathbf{x})| \leqslant \int_0^1 \left| \frac{d}{dt} f(\mathbf{x} + t\boldsymbol{h}) \right| dt = \int_0^1 |\langle \nabla f(\mathbf{x}+t\boldsymbol{h}), \boldsymbol{h} \rangle| dt \leqslant \int_0^1 \|\nabla f(\mathbf{x}+t\boldsymbol{h})\| \cdot \|\boldsymbol{h}\| dt \ , \tag{16}$$

where we used the Cauchy-Schwartz inequality. We can now bound the integrand by the Lipschitz constant $\mathcal{L}$, and the claim follows. □

Lemma 2.3.2. *The function $f : \mathbb{R}^d \to \mathbb{R}^L$, $f \in C^2\left(\mathbb{R}^d\right)$ has Lipschitz continuous gradient with Lipschitz constant $\mathcal{L}$*

$$\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d, \quad \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leqslant \mathcal{L} \cdot \|\mathbf{x} - \mathbf{y}\| \tag{17}$$

if and only if at every point $\mathbf{x} \in \mathbb{R}^d$, $\|\nabla^2 f(\mathbf{x})\| \leqslant \mathcal{L}$ where the norm is the operator norm.

In [3], a formal guarantee for the robustness of a classifier at a given point $\mathbf{x}$ is derived when F is Lipschitz continuous. Note we modify the statement to be consitent with the notation presented here. From the above analysis, this is identical to the condition that $\nabla F(\mathbf{x})$ is bounded.

Theorem 2.3.1. *Let $f : \mathbb{R}^d \to \mathbb{R}^L$, $f \in C^1\left(\mathbb{R}^d\right)$ be multi-class classifier. Given $\mathbf{x} \in \mathbb{R}^d$, suppose f is locally Lipschitz in $B(\mathbf{x}, R) = \left\{\mathbf{y} \in \mathbb{R}^d \mid \|\mathbf{x} - \mathbf{y}\|_p \leqslant R\right\}$. Then $\|F(\mathbf{x})\|$ is bounded on $B(\mathbf{x}, R)$. For all $\boldsymbol{\delta} \in \mathbb{R}^d$ s.t*

$$\|\boldsymbol{\delta}\| \leqslant \max_{R>0} \min \left\{ \frac{F(\mathbf{y})}{\max_{\mathbf{y} \in B(\mathbf{x},R)} \|\nabla F((\mathbf{y}))\|}, R \right\} := \alpha \tag{18}$$

it holds that $\hat{k}(\mathbf{x}) = \hat{k}(\mathbf{x} + \boldsymbol{\delta})$. The classifier decision does not change on $B(\mathbf{x}, \alpha)$.

Provided F is Lipschitz continuous in a neighbourhood of $\mathbf{x}$, $B(\mathbf{x}, R)$ (implying ∇F is bounded on the same neighbourhood), the above theorem gives an upper bound on $\delta_{adv}(\mathbf{x}; f)$. This motivates analysis of robustness based on the curvature and Lipschitz constant of the decision function. However, it turns out that even for a single layer RNN, the computation of the Lipschitz constant is NP-hard. Consider a general multi-class RNN with no output activation and no bias with predictions given by:

$$\mathbf{y}^{\langle t \rangle} = \mathbf{W_{ya}} \sigma \left(\mathbf{W_{aa}} \mathbf{a}^{\langle t-1 \rangle} + \mathbf{W_{ax}} \mathbf{x}^{\langle t \rangle} \right) \tag{19}$$

where $\mathbf{x}^{\langle t \rangle}, \mathbf{a}^{\langle t \rangle} \in \mathbb{R}^{d \times m}$, $\mathbf{W_{ax}}, \mathbf{W_{aa}} \in \mathbb{R}^{n \times m}$, and $\mathbf{W_{ya}} \in \mathbb{R}^{L \times n}$.

Theorem 2.3.2 (Lipschitz constant computation is NP hard)**.** *The computation of the Lipschitz constant of the single layer RNN above is NP hard.*

Proof. We extent the proof given by [6] for a single layer NN, noting the relationship between RNNs and NNs in example (2.2.1) □

Therefore, we move our attention to other measures of curvature, mainly the Hessian of the decision function F. Theoretical justification of this is provided in later sections.

2.4 Random Matrix Theory

Random matrix theory (RMT) [8, 2, 9, 19, 18] studies the properties of high-dimensional matrices $M = [X_{ij}]_{i,j=1}^{n} \in \mathbb{R}^{n \times n}$ where the entries X_{ij} for $1 \leqslant i, j \leqslant n$ are sampled according to fixed probability distributions. The properties studied by RMT include the eigenvalue and sigular value distribution in the limiting case, that is as $n \to +\infty$. In the preamble below, we introduce two classical random matrix ensembles, real Wigner matrices (2.4.1) and real Wishart matrices (2.4.2). Next we state without proof Wigner's semicircle law (2.4.4) and the Marchenko-Pastur law (2.4.5) for the limiting spectral (eigenvalue) distribution of symmetric (Wigner) matrices and sample covariance (Wishart) matrices respectively.

Following this, we briefly introduce important analytical tools in RMT, namely the Stieltjes transform (2.4.6) and the $\mathcal{R}$-transform (2.4.8). In general, the asymptotic behaviour of random matrices is independent of the distribution of the entries. Any matrix with identically independently distributed (i.i.d.) entries with finite mean and standard deviations are expected to behave similarly. This property is termed universality.

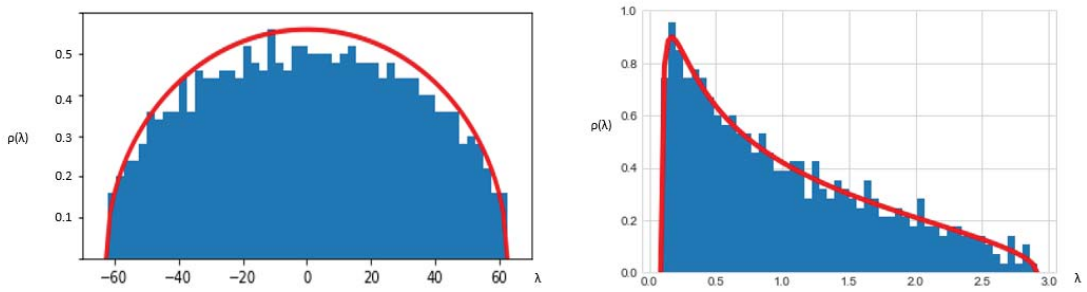

Figure 4: The spectral distribution for random 1000 x 1000 Wigner (left) and Wishart (right) matrix with entries distributed as $\sim \mathcal{N}(0,1)$. The blue histograms show the empirical results while the red curves are the theoretical predictions.

From a practical standpoint, universality allows us to apply well known results to a wide range of matrix ensembles. This is a property we will exploit further on.

Definition 2.4.1 (Real Wigner Matrices)**.** *For $n \in \mathbb{N}$, $\boldsymbol{M} = [X_{ij}]_{i,j=1}^{n} \in \mathbb{R}^{n\times n}$ is a real Wigner matrix if for all $1 \leqslant i < j \leqslant n$, X_{ij} are i.i.d random variables with fixed variance σ^2, $X_{ij} = X_{ji}$, and the diagonal entries M_{ii} for $i \in \{1,\dots,N\}$ are i.i.d random variables with variance σ^2 (with possibly different different distribution to the non-diagonal entries) with variance σ^2.*

A real Wigner Matrix, $\boldsymbol{M}$ is symmetric, that is $\boldsymbol{M} = \boldsymbol{M}^T$. The diagonal and non-diagonal entries of a real Wigner matrix are allowed to have different distributions but must have the same variance σ^2. We note that we can only study square matrices within the framework of Wigner matrices. This leads us to consider the following ensemble of matrices, the Wishart matrices.

Definition 2.4.2 (Real Wishart Matrix)**.** *For $n, p \in \mathbb{N}$, let $\boldsymbol{X} = [X_{ij}]_{i,j=1}^{n,p} \in \mathbb{R}^{n\times p}$ be a (rectangular) matrix such that for all $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant p$, X_{ij} are (i.i.d) random variables with (scaled) variance σ^2/p The matrix $\boldsymbol{M} \coloneqq \boldsymbol{X}\boldsymbol{X}^T$ is a real Wishart matrix.*

We will restrict our analysis of random matrices to the distribution of eigenvalues of a high dimensional random matrix. For a matrix $\boldsymbol{M}_n \in \mathbb{R}^{n\times n}$, with random entries, we study the limit of the empirical spectral density, defined below:

Definition 2.4.3 (Empirical Spectral Density)**.** *For a random matrix $\boldsymbol{M}_n \in \mathbb{R}^{n\times n}$, let $\lambda_i(\boldsymbol{M}_n)$, $i = \{1,\dots,n\}$ denote the set eigenvalues of $\boldsymbol{M}_n$, including multiplicities and let*

$\delta(x)$ denote the Dirac delta function centred at x. The empirical spectral density of $\boldsymbol{M}_n$ is defined as

$$\rho_{\boldsymbol{M}_n}(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\delta\left(\lambda - \lambda_i\left(\boldsymbol{M}_n\right)\right) \tag{20}$$

The limiting spectral density is given by the limit of $\rho_{\boldsymbol{M}_n}(\lambda)$ as $n \to \infty$, if it exists.

The distribution of the limiting spectral density of Wigner and Wishart matrices are well known results in RMT, shown below.

Definition 2.4.4 (Semi-Circle Distribution)**.** *Let $\{\boldsymbol{M}_n\}_{n\in\mathbb{N}} \subseteq \mathbb{R}^{n\times n}$ be a sequence of Wigner matrices with entries M_{ij} sampled from a probability distribution with fixed variance σ^2. The limiting spectral density $\rho_{\mathrm{SC}}(\lambda;\sigma) = \lim_{n\to+\infty}\rho_{\boldsymbol{M}_n}(\lambda)$ is given by the semi-circle law defined as*

$$\rho_{\mathrm{sc}}(\lambda;\sigma) = \begin{cases} \frac{1}{2\pi\sigma^2}\sqrt{4\sigma^2-\lambda^2} & \text{if } |\lambda| \leqslant 2\sigma \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

Note the limiting semi-circle distribution of Wigner matrices is symmetric with the average and most common eigenvalue being zero.

