Formalizing Generalization and Adversarial Robustness of Neural Networks to Weight Perturbation

Yu-Lin Tsai, 10.17.2021
Authors: Yu-Lin Tsai @ {NYCU, TCFSH}, Chia-Yi Hsu @ NYCU, Chia-Mu Yu @ NYCU, Pin-Yu Chen @ IBM

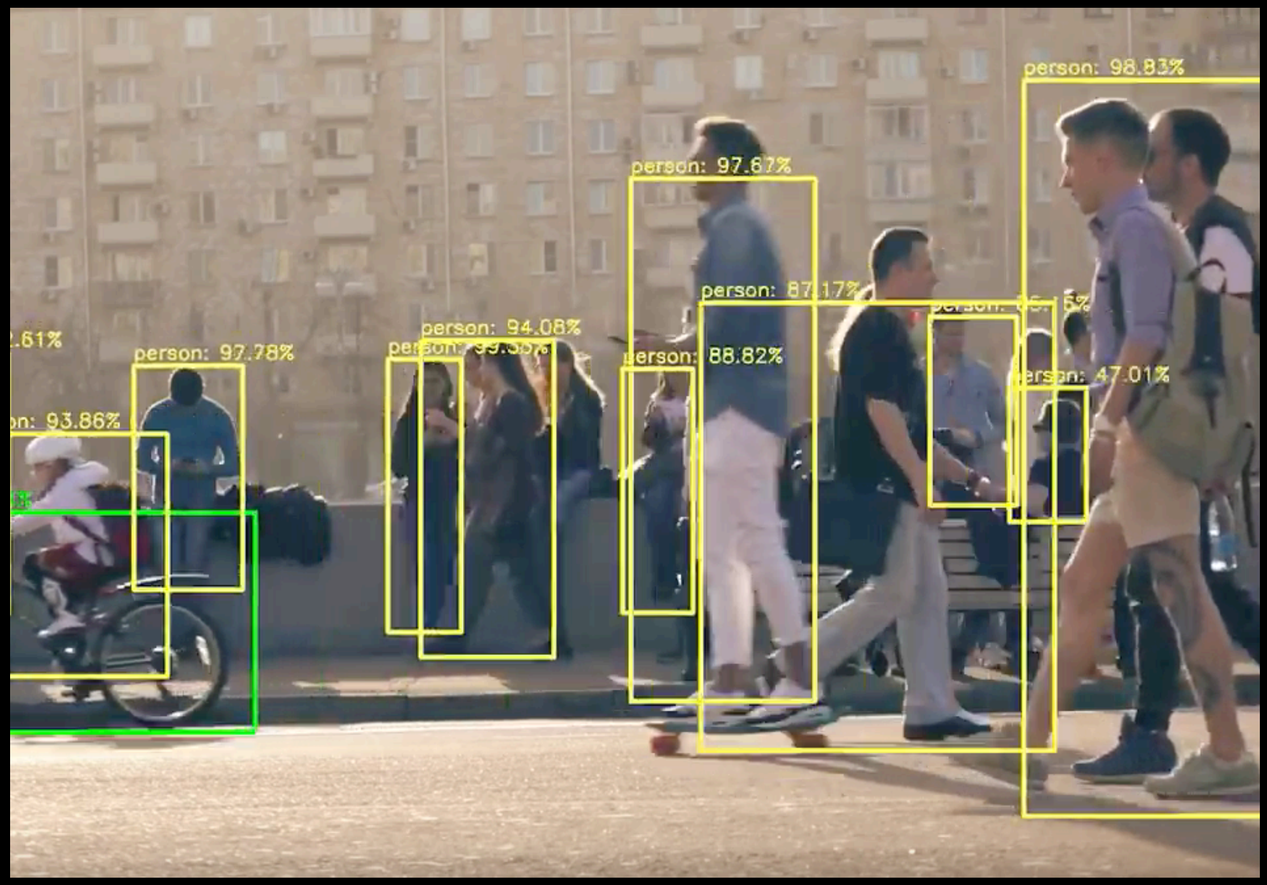
NeurIPS 2021

NYCU* : National Yang-Ming Chiao-Tung University
TCFSH: Taichung First Senior High school

Adversarial Examples

Threats to machine learning in real-world scenario

- In a normal scenario, a well-trained machine learning model could carry out specified task and perform greatly.
- However, if no explicit regularization or procedure was taken to enhance the robustness, a model could easily be susceptible to adversarial attacks.