Definition 2.4.5 (Marchenko-Pastur Distribution)**.** *Let $\{\boldsymbol{M}_n = (\boldsymbol{X}\boldsymbol{X}^T)_n\}_{n\in\mathbb{N}} \subseteq \mathbb{R}^{n\times n}$ with $\boldsymbol{X}_n \in \mathbb{R}^{n\times p}$ be a sequence of Wigner matrices with entries M_{ij} sampled from a probability distribution with fixed variance σ^2/p. Let $\phi \coloneqq n/p$ denote the size of the problem. The limiting spectral density $\rho_{\mathrm{mp}}(\lambda;\sigma,\phi) = \lim_{n\to+\infty}\rho_{\boldsymbol{M}_n}(\lambda)$ is given by the Marchenko-Pastur distribution defined as*

$$\rho_{\mathrm{MP}}(\lambda;\sigma,\phi) = \begin{cases} \rho(\lambda) & \text{if } \phi < 1 \\ \left(1-\phi^{-1}\right)\delta(\lambda) + \rho(\lambda) & \text{otherwise} \end{cases} \tag{22}$$

$$\begin{aligned} \rho(\lambda) &= \frac{1}{2\pi\lambda\sigma\phi}\sqrt{\left(\lambda-\lambda_-\right)\left(\lambda_+-\lambda\right)} \\ \lambda_\pm &= \sigma(1\pm\sqrt{\phi})^2 \end{aligned} \tag{23}$$

We now introduce analytical tools to study eigenvalue distributions. The first such tool is a transform that is analogous to the Fourier transform or characteristic function of a distribution.

Definition 2.4.6 (The Stieltjes Transform)**.** *the Stieltjes transformation of a distribution $\rho : I \subseteq \mathbb{R} \to \mathbb{R}$, $G : z \in \mathbb{C}\backslash I \to \mathbb{C}$ is a function of the complex variable z on $\mathbb{C}\backslash I$*

$$G(z) = \int_{\mathbb{R}} \frac{\rho(t)}{z-t} dt \tag{24}$$

Under certain conditions we can recover the distribution ρ after application of the Stieltjes transform. Given the Stieltjes transformation $G(z)$ of ρ we apply the inverse formula of Stieltjes-Perron to $G(z)$ to recover ρ.

Definition 2.4.7 (Inverse Stieltjes Transform)**.** *Given the the Stieltjes transform, G of a probability distribution ρ continuous throughout I, we can recontruct the density ρ by the Steiltjes-Perron Inversion formula*

$$\rho(\lambda) = -\frac{1}{\pi} \lim_{\epsilon \to 0^+} \operatorname{Im} G(\lambda + i\epsilon) \tag{25}$$

Definition 2.4.8 (The $\mathcal{R}$ Transform)**.** *Given the Stieltjes transform G of a probability distribution ρ, the $\mathcal{R}$ transform is defined by the functional equation*

$$R(G(z)) + \frac{1}{G(z)} = z \tag{26}$$

Note for $\alpha \in \mathbb{R}$, $\mathcal{R}_{(\alpha G)} = \mathcal{R}_G$. In addition, if A and B are (freely) independent, the $\mathcal{R}$ transform linearizes convolution$\mathcal{R}_{A+B} = \mathcal{R}_A + \mathcal{R}_B$. Note we use the results of these transforms without proof.

3 Curvature and Robustness

In the following, we argue that classifiers with less curved decision boundaries are more robust. Note that this is in agreement with existing theoretical results [25].

Suppose point $\mathbf{x} \in \mathbb{R}^d$ is predicted to belong to class $\hat{k}(\mathbf{x}) = \arg\max_k f_k(\mathbf{x})$ with confidence margin $F(\mathbf{x}) = f_{\hat{k}(\mathbf{x})}(\mathbf{x}) - \max_{k \neq \hat{k}(\mathbf{x})} f_k(\mathbf{x}) = t \geqslant 0$. Let $\alpha \geqslant 0$ be the maximal perturbation radius such that $\forall \boldsymbol{\delta} \in \left\{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x}\| \leqslant \alpha\right\}$,

$$|F(\mathbf{x} + \boldsymbol{\delta}) - F(\mathbf{x})| < t \tag{27}$$

Therefore, $F(\mathbf{x}+\boldsymbol{\delta}) \in (0, 2t)$. Given a trajectory from $\mathbf{x}$ to $\mathbf{x}+\boldsymbol{\delta}$, the decrease in confidence of the most probable class $\hat{k}(\mathbf{x})$ is not more than the increase in confidence of the second

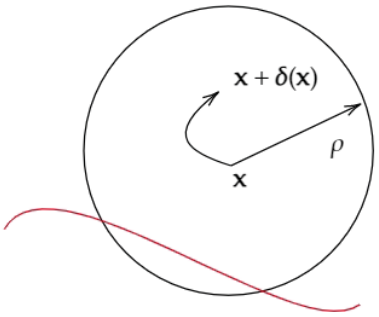

Figure 5: A perturbation radius ρ, s.t if $\|\boldsymbol{\delta}\| \leqslant \rho$, $|F(\mathbf{x}+\boldsymbol{\delta}) - F(\mathbf{x})| < t$. Note the decision boundary B (red) is allowed to intersect the ball in this setting (as long as (27) holds).

most probable class $\max_{k \neq \hat{k}(\mathbf{x})} f_k(\mathbf{x})$. Therefore, the classifier predicts the same class for both points $\mathbf{x}$ and $\mathbf{x}+\boldsymbol{\delta}$.

Let F be second-order differentiable with second order taylor series expansion (w.r.t inputs):

$$F(\mathbf{x}+\boldsymbol{\delta}) \approx F(\mathbf{x}) + \boldsymbol{J}\cdot\boldsymbol{\delta} + \frac{1}{2}\boldsymbol{\delta}^T\boldsymbol{H}\boldsymbol{\delta} \tag{28}$$

Let us now consider minimal perturbations which result in either misclassification or correct classifications with confidence $F(\mathbf{x}) = f_{\hat{k}(\mathbf{x})}(\mathbf{x}) - \max_{k \neq \hat{k}(\mathbf{x})} f_k(\mathbf{x}) < t$. That is, we consider perturbations $\boldsymbol{\delta}$ s.t $|F(\mathbf{x}+\boldsymbol{\delta}) - F(\mathbf{x})| \geqslant t$. Expressing this inequality with the Taylor expansion (28) gives

$$\left|\boldsymbol{J}\cdot\boldsymbol{\delta} + \frac{1}{2}\boldsymbol{\delta}^T\boldsymbol{H}\boldsymbol{\delta}\right| \geqslant t \tag{29}$$

$\boldsymbol{H}$ is symmetric, therefore, $\boldsymbol{H}$ can be diagonalised as $\boldsymbol{H} = \boldsymbol{E}\boldsymbol{\Lambda}\boldsymbol{E}^T$. where $\boldsymbol{E} = [\boldsymbol{e}_0, \cdots, \boldsymbol{e}_n]$ a matrix of eigenvectors of $\boldsymbol{H}$, $[\boldsymbol{e}_0, \cdots, \boldsymbol{e}_n]$ and $\boldsymbol{\Lambda}$ is the diagonal eigenvalue matrix. Let $\boldsymbol{y} = \boldsymbol{E}^T\boldsymbol{\delta}$. Then

$$\boldsymbol{\delta}^T\boldsymbol{H}\boldsymbol{\delta} = \boldsymbol{\delta}^T E\boldsymbol{\Lambda}\boldsymbol{E}^T\boldsymbol{\delta} = \left(\boldsymbol{E}^T\boldsymbol{\delta}\right)^T\boldsymbol{\Lambda}\left(\boldsymbol{E}^T\boldsymbol{\delta}\right) = \boldsymbol{y}^T\boldsymbol{\Lambda}\boldsymbol{y} \tag{30}$$

Therefore, we have that the following bound holds

$$\left|\boldsymbol{J}\cdot\boldsymbol{\delta} + \frac{1}{2}\boldsymbol{\delta}^T\boldsymbol{H}\boldsymbol{\delta}\right| \leqslant |\boldsymbol{J}\cdot\boldsymbol{\delta}| + \frac{1}{2}\left|\boldsymbol{y}^T\boldsymbol{\Lambda}\boldsymbol{y}\right| \leqslant \sum_{i=1}^{n}|J_i|\cdot|\delta_i| + \frac{1}{2}\sum_{i=1}^{n}|\lambda_i|\cdot\left|y_i^2\right| \tag{31}$$

Therefore, the bound (29) can be expressed as follows:

$$\sum_{i=1}^{n} |J_i| \cdot |\delta_i| + \frac{1}{2} \sum_{i=1}^{n} |\lambda_i| \cdot |y_i^2| \geqslant \left| \boldsymbol{J} \cdot \boldsymbol{\delta} + \frac{1}{2} \boldsymbol{\delta}^T \boldsymbol{H} \boldsymbol{\delta} \right| \geqslant t \tag{32}$$

If the magnitude of every element of $\boldsymbol{J}$, J_i and eigenvalue λ_i of $\boldsymbol{H}$ is large enough, the classifier will either misclassify the point $\mathbf{x}$ or make a correct prediction but with low confidence. That is, the more curved the decision boundary, the more vulnerable the classifier is to adversarial perturbations.

Let's continue looking at minimal perturbations which result in either correct classifications with confidence $F(\mathbf{x}) = \hat{k}(\mathbf{x}) - \max_{k \neq \hat{k}(\mathbf{x})} f_k(\mathbf{x}) < t$ or misclassification. Define $\boldsymbol{\delta}^*$ such that

$$\boldsymbol{\delta}^* := \arg\min_{\boldsymbol{\delta}} \|\boldsymbol{\delta}\| \text{ s.t. } |F(\mathbf{x} + \boldsymbol{\delta}) - F(\mathbf{x})| \approx \left| \boldsymbol{J} \cdot \boldsymbol{\delta} + \frac{1}{2} \boldsymbol{\delta}^T \boldsymbol{H} \boldsymbol{\delta} \right| \geqslant t \tag{33}$$

Note this definition of $\boldsymbol{\delta}^*$ is distinct to the definition of adversarial perturbation, $\boldsymbol{\delta_{adv}}(\mathbf{x}, f)$ (2.1.5). This is because we are considering the decision boundary with confidence $B'(t)$ rather than the true decision boundary B.

We bound $\|\boldsymbol{\delta}^*\|$ as a function of curvature of the decision boundary. In particular, these bounds are a function of the Jacobian magnitude $\|\boldsymbol{J}\|$ and the maximal Hessian eigenvalue, $\nu := \lambda_{\max}(\mathbf{H})$ of the decision function F (taken w.r.t inputs, as before.) Note we follow a similar argument provided in [15] applied to the loss function.