Perspective from the eye of Machine Learning Models
(Credits: https://learn.alwaysai.co/detect-people-using-alwaysai)

Adversarial Examples Continued

Existence and Abundance of Adversarial Examples

- Mechanisms had been proposed to search for such examples
  - Adversarial Attack with PGD (Madry et al.)
  - FGSM (Goodfellow et al.)
- Real-World Examples have as well been found

Real-World Adversarial Examples
(Credits: Xu et al. Adversarial T-shirt! Evading Person Detectors in A Physical World)

Adversarial Perturbations

Adversarial Devastations on Weights

- Model also suffer from the perils of adversarial weight perturbations (Liu et al., 2017 / Zhao et al., 2019)
- Adversarial Weight Perturbation presents various interest of studies
  - Via perturbations, one could better understand the loss landscape and the relative generalization behavior (Cheney et al., 2017 / Widrow & Lehr, 1990)
  - Robustness to weight quantization is critical to reducing memory size of low-precision training and inferencing (Stutz et al., 2020 / Hubara et al., 2017)

Inspiration and Proposed Solution

- While input perturbation has been explicitly studied (Yin, D., 2019), there are few researches focus on exploring relationship between both the robustness and the generalization of the model against weight perturbation.
- While adversarial training is useful against input perturbation, straight-forward extension is not meaningful in the context of weight perturbation as the minimization and maximization are both conducted in the parameter space
- This paper investigates the worst-case error of a given neural network under weight perturbation, and proposes the surrogate loss for robust weight training and derives its Rademacher Complexity for generalization studies

Brief Introduction of Main Theorems

- Instead of providing direct proofs and presentation of theorems, here we would like to provide an overview on the implication and creation of these theorems.
- We will start by stating the pairwise margin bound, to the surrogate loss and the generalization bound against weight perturbation, till lastly we propose a theory-driven loss function against weight perturbation based on our observation.

Theory Results

Pairwise Margin Bound

- Firstly, we would like to measure the worst-case error in a feed-forward neural network under weight perturbation. Therefore, we choose the quantity of margin to express this certain bound provided in theorem 1
  - To be more precise, the term pairwise margin stands for the subtracted quantity of two classes which can be easily applied to look for the differences between confidence in each class.
  - The proof concept can be understand under the notion of layer-wise error propagation which are composed of the perturbation, preceding input magnitude, and posterior propagation via weight matrices.

Theory Results

Pairwise Margin Bound

- Here we present the general theorem along with the additional term definition for the better understanding of error propagation.

Theorem 3 (multiple-layer perturbation) *Let* $f_{\boldsymbol{W}}(\boldsymbol{x}) = \boldsymbol{W}^L(...\rho(\boldsymbol{W}^1\boldsymbol{x})...)$ *denote an L-Layer neural network. Given an index set* $I \subseteq [L]$*, we define the perturbed neural network as* $f_{\widehat{\boldsymbol{W}}}(\boldsymbol{x}) = \tilde{\boldsymbol{W}}^L(...\tilde{\boldsymbol{W}}^N...\rho(\tilde{\boldsymbol{W}}^1\boldsymbol{x})...)$ *with*

$$\begin{cases} \tilde{\boldsymbol{W}}^k = \boldsymbol{W}^k, \ \forall k \in [L] \setminus I \\ \tilde{\boldsymbol{W}}^k = \hat{\boldsymbol{W}}^k, \ \hat{\boldsymbol{W}}^k \in \mathbb{B}^{\infty}_{\boldsymbol{W}^k}(\epsilon_k), \forall k \in I \end{cases}$$