Theorem 3.0.1 (Upper and Lower bounds for Robustness)**.** *Let* $\mathbf{x}$ *be such that* $F(\mathbf{x}) = t \geqslant 0$*, and let* $\boldsymbol{J} = \nabla F(\mathbf{x})$*. Assume* $\nu := \lambda_{\max}(\mathbf{H}) \geqslant 0$*, and let* $\boldsymbol{u}$ *be the eigenvector corresponding to* ν*. That is,* $\mathbf{H}\boldsymbol{u} = \nu \boldsymbol{u}$*. Then, we have*

$$\frac{\|\boldsymbol{J}\|}{\nu} \left(\sqrt{1 + \frac{2\nu t}{\|\boldsymbol{J}\|^2}} - 1 \right) \leqslant \|\boldsymbol{\delta}^*\| \leqslant \frac{|\boldsymbol{J}^T u|}{\nu} \left(\sqrt{1 + \frac{2\nu t}{(\boldsymbol{J}^T \boldsymbol{u})^2}} - 1 \right) \tag{34}$$

Proof. Lower bound. Let $\alpha := \|\boldsymbol{\delta}^*\|$. We note that from the Taylor expansion (28), we have that α satisfies:

$$-t + \|\boldsymbol{J}\|\alpha + \frac{\nu}{2}\alpha^2 \geqslant -t + \boldsymbol{J}^T \boldsymbol{\delta}^* + \frac{1}{2} (\boldsymbol{\delta}^*)^T \boldsymbol{H} \boldsymbol{\delta}^* \geqslant 0$$

Solving the above second-order inequality, and noting only the inequality for which $\alpha \geqslant 0$

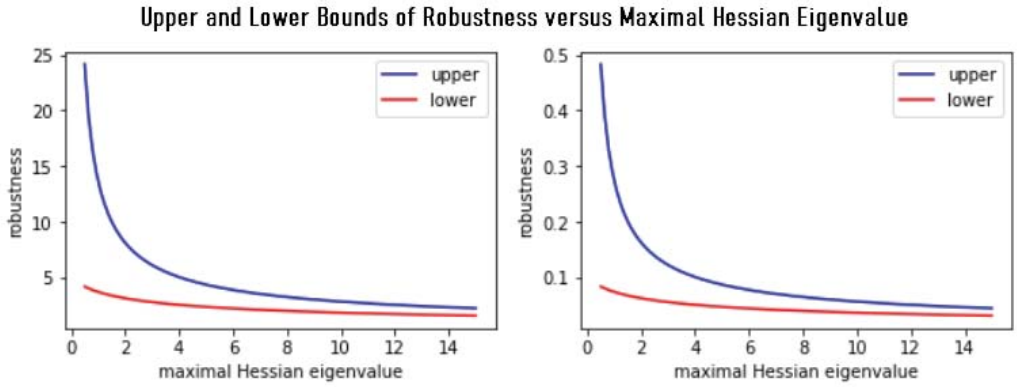

Figure 6: The above is a plot of the upper and lower bounds versus maximal Hessian eigenvalue. In both we fix t = 0.5. Note we take $\|\boldsymbol{J}\| = 0.1$ (left) and $\|\boldsymbol{J}\| = 5$ (right). The above plots show the bounds are not significantly affected by the Jacobian magnitude. Only the maximal Hessian eigenvalue matterss. This motivates us to investigate the Hessian further and look into Hessian regularisation later.

holds, we get

$$\alpha \geqslant \frac{\|\boldsymbol{J}\|}{\nu}\left(\sqrt{1+\frac{2\nu t}{\|\boldsymbol{J}\|^2}}-1\right)$$

Upper bound: Let $\alpha \geqslant 0$. Define $\boldsymbol{\delta} := \alpha \boldsymbol{u}$, and let us find the minimal α such that

$$-t + \boldsymbol{J}^T\boldsymbol{\delta} + \frac{1}{2}\boldsymbol{\delta}^T\boldsymbol{H}\boldsymbol{\delta} = -t + \alpha\boldsymbol{J}^T\boldsymbol{u} + \frac{\alpha^2\nu}{2} \geqslant 0$$

Given any $\alpha \geqslant \alpha_{\min} \coloneqq \alpha_{\min} = \frac{|\boldsymbol{J}^T\boldsymbol{u}|}{\nu}\left(\sqrt{1+\frac{2\nu t}{(\boldsymbol{J}^T\boldsymbol{u})^2}}-1\right)$, the above inequality holds true. Hence, we have that $\|\boldsymbol{\delta}^*\| \leqslant |\alpha_{\min}|$, which concludes the proof of the upper bound. □

Note that upper and lower bounds on the robustness decrease with increasing curvature. In other words, under the second order approximation, small curvature is beneficial to obtain classifiers with higher robustness.

4 Probabilistic Bounds on Adversarial Distance

Given a data point $\mathbf{x} \in \mathbb{R}^d$, we aim to characterise the probability of misclassification of points on the ball of radius ρ centred at $\mathbf{x}$. This probability is proportional to the fraction of misclassified points on the ball. This viewpoint is useful for experimental validation with Monte-Carlo simulations (1). In this section we shall focus on theoretic bounds, and

experimentally validate them later. This section closely follows [14] which will be a useful reference.

We formalise this idea as follows. Given a fixed probability of misclassification, $\delta \in [0, 1]$, we seek a radius of perturbation $\rho \geqslant \delta_{adv}(\mathbf{x}; f)$ of minimal length such that $\underset{\boldsymbol{\delta} \in \mathbb{S}}{\mathbb{P}}(\hat{k}(\boldsymbol{x} + \rho\boldsymbol{\delta}) \neq \hat{k}(\boldsymbol{x})) \leqslant \delta$. We aim to find the perturbation radius of minimal distance such that the probability of misclassification with this perturbation is at most δ.

Recall the second order Taylor expansion of the decision function $F(\mathbf{x} + \boldsymbol{v})$ about $\mathbf{z} = \mathbf{x} + \boldsymbol{\delta}(\mathbf{x})$ implies

$$F(\mathbf{x} + \boldsymbol{v}) \approx (\mathbf{v} - \boldsymbol{\delta}(\mathbf{x}))^T \nabla F(\mathbf{z}) + \frac{1}{2}(\boldsymbol{v} - \boldsymbol{\delta}(\mathbf{x}))^T (\mathbf{H_z})(\boldsymbol{v} - \boldsymbol{\delta}(\mathbf{x})) \tag{35}$$

Note that if $F(\mathbf{x} + \boldsymbol{v}) < 0$, $\hat{k}(\mathbf{x} + \boldsymbol{v}) \neq \hat{k}(\mathbf{x})$ (according to the second order approximation). Therefore, $(\mathbf{v} - \boldsymbol{\delta}(\mathbf{x}))^T \nabla F(\mathbf{z}) + \frac{1}{2}(\boldsymbol{v} - \boldsymbol{\delta}(\mathbf{x}))^T (\mathbf{H_z})(\boldsymbol{v} - \boldsymbol{\delta}(\mathbf{x})) < 0 \implies \hat{k}(\mathbf{x} + \boldsymbol{v}) \neq \hat{k}(\mathbf{x})$. Note it should be clear that this is only true in the second order approximation. In reality, higher order terms in the Taylor expansion interfere. In practice, the assumption holds well if we consider small perturbation radii only (refer to the figure above (4)). We assume that across the domain, for perturbations of sufficiently small radius ρ, a negative second order Taylor approximation results in misclassification. We formalise this assumption below, termed the second order decision boundary model, $\mathcal{L}(\mathbf{x}, \rho)$ taking inspiration from [14]. Note we sample $\mathbf{x}$ from the domain according to the measure μ ($\mathbf{x} \sim \mu$). We refer to the ball of radius ρ as $B(\rho)$

$$\begin{aligned} &\mathcal{L}(\mathbf{x}, \rho) : \exists \rho > 0 \text{ s.t } \forall \boldsymbol{v} \in B(\rho), \\ &\qquad \alpha_{\mathbf{x}}^{-1} (\boldsymbol{v} - \boldsymbol{\delta}(\mathbf{x}))^T \boldsymbol{H_z} (\boldsymbol{v} - \boldsymbol{\delta}(\mathbf{x})) + \boldsymbol{\delta}(\mathbf{x})^T (\boldsymbol{v} - \boldsymbol{\delta}(\mathbf{x})) \leqslant 0 \\ &\qquad \implies \hat{k}(\mathbf{x} + \boldsymbol{v}) \neq \hat{k}(\mathbf{x}) \\ &\qquad \text{holds for almost all } \mathbf{x} \sim \mu \end{aligned} \tag{36}$$

In addition, we make a further second assumption. We shall work with the decision function Hessian $\boldsymbol{H_z}$ restricted to $\text{span}(\boldsymbol{\delta}(\mathbf{x}), \boldsymbol{v})$, where we uniformly sample a random vector $\boldsymbol{v} \in B(\rho)$ centered at $\mathbf{x} + \boldsymbol{\delta}(\mathbf{x})$. Any probability discussed below is w.r.t uniformly sampling $\boldsymbol{v}$. We assume that given any curvature $\kappa > 0$, there exists a subspace of the domain $\mathcal{S}$ s.t the probability of the (normalised) quadratic form of $\boldsymbol{H_z}$ restricted to $\text{span}(\boldsymbol{\delta}(\mathbf{x}), \boldsymbol{v})$ being greater than κ is bounded below by $1 - \beta$. This quadratic form is defined on a two dimensional subspace, described formally below. Note [14] experimentally verify this

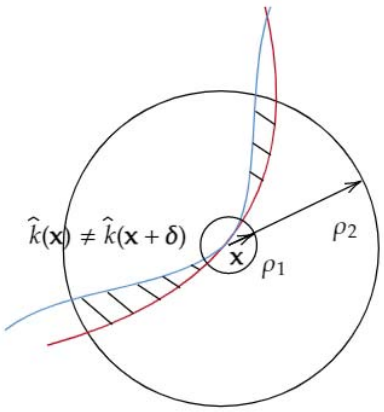

Figure 7: The above figure shows the decision boundary (blue) and the second order approximation (red). Note that Assumption 1 holds well in the case of small radius ρ_1 but it is clear that it does not hold for any perturbation distance in general. The points in between the red and blue curves clearly violate the assumption.

assumption for deep NNs. Therefore, we will take this assumption without scrutiny and further build upon it. We formalise this assumption as follows:

Given $\kappa > 0$, assume there exists $\beta > 0$, and $\mathcal{S} \subseteq \mathbb{R}^d$, an m dimensional subspace such that

$$\mathop{\mathbb{P}}_{\boldsymbol{v}\in\mathbb{S}}\left(\forall \boldsymbol{u} \in \mathbb{R}^2, \alpha_{\mathbf{x}}^{-1}\boldsymbol{u}^T \boldsymbol{H}_{\mathbf{z}}^{\boldsymbol{\delta}(\mathbf{x}),\boldsymbol{v}}\boldsymbol{u} \geqslant \kappa\|\boldsymbol{u}\|_2^2\right) \geqslant 1-\beta \text{ for almost all } \mathbf{x} \sim \mu. \tag{37}$$

$H_{\mathbf{z}}^{\boldsymbol{\delta}(\mathbf{x}),\boldsymbol{v}} = \boldsymbol{E}^T \boldsymbol{H}_{\mathbf{z}} \boldsymbol{E}$ where $\boldsymbol{E}$ is an orthonormal basis for $\text{span}(\boldsymbol{\delta}(\mathbf{x}), \boldsymbol{v})$ and $\mathbb{S}$ denotes the unit sphere in $\mathcal{S}$. In the following theorem, assuming (36) and (37), we find a bound on the probability of misclassification over a subspace $\mathcal{S}$ of the domain. Then, we extend this to the entire domain $\mathbb{R}^d$ by simple application of the Semi-Circle distribution (2.4.4) from RMT.

Theorem 4.0.1 (Probabilistic Bounds on Adversarial Distance)**.** *Assume both Assumption 1 (36) and Assumption 2 (37) hold. Consider a multi-class classifier $f : \mathbb{R}^d \to \mathbb{R}^L$. Let $\kappa > 0$, $\delta > 0$ and $d \geqslant m \in \mathbb{N}$, where m is the dimension of the subspace defined in (37). The probability of misclassification on the sphere of radius ρ satisfying (36) for a point $\mathbf{x} \sim \mu$ is bounded above as follows:*

$$\mathop{\mathbb{P}}_{\boldsymbol{v}\sim\rho\mathbb{S}}\left(\hat{k}(\mathbf{x}+\boldsymbol{v}) \neq \hat{k}(\mathbf{x})\right) \leqslant 2\exp\left(-\frac{m(\kappa\rho^2-1)^2}{2\rho^2(1-2\kappa)^2}\right) + \beta \tag{38}$$

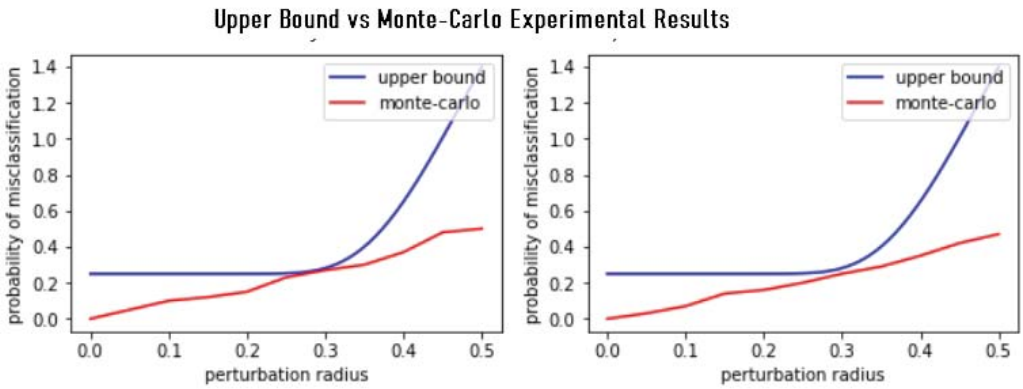

Figure 8: Monte-Carlo simulations (1) on a 1000-dimensional classifier with quadratic decision boundary $\sim \|\mathbf{x}\|^2$. The above graphs show our bounds (4.0.1) hold for a high dimensional quadratic classifier. Further experiments should be conducted to investigate the validity of the bound for more general decision boundaries.

Proof. Let $\mathbf{x} \sim \mu$. We have

$$\begin{aligned}
&\mathop{\mathbb{P}}_{\boldsymbol{v}\sim\rho\mathbb{S}}\left(\hat{k}(\mathbf{x}+\boldsymbol{v}) \neq \hat{k}(\mathbf{x})\right) \\
&\approx \mathop{\mathbb{P}}_{\boldsymbol{v}\sim\rho\mathbb{S}}\left(\alpha_{\mathbf{x}}^{-1}(\boldsymbol{v}-\boldsymbol{\delta}(\mathbf{x}))^T\boldsymbol{H}_{\boldsymbol{z}}(\boldsymbol{v}-\boldsymbol{\delta}(\mathbf{x})) + \boldsymbol{\delta}(\mathbf{x})^T(\boldsymbol{v}-\boldsymbol{\delta}(\mathbf{x})) \leqslant 0\right)
\end{aligned}$$

By the quadratic decision boundary model assumption, $\mathcal{L}(\boldsymbol{x}, \rho)$ we have that

$$\begin{aligned}
&\mathop{\mathbb{P}}_{\boldsymbol{v}\sim S}\left(\alpha_{\mathbf{x}}^{-1}(\rho\boldsymbol{v}-\boldsymbol{\delta}(\mathbf{x}))^T\boldsymbol{H}_{\boldsymbol{z}}(\rho\boldsymbol{v}-\boldsymbol{\delta}(\mathbf{x})) + (\boldsymbol{\delta}(\mathbf{x}))^T(\rho\boldsymbol{v}-\boldsymbol{\delta}(\mathbf{x})) \leqslant 0\right) \\
&\leqslant \mathop{\mathbb{P}}_{\boldsymbol{v}\sim \mathrm{S}}\left(\kappa\|\rho\boldsymbol{v}-\boldsymbol{\delta}(\mathbf{x})\|_2^2 + (\boldsymbol{\delta}(\mathbf{x}))^T(\rho\boldsymbol{v}-\boldsymbol{\delta}(\mathbf{x})) \leqslant 0\right) + \beta \\
&\leqslant \mathop{\mathbb{P}}_{\boldsymbol{v}\sim \mathrm{S}}\left(\rho(1-2\kappa)\boldsymbol{v}^T\boldsymbol{\delta}(\mathbf{x}) + \kappa\rho^2 + (\kappa-1) \leqslant 0\right) + \beta \\
&\leqslant \mathop{\mathbb{P}}_{\boldsymbol{v}\sim \mathrm{S}}\left(\rho(1-2\kappa)\boldsymbol{v}^T\boldsymbol{\delta}(\mathbf{x}) \leqslant \epsilon\right) + \mathop{\mathbb{P}}_{\boldsymbol{v}\sim \mathrm{S}}\left(\kappa\rho^2 + (\kappa-1) \leqslant \epsilon\right) + \beta,
\end{aligned}$$

Note we may choose an arbitrary $\epsilon > 0$ to simplify the above expression. Let $\epsilon = \kappa\rho^2 - 1 \implies \rho^2 = (\epsilon+1)/\kappa$. By construction, $\kappa\rho^2 + (\kappa-1) \geqslant \epsilon$. Therefore, the term $\mathop{\mathbb{P}}_{\boldsymbol{v}\sim \mathrm{S}}(\kappa\rho^2 + (\kappa-1) \leqslant \epsilon)$ vanishes. Using the concentration of measure on the sphere [13], we have

$$\mathop{\mathbb{P}}_{\mathbf{v}\sim\mathbf{S}}\left(\boldsymbol{v}^T\boldsymbol{\delta}(\mathbf{x}) \leqslant \frac{-\epsilon}{\rho(1-2\kappa)}\right) = \mathop{\mathbb{P}}_{\mathbf{v}\sim\mathbf{S}}\left(\boldsymbol{v}^T\boldsymbol{\delta}(\mathbf{x}) \leqslant \frac{-(\kappa\rho^2-1)}{\rho(1-2\kappa)}\right) \leqslant 2\exp\left(-\frac{m(\kappa\rho^2-1)^2}{2\rho^2(1-2\kappa)^2}\right) \tag{39}$$

Therefore, we have a bound on the probability of misclassification in a ball of radius ρ centered at a point $\mathbf{x} \sim \mu$ □

Taking this further, we would like to find a constraint on ρ such that we can guarantee that we can bound the above inequality given in (4.0.1) by a specific $\delta > \beta \geqslant 0$. That is we wish to find a bound on ρ s.t the following bound holds true for almost all $\mathbf{x} \sim \mu$.

$$\mathop{\mathbb{P}}_{\boldsymbol{v}\sim\rho\mathbb{S}}\left(\hat{k}(\boldsymbol{x}+\boldsymbol{v}) \neq \hat{k}(\boldsymbol{x})\right) \leqslant \delta \tag{40}$$

Set $\epsilon = C\rho/\sqrt{m}$, where $C = \sqrt{2\log(2/(\delta-\beta))}$. Therefore, since $\rho^2 = (\epsilon+1)/\kappa$, ρ satisfies the following equation:

$$\rho^2 = \kappa^{-1}\left(C\rho m^{-1/2} + 1\right)$$

The solution of this second order equation gives

$$\rho = \frac{C\kappa^{-1}m^{-1/2} + \sqrt{\kappa^{-2}C^2m^{-1} + 4\kappa^{-1}}}{2} \leqslant C\kappa^{-1}m^{-1/2} + \kappa^{-1/2} \tag{41}$$

Hence, for this choice of ρ, we have by construction

$$\mathop{\mathbb{P}}_{\mathbf{v}\sim S}\left(\alpha_{\mathbf{x}}^{-1}(\rho\boldsymbol{v} - \boldsymbol{\delta}(\mathbf{x}))^T\boldsymbol{H}_{\boldsymbol{z}}(\rho\boldsymbol{v} - \boldsymbol{\delta}(\mathbf{x})) + (\boldsymbol{\delta}(\mathbf{x}))^T(\rho\boldsymbol{v} - \boldsymbol{\delta}(\mathbf{x})) \leqslant 0\right) \leqslant \delta$$

Therefore, given a prpbability of misclassification on the sphere δ, we can find an upper bound for the radius of perturbation ρ s.t given any point $\mathbf{x} \sim \mu$, the probability of misclassification is no greater than δ.

Note that if we are in the high dimensional setting, we may approximate the Hessian, $H_{\mathbf{z}}$ as a real Wigner matrix. Then, we may approximate the constant β from the semi-circle distribution, assuming independence of the eigenvalues of $H_{\mathbf{z}}$. Therefore, we do not have to find a subspace $\mathcal{S} \in \mathbb{R}^d$ in the above theorem (4.0.1), further simplifying the problem setting. This is precisely the motivation for introducing RMT in this analysis.