Then, for any pairwise margin between $f^{ij}_{\widehat{\boldsymbol{W}}}(\boldsymbol{x})$ *and* $f^{ij}_{\boldsymbol{W}}(\boldsymbol{x})$*,*

$$f^{ij}_{\widehat{\boldsymbol{W}}}(\boldsymbol{x}) \leq f^{ij}_{\boldsymbol{W}}(\boldsymbol{x}) + \underbrace{\sum_{k \in I \setminus \{L\}} \Delta(\epsilon_k; \boldsymbol{z}^{k-1*}; f)}_{\textit{Perturbed Layer Error}} + \underbrace{\mathbb{1}(L \in I) 2\epsilon_L \left\| \boldsymbol{z}^{L-1^*} \right\|_1}_{\textit{Final Layer Error}}$$

$$:= f^{ij}_{\boldsymbol{W}}(\boldsymbol{x}) + \underbrace{\eta^{ij}_{\boldsymbol{W}}(\boldsymbol{x}|I)}_{\textit{Error of Weight Perturbation}}$$

where $\boldsymbol{z}^{k^*} = \rho(\boldsymbol{W}^{k^*}...\rho(\boldsymbol{W}^{1^*}\boldsymbol{x})$ *with* $\boldsymbol{W}^{k^*}$ *defined as*

$$W^{k^*}_{i,j} = \begin{cases} W^k_{i,j} + \epsilon_k, \ \forall i,j \ \forall k \in [L] \cap I \setminus \{1\} \\ W^k_{i,j}, \ \forall i,j \ \forall k \in [L] \setminus (I \cup \{1\}) \end{cases}$$

$$W^{1^*}_{i,j} = \begin{cases} W^1_{i,j} + sgn([\boldsymbol{x}]_j)\epsilon_1, \ \forall i,j \ \ if \ 1 \in I \\ W^1_{i,j}, \ \forall i,j \qquad otherwise \end{cases}$$

and $\boldsymbol{z}^{0^*} = \boldsymbol{x}$*.*

Proof: See Appendix A.3.3.

Definition of Layer-wise Propagated Error

$$\Delta(\epsilon_k; \mathbf{z}; f) = \epsilon_k \left\| W^L_{i,:} - W^L_{j,:} \right\|_1 \|\mathbf{z}\|_1 (\Pi^{L-1}_{m=k+1} \|\mathbf{W}^m\|_{1,\infty})$$

Surrogate Loss against weight perturbation

- Secondly, in place of the adversarial training against weight perturbation, we derive an empirical surrogate loss which will be later used to analyze the generalization property of models against weight perturbation.

Lemma 2 (robust surrogate ramp loss) *Let $N \in [L]$ denote the perturbed layer index and let*

$$\Psi(f_{\boldsymbol{W}}(\boldsymbol{x})) = 2 \max_{k \in [K]} \epsilon_N \left\| W^L_{k,:} \right\|_1 \Pi_{m=1}^{N-1} \|\boldsymbol{W}^m\|_{1,\infty} \Pi_{k=1}^{L-N-1} \left\| (\boldsymbol{W}^{L-k})^T \right\|_{1,\infty} \|\boldsymbol{x}\|_1$$

be the worst case error and

$$\hat{\ell}(f_{\boldsymbol{W}}(\boldsymbol{x}), y) := \phi_\gamma \{ \underbrace{M(f_{\boldsymbol{W}}(\boldsymbol{x}), y)}_{\textit{margin}} - \underbrace{\Psi(f_{\boldsymbol{W}}(\boldsymbol{x}))}_{\textit{worst-case error}} \}$$

Then we have upper and lower bounds of $\hat{\ell}$ in terms of 0-1 losses expressed as

$$\max_{\hat{W}^N \in \mathbb{B}^\infty_{W^N}(\epsilon)} \mathbb{1}\{y \neq \arg\max_{y' \in [K]} [f_{\hat{\boldsymbol{W}}}(\boldsymbol{x})]_{y'}\} \leq \hat{\ell}(f_{\boldsymbol{W}}(\boldsymbol{x}), y) \leq \mathbb{1}\{M(f_{\boldsymbol{W}}(\boldsymbol{x}), y) - \Psi(f_{\boldsymbol{W}}(\boldsymbol{x})) \leq \gamma\}.$$

Generalization Gap under weight perturbation

- To proceed, we then turn to the focus of analyzing the generalization gap under weight perturbation via the previous surrogate loss and derive the following theorem.
- To recall, generalization gap measures the difference of model behavior in both testing and training time via the expected performance (loss). Therefore, it can be written in the form of
- $$\text{Testing Performance} \leq \text{Training Performance} + \text{Generalization Gap}$$

Generalization Gap under weight perturbation

Secondly, we denote the following empirical Rademacher complexity of both margin function and worst-case error as

$$\mathcal{R}_{\mathcal{S}}(M_{\mathcal{F}}) = \frac{4}{n^{3/2}} + \frac{60\log(n)\log(2d_{max})}{n} \|\mathbf{X}\|_F \left(\Pi_{h=1}^{L} s_h\right)\left(\sum_{j=1}^{L}\left(\frac{b_j}{s_j}\right)^{2/3}\right)^{3/2}$$

$$\mathcal{R}_{\mathcal{S}}(\Psi_{\mathcal{F}}) = \frac{2\epsilon_N \sup_{f\in\mathcal{F}} \Pi_{m=1}^{N-1}\|\mathbf{W}^m\|_{1,\infty} \Pi_{k=0}^{L-N-1}\left\|(\mathbf{W}^{L-k})^T\right\|_{1,\infty}}{n} \|\mathbf{X}\|_{1,2}$$

Theorem 4 (generalization gap for robust surrogate loss) *With Lemma 2, consider the neural network hypothesis class* $\hat{\mathcal{F}} = \{f_{\widehat{\boldsymbol{W}}}(\boldsymbol{x}) | \boldsymbol{W} = (\boldsymbol{W}^1, ...\hat{\boldsymbol{W}}^N, ..., \boldsymbol{W}^L), \hat{\boldsymbol{W}}^N \in \mathbb{B}^{\infty}_{\boldsymbol{W}^N}(\epsilon_N) \left\|\boldsymbol{W}^h\right\|_\sigma \leq s_h, \left\|(\boldsymbol{W}^h)^T\right\|_{2,1} \leq b_h, h \in [L]\}$. *For any* $\gamma > 0$, *with probability at least* $1-\delta$, *we have for all* $f_{\boldsymbol{W}}(\cdot) \in \mathcal{F}$

$$R(f) \leq R_n(f) + \frac{1}{\gamma}\Big(\underbrace{\mathcal{R}_{\mathcal{S}}(M_{\mathcal{F}})}_{\text{standard generalization gap}} + \underbrace{\mathcal{R}_{\mathcal{S}}(\Psi_{\mathcal{F}})}_{\text{complexity term of robust training}}\Big) + 3\sqrt{\frac{\log\frac{2}{\delta}}{2n}}$$

- As one can see in the above theorem, by training against weight perturbation, one may encounter additional complexity term which will in turn widen the generalization gap where it could grow out of bound when not properly contained.

Theory-Driven Loss Function

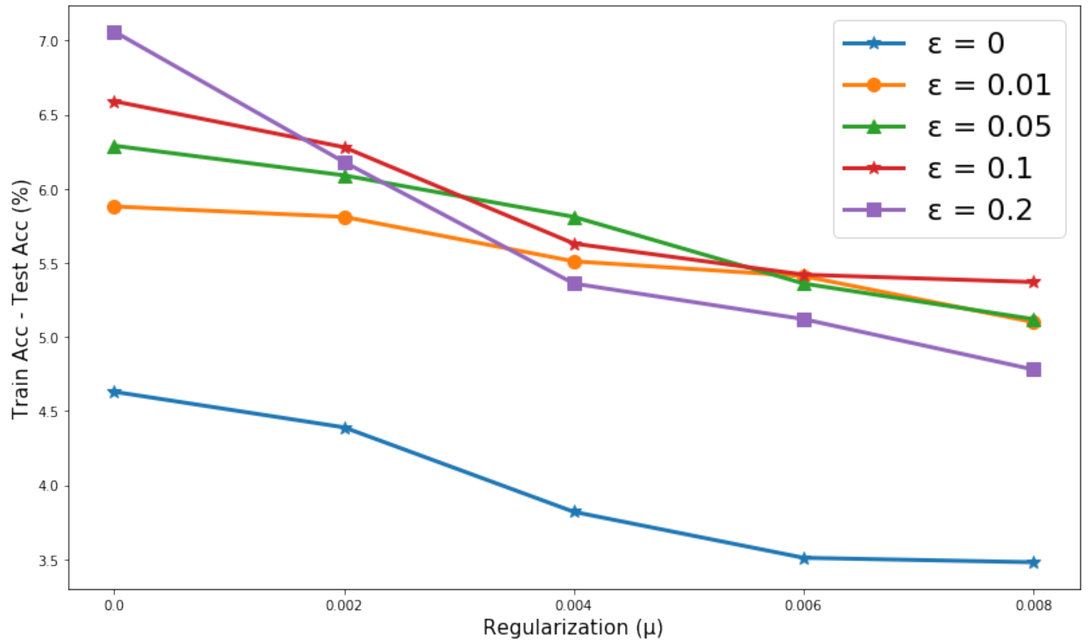
- Lastly, based in tandem with the robustness and the generalization behavior analysis, we present the robust and generalizable loss function