Lemma 4.0.1. *Consider an independent Hessian (Wigner) matrix $H \in \mathbb{R}^{d\times d}$ with i.i.d entries H_{ij} with finite variance σ^2. In addition, suppose $0 < \kappa \leqslant 2\sigma$. Then,*

$$\mathbb{P}\left(\forall \boldsymbol{u} \in \mathbb{R}^2, \alpha_{\mathbf{x}}^{-1}\boldsymbol{u}^{T\boldsymbol{H}\boldsymbol{u}\geqslant\kappa\|\boldsymbol{u}\|_2^2}\right) \geqslant 1 - \left(\int_{-\infty}^{\kappa}\frac{1}{2\pi\sigma^2}\sqrt{4\sigma^2 - x^2}dx\right)^2 \tag{42}$$

Proof. If we take the eigenvalues of $\boldsymbol{H}$ to be independent, then the probability that two randomly selected eigenvalues (corresponding to two randomly chosen vectors) are both

smaller than κ is given by

$$\left(\int_{-\infty}^{\kappa} \frac{1}{2\pi\sigma^2}\sqrt{4\sigma^2 - x^2}dx\right)^2$$

Then, the probability that the two randomly selected eigenvalues are both greater than κ is given by

$$1 - \left(\int_{-\infty}^{\kappa} \frac{1}{2\pi\sigma^2}\sqrt{4\sigma^2 - x^2}dx\right)^2$$

Where we have approximated the eigenvalue distrubution of H by the semi-circle distribution. Note that this distribution is independent of the basis, therefore it does not matter whether the randomly chosen vectors are not eigenvectors of $\boldsymbol{H}$. This resut holds w.l.o.g. We have not explicitly given the integral as it does not give any further insight. □

5 Monte-Carlo Simulation

The following is pseudo-code for the Monte-Carlo simulation performed to investigate the bound (4.0.1) in (8). We include this for completeness sake.

Algorithm 1: Estimate of the probability of misclassification of points within a circle C_0

```
function ProbabilityOfMisclassification (x, ρ, n = n_0);
  Input  : centre of circle, x, radius of circle C_0, ρ > 0, and the number of inner test
           points, n
  Output: proportion of points within the circle that are misclassified
k ≔ k̂(x)
prop ≔ 0
for count = 1 to n do
    generate random points uniformly within the circle C_0
    if class of random point ≠ k :
    prop + 1
    count + 1
end
return prop/count
```

6 Jacobian and Hessian Regularisation

We introduce a regularisation term to the loss objective which penalises large curvature of the decision boundary. Previously, we considered the Jacobian and Hessian of the the decision function $F(\mathbf{x}) = f_{\hat{k}(\mathbf{x})}(\mathbf{x}) - \max_{k \neq \hat{k}(\mathbf{x})} f_k(\mathbf{x})$ w.r.t inputs.

In practice, it is easier to consider the cross output terms $g_{ij}(\mathbf{x}) = f_i(\mathbf{x}) - f_j(\mathbf{x})$. Note that given an input $\mathbf{x}$, when $i = \hat{k}(\mathbf{x})$ and $j = \arg\max_{k \neq \hat{k}(\mathbf{x})} f_k(\mathbf{x})$, $g_{ij}(\mathbf{x}) = F(\mathbf{x})$. Therefore, by minimising the curvature of the function g_{ij}, $1 \leqslant i, j \leqslant n$, we implicitly minimise the curvature of the decision function. We minimise the curvature by minimising the Jacobian and Hessian cross terms, $\|\nabla f_i - \nabla f_j\|_2^2$ and $\|\nabla^2 f_i - \nabla^2 f_j\|_2^2$ respectively.

We show that can minimise the curvature of g_{ij} by minimising the Frobenius norm of the Jacobian and Hessian of f. We prove this below for the Jacobian, which is then extended to the Hessian.

Lemma 6.0.1 (Bounded Jacobian Frobenius norm implies Bounded Jacobian Cross Terms)**.** *If the Frobenius norm of the Jacobian (w.r.t inputs) of classifier f, $\|J(\mathbf{x})\|_F^2 \leqslant M$, then $\|\nabla g_{ij}\|_2 \leqslant \sqrt{M}$.*

Proof.

$$\|J(\mathbf{x})\|_F^2 = \sum_i \sum_j \left| \frac{\partial f_i}{\partial x_j}(\mathbf{x}) \right|^2 \leqslant M \tag{43}$$

Then, notice that

$$\|\nabla g_{ij}(\mathbf{x})\|_2^2 = \|\nabla f_i(\mathbf{x}) - \nabla f_j(\mathbf{x})\|_2^2 = \sum_l \left| \frac{\partial f_i}{\partial x_l}(\mathbf{x}) - \frac{\partial f_j}{\partial x_l}(\mathbf{x}) \right|^2 \leqslant \sum_l \left| \frac{\partial f_i}{\partial x_l}(\mathbf{x}) \right|^2 + \left| \frac{\partial f_j}{\partial x_l}(\mathbf{x}) \right|^2 \tag{44}$$

Then, we have that $\|\nabla g_{ij}(\mathbf{x})\| = \|\nabla f_i(\mathbf{x}) - \nabla f_j(\mathbf{x})\| \leqslant \sqrt{M}$ □

Lemma 6.0.2 (Bounded Hessian norm implies Locally Bounded Jacobian Cross Terms)**.** *Given $\mathbf{x} \in \mathbb{R}^d$, assume there exists $M > 0$ s.t $\forall \mathbf{y} \in B(\mathbf{x}, R)$, $\sum_{i=1}^L \|\nabla^2 f_i(\mathbf{y})\|^2 \leqslant M^2$ Then, for any $i, j \in \{1, \ldots, L\}$ we have that*

$$\max_{\mathbf{y} \in B(\mathbf{x}, R)} \|\nabla g_{ij}(\mathbf{y})\|^2 \leqslant (RM)^2 + \|\nabla g_{ij}(\mathbf{x})\|^2 \tag{45}$$

Proof. Since $\sum_{i=1}^L \|\nabla^2 f_i(\mathbf{x})\|^2 \leqslant M^2$, it follows that for $\mathbf{y} \in B(\mathbf{x}, R)$ and for each i, j,

$$\left\|\nabla^2 g_{ij}(\mathbf{y})\right\|^2 = \left\|\nabla^2 f_i(\mathbf{y}) - \nabla^2 f_j(\mathbf{y})\right\|^2 \leqslant \left\|\nabla^2 f_i(\mathbf{y})\right\|^2 + \left\|\nabla^2 f_j(\mathbf{y})\right\|^2 \leqslant M^2 \tag{46}$$

If we assume $L \geqslant 2$ (which is sensible), by (2.3.2), it follows that for any $\mathbf{x}$ and $\mathbf{y} \in B(\mathbf{x}, R)$, $\|\nabla g_{ij}(\mathbf{x}) - \nabla g_{ij}(\mathbf{y})\| \leqslant R \cdot M$. Then, we have that

$$\|\nabla g_{ij}(\mathbf{y})\|^2 \leqslant \|\nabla g_{ij}(\mathbf{x})\|^2 + \|\nabla g_{ij}(\mathbf{y}) - \nabla g_{ij}(\mathbf{x})\|^2 \leqslant \|\nabla g_{ij}(\mathbf{x})\|^2 + R^2 M^2 \tag{47}$$

The rest follows by plugging in this bound. □

This motivates using the Hessian terms as regularisation. Note [17] propose Hessian regularisation while this paper was being written. However, the method and algorithm is different to ours. In the following, we consider notation from [4]. Consider training an RNN optimised with Stochastic Gradient Descent (SGD) on a supervised learning problem. SGD trains the classifier with mini-batches $\mathcal{B}$ of labelled data, $\{\boldsymbol{x}^\alpha, \boldsymbol{y}^\alpha\}_{\alpha\in\mathcal{B}}$. We do not introduce notation to deal with the time dependence for simplicity.

At each iteration, a supervised loss function, $\mathcal{L}_{\text{super}}$ is minimised (possibly with another regulariser $\mathcal{R}(\theta)$) over a mini-batch $\mathcal{B}$ in parameter space. In other words, we minimise the following bare loss function:

$$\mathcal{L}_{\text{bare}}\ (\{\boldsymbol{x}^\alpha, \boldsymbol{y}^\alpha\}_{\alpha\in\mathcal{B}}\,;\theta) = \frac{1}{|\mathcal{B}|}\sum_{\alpha\in\mathcal{B}} \mathcal{L}_{\text{super}}\ [f\,(\boldsymbol{x}^\alpha)\,;\boldsymbol{y}^\alpha] + \mathcal{R}(\theta) \tag{48}$$

We propose such a regulariser $\mathcal{R}(\theta)$ which misimises the curvature of g_{ij} by minimising the Forbenius norm of the Hessian. Note this idea is motivated by [4] who propose a Jacobian regulariser based on the Jacobian Frobenius norm. Consider the Taylor expansion of the classifier output $f(\mathbf{x}) = \mathbf{z}$. Note in this section we denote the classifier output by $f(\mathbf{x}) = \mathbf{z}$ so we can easily identify that it is a vector.

$$f_c(\mathbf{x}+\boldsymbol{\delta}) \approx f_c(\mathbf{x}) + \sum_{i=1}^{I}\delta_i\frac{\partial f_c}{\partial x_i}(\mathbf{x}) + \frac{1}{2}\sum_{i,j=1}^{I}\delta_i\delta_j\frac{\partial^2 f_c}{\partial x_i\partial x_j}(\mathbf{x}) \tag{49}$$

$$\begin{aligned} J_{c;i}(\mathbf{x}) &\coloneqq \frac{\partial f_c}{\partial x_i}(\mathbf{x}) \quad H_{c;i,j}(\mathbf{x}) \coloneqq \frac{\partial f_c}{\partial x_i\partial x_j}(\mathbf{x}) \\ \|\boldsymbol{J}(\mathbf{x})\|_{\text{F}}^2 &\equiv \sum_{c,i}\left[J_{c;i}(\mathbf{x})\right]^2 \quad \|\boldsymbol{H}(\mathbf{x})\|_{\text{F}}^2 \coloneqq \sum_{c;i,j}\left[H_{c;i,j}(\mathbf{x})\right]^2 \end{aligned} \tag{50}$$

In practice, classifier optimization is based on backpropogation through time, (BPTT) [23] which is based on the gradient of the loss w.r.t parameters. Therefore, it is essential to efficiently compute gradients of the Frobenius norm of the Hessian. ML libraries are generally typicallly built in with automatic differentiation methods [1] which compute the derivative of a vector valued function $f(\mathbf{x}) = \boldsymbol{z}$ w.r.t inputs, provided $\boldsymbol{z}$ is first contracted with another fixed vector. To take advantage of this functionality, we rewrite the squared Frobenius norm

using the trace property as

$$\|\boldsymbol{H}(\mathbf{x})\|_{\mathrm{F}}^2 = \mathrm{Tr}\left(\boldsymbol{H}^T\boldsymbol{H}\right) = \sum_{\{\boldsymbol{e}\}} \boldsymbol{e}^T\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{e} = \sum_{\{\boldsymbol{e}\}} \left[\frac{\partial^2(\boldsymbol{e}\cdot\boldsymbol{z})}{\partial^2\mathbf{x}}\right]^T \left[\frac{\partial^2(\boldsymbol{e}\cdot\boldsymbol{z})}{\partial^2\mathbf{x}}\right] \tag{51}$$

where $\{\boldsymbol{e}\}$, is an orthonormal basis. In the last step, we move the basis vectors $\boldsymbol{e} \in \{\boldsymbol{e}\}$ inside the derivative. We can utilise automatic differentiation in this setting. To mke computation more efficient, we express (51) in terms of the expectation of an unbiased estimator:

$$\|\boldsymbol{H}(\mathbf{x})\|_{\mathrm{F}}^2 = L\mathbb{E}_{\hat{\boldsymbol{v}}\sim\mathbb{S}^{L-1}}\left[\hat{\boldsymbol{v}}(\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}})\hat{\boldsymbol{v}}^{\mathrm{T}}\right] \tag{52}$$

where the random vector $\hat{\boldsymbol{v}}$ is drawn from the $(L-1)$-dimensional unit sphere $\mathbb{S}^{L-1}$.

We argue for the identity (52) below. Note $\mathbb{E}_{\hat{\boldsymbol{v}}\sim\mathbb{S}^{L-1}}[F(\hat{\boldsymbol{v}})]$ is the average value of the (arbitrary) function F taken over L -dimensional vectors $\boldsymbol{v}$ uniformly sampled from the unit sphere $\mathbb{S}^{L-1}$. In our derivation, we assume the following from [4]:

$$\mathbb{E}_{\hat{\boldsymbol{v}}\sim\mathbb{S}^{L-1}}[F(\hat{\boldsymbol{v}})] = \int \mathrm{d}\mu(\boldsymbol{O})F(\boldsymbol{O}\boldsymbol{e}) \tag{53}$$

where $\boldsymbol{e} \in \mathbb{S}^{L-1}$ and $\int \mathrm{d}\mu(\boldsymbol{O})[\ldots]$ is an integral over orthogonal matrices $\boldsymbol{O}$ over the Haar measure with normalization $\int \mathrm{d}\mu(\boldsymbol{O})[\ldots] = 1$. The following derivation is almost identical to that presented in [4]

$$\begin{aligned}
\|\boldsymbol{H}(\mathbf{x})\|_F^2 &= \mathrm{Tr}\left(\boldsymbol{H}^T\boldsymbol{H}\right) \\
&= \int \mathrm{d}\mu(\boldsymbol{O})\,\mathrm{Tr}\left(\boldsymbol{O}^T\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{O}\right) \\
&= \int \mathrm{d}\mu(\boldsymbol{O}) \sum_{\{\boldsymbol{e}\}} \boldsymbol{e}^T\boldsymbol{O}^T\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{O}\boldsymbol{e} \\
&= \sum_{\{\boldsymbol{e}\}} \mathbb{E}_{\hat{v}\sim\mathbb{S}^{L-1}}\left[\hat{\boldsymbol{v}}^T\boldsymbol{H}^T\boldsymbol{H}\hat{\boldsymbol{v}}\right] \\
&= L\mathbb{E}_{\hat{v}\sim\mathbb{S}^{L-1}}\left[\hat{\boldsymbol{v}}^T(\boldsymbol{H}^T\boldsymbol{H})\hat{\boldsymbol{v}}\right]
\end{aligned} \tag{54}$$

In the second line, we introduce the identity matrix as $\mathbf{1} = \boldsymbol{O}^T\boldsymbol{O}$ and exploit the cyclic property of the trace. In the next line, we express the trace as a sum over the orthonormal basis $\{\boldsymbol{e}\}$. We use (52) in the second to last line. Finally, in the last line we compute the sum, noting that the expectation is independent of the basis vectors $\boldsymbol{e}$.

Using this relationship, we can use samples of n_{proj} random vectors $\hat{\boldsymbol{v}}^\mu$ to estimate the

square of the norm as:

$$\|\boldsymbol{H}(\mathbf{x})\|_{\mathrm{F}}^{2} \approx \frac{1}{n_{\mathrm{proj}}} \sum_{\mu=1}^{n_{\mathrm{proj}}} \left[\frac{\partial^{2} \left(\hat{\boldsymbol{v}}^{\mu} \cdot \boldsymbol{z} \right)}{\partial^{2} \boldsymbol{x}} \right]^{2} \tag{55}$$

The motivation for this is to propose an efficient algorithm to find the gradient of the Hessian Frobenius norm. This justifies the use of a Hessian regularisation term in the form of the Hessian Frobenius norm. The loss function we use is then given by

$$\mathcal{L}_{\mathrm{joint}}^{\mathcal{B}}(\theta) = \mathcal{L}_{\mathrm{bare}}\left(\{\mathbf{x}^{\alpha}, \mathbf{y}^{\alpha}\}_{\alpha \in \mathcal{B}}; \theta \right) + \frac{\lambda_{\mathrm{HR}}}{2} \left[\frac{1}{|\mathcal{B}|} \sum_{\alpha \in \mathcal{B}} \|\boldsymbol{H}\left(\mathbf{x}^{\alpha}\right)\|_{\mathrm{F}}^{2} \right] \tag{56}$$

where λ_{HR} is a parameter that gives the ratio of importance of the Hessian regularisation term to the bare loss, $\mathcal{L}_{bare}$ in the joint loss function, $\mathcal{L}_{joint}$.

Algorithm 2: Approximate gradient of the Hessian regulariser

function HessianFrobeniusNormGradient $(\{\boldsymbol{x}^{\alpha}, \boldsymbol{y}^{\alpha}\}_{\alpha \in \mathcal{B}}, \mathbf{z}^{\alpha})$;
Input : mini-batch of $|\mathcal{B}|$ examples $\{\boldsymbol{x}^{\alpha}, \boldsymbol{y}^{\alpha}\}_{\alpha \in \mathcal{B}}$, model outputs $\mathbf{z}^{\alpha}$, and number of projections n_{proj}
Output: Square of the Frobenius norm of the Jacobian $\mathcal{H}_F$ and its gradient w.r.t parameters θ, $\nabla_{\theta}\mathcal{H}_F$
$\mathcal{H}_F \coloneqq 0$
for *count = 1 to* n_{proj} **do**
  for α *= 1 to* $\mathcal{B}$ **do**
    Sample a vector $\hat{\boldsymbol{v}}^{\alpha}$ uniformly from the unit sphere
    Normalise $\hat{\boldsymbol{v}}^{\alpha}$ by setting $\hat{\boldsymbol{v}}^{\alpha} = \hat{\boldsymbol{v}}^{\alpha} / \|\hat{\boldsymbol{v}}^{\alpha}\|$
    $\mathcal{H}v = \partial^2(\boldsymbol{z}^{\alpha} \cdot \hat{\boldsymbol{v}}^{\alpha}) / \partial \mathbf{x}^2$
    $\mathcal{H}_F += (\mathcal{H}v)^T \mathcal{H}v / (n_{proj}\mathcal{B})$
  end
end
$\nabla_{\theta}\mathcal{H}_F = \partial \mathcal{H}_F / \partial \theta$
return $\mathcal{H}_F, \nabla_{\theta}\mathcal{H}_F$

7 Spectral Distribution of the Loss Surface

Previously, we considered the decision function F and the decision boundary B of a classifier f. We now move to the loss surface framework. Given a loss function $\mathcal{L} : \mathbf{x} \rightarrow R$, the loss surface is defined as the set $\{(\mathbf{x}, \mathcal{L}(\mathbf{x}))$ s.t $\mathbf{x} \in \mathbb{R}^d\}$. The loss surface is the association of each data point $\mathbf{x}$ with its loss value $\mathcal{L}(\mathbf{x})$. From [15], we note that robustness increases with a decrease in curvature of the loss surface. We aim to investigate the robustness of a classifier by studying the spectral distribution of the Hessian of the loss function (w.r.t inputs). Note

that in the other sections, by input we refer to a column vector $\mathbf{x} \in \mathbb{R}^d$. However, in this section exclusively by input we refer to a matrix of training data, $\mathbf{x} \in \mathbb{R}^{d \times m}$. We do this so that we can utilise RMT.

Since we are in the framework of supervised learning, we assume no knowledge of the loss surface other than what we can infer from the given training dataset. This is of practical importance since in general, only a labelled dataset is available to train a classifier. In this regard, we extend the work of [18], leveraging RMT to work out the spectral distribution of a general single layer RNN with cross-entropy loss (rather than mean square error), with the hessian of the empirical loss taken with respect to inputs (rather than weights). Since the hessian is taken with respect to inputs, we can deduce global properties of the loss surface rather than at critical points in [18]. It is recommended to refer to [18] while reading this section, as the woorking out is similar.

Consider a general single layer multi-class RNN with cross-entropy loss, with predictions given by

$$\mathbf{y}^{\langle 1 \rangle} = \sigma\left(\mathbf{W_{ya}}\sigma\left(\mathbf{W_{aa}}\mathbf{a}^{\langle 0 \rangle} + \mathbf{W_{ax}}\mathbf{x}^{\langle 1 \rangle} + \mathbf{b}_a\right) + \mathbf{b}_y\right) \tag{57}$$

Let $\mathbf{x}^{\langle 1 \rangle}, \in \mathbb{R}^{n_0 \times m}$ denote a random sample of m data points and $\mathbf{a}^{\langle 0 \rangle} \in \mathbb{R}^{n_0 \times m}$ denote the variable bias input. The weights, $\mathbf{W_{ax}}, \mathbf{W_{ad}} \in \mathbb{R}^{n_1 \times n_0}$, $\mathbf{b}_a \in \mathbb{R}^{n_1 \times m}$, $\mathbf{W_{ya}} \in \mathbb{R}^{n_2 \times n_1}$, and $\mathbf{b}_y \in \mathbb{R}^{n_2 \times m}$ are taken to be random. We assume $n_0 \approx n_1 \approx n_2$ and define $\phi \coloneqq n_1/m$ to be a measure of the size of the system. We shall compute the limiting spectral density of the Hessian of the cross-entropy loss function with respect to the input $\mathbf{x}^{\langle 1 \rangle}$. The steps we take are described below:

1. Decompose the Hessian of the loss function into two matrices $\boldsymbol{H} = \boldsymbol{H}_0 + \boldsymbol{H}_1$ where $\boldsymbol{H}_0$ is a real Wigner matrix and $\boldsymbol{H}_1$ is a real Wishart matrix. From (2.4.4) and (2.4.5) we know the limiting spectral densities $\rho_{\boldsymbol{H}_0}$ and $\rho_{\boldsymbol{H}_1}$ of $\boldsymbol{H}_0$ and $\boldsymbol{H}_1$ respectively.

2. Compute the Stieltjes transforms $G_{\boldsymbol{H}_0}, G_{\boldsymbol{H}_1}$ of $\rho_{\boldsymbol{H_0}}$ and $\rho_{\boldsymbol{H}_1}$ respectively.

3. From the Stieltjes transforms $G_{\boldsymbol{H}_0}, G_{\boldsymbol{H}_1}$, deduce the $\mathcal{R}$ transforms $\mathcal{R}_{\boldsymbol{H}_0}$ and $\mathcal{R}_{\boldsymbol{H}_1}$.

4. From $\mathcal{R}_{\boldsymbol{H}_0+\boldsymbol{H}_1} = \mathcal{R}_{\boldsymbol{H}_0} + \mathcal{R}_{\boldsymbol{H}_1}$ (assume independence) compute the Stieltjes transform $G_{\boldsymbol{H}_0+\boldsymbol{H}_1}$.

5. Invert the Stieltjes transform to obtain the limiting spectral distribution of $\boldsymbol{H}$ $\rho_{\boldsymbol{H}=\boldsymbol{H}_0+\boldsymbol{H}_1}$.

Since $\mathbf{x}^{\langle 1\rangle} \in \mathbb{R}^{n_0 \times m}$, we have m input examples with n_0 features each. Since $\mathbf{y}^{\langle 1\rangle} \in \mathbb{R}^{n_2 \times m}$, we have n_2 classes. In the following $\hat{y}_{\mu\nu}$ refers to predictions for the μ-th class of example ν in the dataset $\mathbf{x}$ (we remove the time dependence here), while $y_{\mu\nu}$ refers to the true values. We consider the cross-entropy loss since we are dealing with the multi-class setting given by:

$$\mathcal{L}_{CE}(y_{\mu\nu}(\mathbf{x}), \hat{y}_{\mu\nu}(\mathbf{x})) = -\frac{1}{m}\sum_{\mu,\nu=1}^{n_2,m} y_{\mu,\nu}(\mathbf{x}) \log \hat{y}_{\mu,\nu}(\mathbf{x}) = \epsilon n_2 \geqslant \epsilon n_2 \log(\alpha) \tag{58}$$

We decompose the Hessian, $\boldsymbol{H}$ into two parts, $H = [\boldsymbol{H}_0]_{ij} + [\boldsymbol{H}_1]_{ij}$. Note that terms in the cross-entropy loss only arise from predictions of the correct class. Therefore, it is convenient to define $\alpha \in [0, 1]$ as the minimal prediction for the correct class across all sampled data points. Note by x_i, x_j we mean entries of the input matrix $\mathbf{x}$

$$[\boldsymbol{H}_0]_{\alpha\beta} = \frac{1}{m}\sum_{\mu,\nu=1}^{n_2,m} \frac{y_{\mu\nu}(\mathbf{x})}{(\hat{y}_{\mu\nu}(\mathbf{x}))^2}\frac{\partial \hat{y}_{\mu\nu}(\mathbf{x})}{\partial x_i}\frac{\partial \hat{y}_{\mu\nu}(\mathbf{x})}{\partial x_j} \leqslant \frac{1}{m\alpha^2}\left[\boldsymbol{J}\boldsymbol{J}^T\right]_{\alpha\beta} \tag{59}$$

$$[\boldsymbol{H}_1]_{ij} \equiv \frac{1}{m}\sum_{\mu,\nu=1}^{n_2,m} \frac{y_{\mu\nu}(\mathbf{x})}{\hat{y}_{\mu\nu}(\mathbf{x})}\left(\frac{\partial^2 \hat{y}_{\mu\nu}(\mathbf{x})}{\partial x_i \partial x_j}\right) \leqslant \frac{1}{\alpha}[\boldsymbol{H}_1]_{ij} \tag{60}$$

The Hessian, $\nabla^2_{\mathbf{x}}\mathcal{L}_{CE}(y_{\mu\nu}, \hat{y}_{\mu\nu})$given the training set $\mathbf{x}$ is:

$$\begin{aligned}
\boldsymbol{H} &= \frac{1}{m}\sum_{\mu,\nu=1}^{n_2,m}\left(\frac{y_{\mu\nu}(\mathbf{x})}{(\hat{y}_{\mu\nu}(\mathbf{x}))^2}\frac{\partial \hat{y}_{\mu\nu}(\mathbf{x})}{\partial x_i}\frac{\partial \hat{y}_{\mu\nu}(\mathbf{x})}{\partial x_j} - \frac{y_{\mu\nu}(\mathbf{x})}{\hat{y}_{\mu\nu}(\mathbf{x})}\frac{\partial^2 \hat{y}_{\mu\nu}(\mathbf{x})}{\partial x_i \partial x_j}\right) \\
&= \frac{1}{m}\sum_{\mu,\nu=1}^{n_2,m}\frac{y_{\mu\nu}(\mathbf{x})}{(\hat{y}_{\mu\nu}(\mathbf{x}))^2}\frac{\partial \hat{y}_{\mu\nu}(\mathbf{x})}{\partial x_i}\frac{\partial \hat{y}_{\mu\nu}(\mathbf{x})}{\partial x_j} - \frac{1}{m}\sum_{i,\mu=1}^{n_2,m}\frac{y_{\mu\nu}(\mathbf{x})}{\hat{y}_{\mu\nu}(\mathbf{x})}\left(\frac{\partial^2 \hat{y}_{\mu\nu}(\mathbf{x})}{\partial x_i \partial x_j}\right) \\
&= [\boldsymbol{H}_0]_{ij} - [\boldsymbol{H}_1]_{ij} \\
&\leqslant \frac{1}{m(\alpha)^2}\left[\boldsymbol{J}\boldsymbol{J}^T\right]_{ij} - \frac{1}{\alpha}[\boldsymbol{H}_1]_{ij}
\end{aligned} \tag{61}$$

We assume $\boldsymbol{H_0}$ and $\boldsymbol{H_1}$ are freely independent. In addition, we assume the data and weights are i.i.d. normal random variables which has been theoretically and experimentally justified to be a reasonable assumption for a single layer NN [18]. Further analysis is necessary to experimentally verify this for RNNs although we leave this for future work.

Assuming the elements of $\boldsymbol{J}$ and $\boldsymbol{H}_1$ are i.i.d. normal random variables, we approximate $\boldsymbol{H}_0$ as a Wishart matrix and $\boldsymbol{H}_1$ as a Wigner matrix. Both matrices will be sparse since terms in the cross-entropy loss only arise from predictions of the correct class. However, the spectral distribution of a sparse matrix will still have a similar distribution to a Wigner

or Wishart matrix by the property of universality [22, 10]. It is clear that for a trained RNN, the entries of $\boldsymbol{J}$ and $\boldsymbol{H}_1$ will not be i.i.d. The lack of independence discussed above is a significant issue and until this is addressed, this analysis can only be considered to be a simple theoretical toy model.

We study the spectral distribution of $\boldsymbol{H}$ as a function of the empirical loss ϵ and the minimal prediction value α. This distribution depends only on the relative scaling between $\boldsymbol{H}_0$ and $\boldsymbol{H}_1$ [18]. Therefore, w.l.o.g. we take $\sigma_{\boldsymbol{H}_0} = 1$ and $\sigma_{\boldsymbol{H}_1} = \epsilon \cdot log(\alpha)$. Note by the normality assumption, the spectral distribution of -$\boldsymbol{H}_1$ is identical to that of $\boldsymbol{H}_1$

$$\rho_{\boldsymbol{H}_0}(\lambda) = \rho_{\text{MP}}(\lambda; 1, \phi), \quad \rho_{\boldsymbol{H}_1}(\lambda) = \rho_{\text{SC}}(\lambda; \sqrt{\epsilon \cdot log(\alpha)}) \tag{62}$$

The Steiltjes Transform of which are $G_{\boldsymbol{H}_0}$ and $G_{\boldsymbol{H}_1}$, which are standard results of RMT. We then compute the $\mathcal{R}$ transforms, given by:

$$\mathcal{R}_{\boldsymbol{H}_0}(z) = \frac{1}{1 - z\phi}, \quad \mathcal{R}_{\boldsymbol{H}_1}(z) = (\epsilon \cdot log(\alpha))z \tag{63}$$

Asssuming independance of $\boldsymbol{H}_0$ and $\boldsymbol{H}_1$, we compute the $\mathcal{R}$ transform of $\boldsymbol{H}$:

$$\mathcal{R}_H = \mathcal{R}_{\boldsymbol{H}_0} + \mathcal{R}_{\boldsymbol{H}_1} = \frac{1}{1 - z\phi} + (\epsilon \cdot log(\alpha))z \tag{64}$$

From the $\mathcal{R}$ transform functional equation, we note the Stieltjes transform of $\boldsymbol{H}$, $G_{\boldsymbol{H}}$ solves the following cubic equation:

$$(\epsilon \cdot \log(\alpha)\phi)G_{\boldsymbol{H}}^3 - (\epsilon \cdot log(\alpha) + z\phi)G_{\boldsymbol{H}}^2 + (z + \phi - 1)G_{\boldsymbol{H}} - 1 = 0 \tag{65}$$

The correct root of this equation exhibits and inverse relationship with z in the asymptotic limit. That it, $G_H \sim 1/z$ as $z \to \infty$ [21]. From this root, the spectral density can be derived through the Stieltjes inversion formula. We solve for this numerically and plot it in (7). We note the similarity of the distribution for the single layer NN and single layer RNN case, suggesting a universality across architectures.

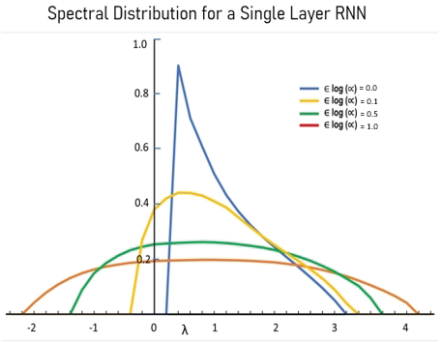

Figure 9: Sepctral distribution of a single layer RNN where $\phi = 0.5$. The spectral density resembles the Marchenko-Pastur distribution for small $\epsilon, \log \alpha$. As $\epsilon, \log \alpha$ the spectrum becomes semicircular.

8 Experiments

Adversarial training is commonly used as a defense against adversarial attack [24]. An adversarial example is an instance of data with minimal perturbations added which results in misclassification. In adversarial training, a set of adversarial examples is added to the training set. This allows the classifier to learn how to be more robust to adversarial attack.

We focus our attention on Projected Gradient Descent (PGD) [12, 24]. In the case of the PGD attack, the adversary crafts adversarial examples, having full knowledge of the model architecture. Given a point $\mathbf{x}$, the PGD adversary achieves this by finding the perturbation $\boldsymbol{\rho}(\mathbf{x})$ that maximises the loss function at $\mathbf{x}$, while keeping the size of the perturbation within $\|\boldsymbol{\rho}(\mathbf{x})\| \leqslant \epsilon$ for some $\epsilon > 0$. PGD is a gradient based attack with step size given by the parameter α.

Figure 10 shows the performance of Hessian and Jacobian regularisation against regular training (without regularisation). In each instance, we consider a two layer MNIST RNN model. The MNIST dataset [7] consists of 60000 labelled handwritten digits ranging from 0 to 9, which we split into 55000 training images, 5000 validation images and 10000 test images. Each image is 28×28 pixels which we treat as a sequence of data. We take the first row as the first (time-)step and so on. Therefore, n_{steps} = number of rows and $n_{features}$ = number of columns. We train a two layer RNN to predict the correct digit (class) given an image.

The parameters for the PGD attack performed for 30 epochs are $\epsilon \in [1, 30], \alpha = 2$. Note the parameters are normalised in the range (0, 1). Hessian regularisation is shown

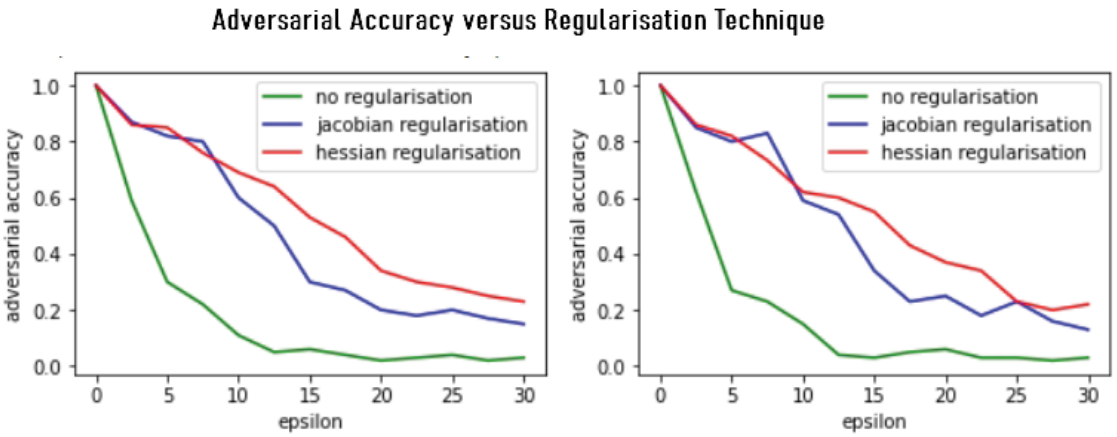

Figure 10: Adversarial accuracy versus regularisation technique curve for a PGD attack on an RNN trained on the MNIST dataset.

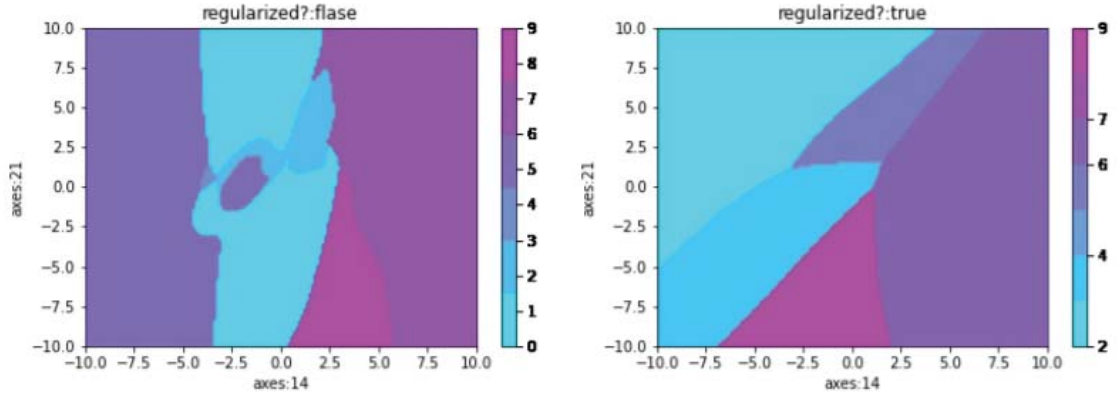

Figure 11: A 2-dimensional slice of the decision boundary along 2 randomly picked axes with Hessian regularisation (left) and without Hessian regularisation (right). This image is provided courtesy of my project partner Peter Fazekas.

to outperform Jacobian regularisation. The Jacobian regularisation method in [4] and the Hessian regularisation method (which is distinct to ours) in [14] has been shown to outperform adversarial training. Therefore our proposed Hessian regulariser has the potential to outperform adversarial training. We leave this to verify as future work.

9 Conclusion

We extended and contextualised well known results on the robustness of deep NNs to the case of single layer RNNs. As such, the majority of the results including the deterministic upper and lower bounds for adversarial robustness and the probabilistic bound for misclassification can be applied to classical feedforward NNs without issue. Single layer RNNs are the simplest type of the simplest architecture of sequence model. Therefore, further work would include extending these results to deeper RNNs, followed by other sequence model architectures. Our motivation for considering single layer RNNs was to develop a framework for robustness of sequence models, which we have successfully achieved. Further work will involve building up from this framework. Future work would also include performing more experiments on Hessian regularisation. We restricted ourselves to considering an RNN trained on the MNIST dataset. Traditionally a sequence model would not be trained on this dataset. Hence, we cannot say for certain if Hessian regularisation is superior to Jacobian regularisation. We briefly comment on the potential of Hessian regularisation to outperform adversarial training. However, this is yet to be tested and we leave this to future work. In addition, future work would include delving deeper into RMT, to identify the constant β in the probabilistic bound and also to relax the assumptions made in approximating the spectral distribution of the loss surface Hessian. The spectral distribution worked out is a theoretical toy model demonstrating the applicability of RMT in the context of ML. In particular, since less curves loss functions are more robust, it would be a natural next step to analyse the loss surface spectral distribution of loss functions with regularisation. This could provide further theoretical justification for Hessian regularisation. In conclusion, we have explored a breath of fields successfully in the context of the robustness of sequence models, and we hope that this inspires future work in this field.

References

[1] Alexey Andreyevich Radul Atılım Gunes Baydin, Barak A. Pearlmutter and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. 2017.

[2] Nicholas P Baskerville, Diego Granziol, and Jonathan P Keating. Applicability of random matrix theory in deep learning, 2021.

[3] Matthias Hein and Maksym Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation, 2017.

[4] Judy Hoffman, Daniel A. Roberts, and Sho Yaida. Robust learning with jacobian regularization, 2019.

[5] Jungsik Hwang. Modeling financial time series using lstm with trainable initial hidden states, 2020.

[6] Fabian Latorre, Paul Rolland, and Volkan Cevher. Lipschitz constant estimation of neural networks via sparse polynomial optimization. In *International Conference on Learning Representations*, 2020.

[7] Bottou L. Bengio Y. LeCun, Y. and P. Haffner. The mnist dataset of handwritten digits (images). 1998.

[8] Giacomo Livan, Marcel Novaes, and Pierpaolo Vivo. Introduction to random matrices. *SpringerBriefs in Mathematical Physics*, 2018.

[9] Cosme Louart, Zhenyu Liao, and Romain Couillet. A random matrix approach to neural networks, 2017.

[10] Xiaochuan Lu and Hitoshi Murayama. Universal asymptotic eigenvalue distribution of large n random matrices — a direct diagrammatic proof to marchenko-pastur law —, 2015.

[11] Yan Luo, Xavier Boix, Gemma Roig, Tomaso A. Poggio, and Qi Zhao. Foveation-based mechanisms alleviate adversarial examples. *CoRR*, abs/1511.06292, 2015.

[12] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2019.

[13] Jiri Matousek. Lectures on discrete geometry,volume 108 springer-verlag new york. 2002.

[14] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, Pascal Frossard, and Stefano Soatto. Robustness of classifiers to universal perturbations: A geometric perspective. In *International Conference on Learning Representations*, 2018.

[15] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Jonathan Uesato, and Pascal Frossard. Robustness via curvature regularization, and vice versa, 2018.

[16] Mario Morvan, Nikolaos Nikolaou, Angelos Tsiaras, and Ingo P. Waldmann. Detrending exoplanetary transit light curves with long short-term memory networks. *The Astronomical Journal*, 159(3):109, Feb 2020.

[17] Waleed Mustafa, Robert A. Vandermeulen, and Marius Kloft. Input hessian regularization of neural networks, 2020.

[18] Jeffrey Pennington and Yasaman Bahri. Geometry of neural network loss surfaces via random matrix theory, 2015.

[19] Tiago Pereira. Lectures on random matrices, department of mathematics, imperial college london, uk.

[20] Chawin Sitawarin, Arjun Nitin Bhagoji, Arsalan Mosenia, Mung Chiang, and Prateek Mittal. DARTS: deceiving autonomous cars with toxic signs. *CoRR*, abs/1802.06430, 2018.

[21] Terence Tao and Van Vu. Random covariance matrices: Universality of local statistics of eigenvalues. *The Annals of Probability*, 40(3), May 2012.

[22] Terence Tao, Van Vu, and Manjunath Krishnapur. Random matrices: Universality of esds and the circular law, 2009.

[23] Pantelis R. Vlachas, Jaideep Pathak, Brian R. Hunt, Themistoklis P. Sapsis, Michelle Girvan, Edward Ott, and Petros Koumoutsakos. Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics, 2020.

[24] Han Xu, Yao Ma, Haochen Liu, Debayan Deb, Hui Liu, Jiliang Tang, and Anil K. Jain. Adversarial attacks and defenses in images, graphs and text: A review, 2019.

[25] Fuxun Yu, Chenchen Liu, Yanzhi Wang, Liang Zhao, and Xiang Chen. Interpreting adversarial robustness: A view from decision surface in input space, 2018.