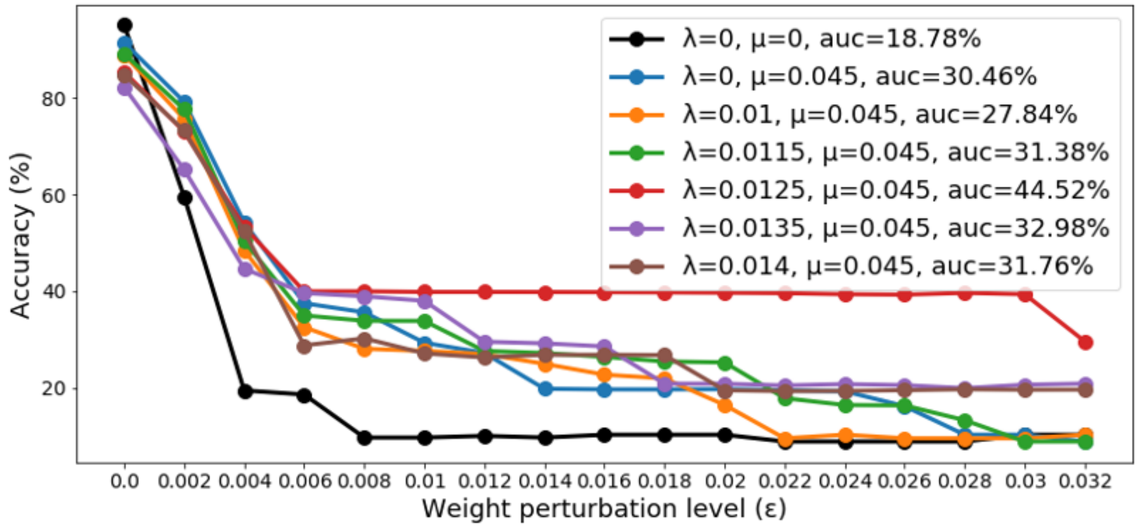
$$\ell^*(f_{\boldsymbol{W}}(\mathbf{x}), y) = \underbrace{\ell_{cls}(f_{\boldsymbol{W}}(\mathbf{x}), y)}_{\text{standard loss}} + \lambda \cdot \underbrace{\max_{y' \neq y} \{\eta^{y'y}_{\boldsymbol{W}}(\mathbf{x}|I)\}}_{\text{robustness loss from Thm. 3}} + \mu \cdot \underbrace{\sum_{m=1}^{L} \left(\left\| (\boldsymbol{W}^m)^T \right\|_{1,\infty} + \|\boldsymbol{W}^m\|_{1,\infty} \right)}_{\text{generalization gap regularization from Thm. 4}}$$

- Specifically, we can decompose it into three terms. Standard loss stands for accuracy-improving term, robustness loss represents robust training and lastly the third term controls the overall generalization gap.

Empirical Performance

- We provide two experiments upon validation (and more in appendices) with one being the validation of our finding in Theorem 4 and another one corresponds to the efficacy of out proposed loss function against weight perturbation.





Empirical Performance — Continued

- Here we also provide two additional experiments related to extension of this content with one being the comparison of different regularization technique and the efficacy of our loss function on convolutional models.

Generalization Gap (%) / Test accuracy (with $\epsilon = 0.01$)

Coefficient μ	**0**	**0.002**	**0.004**	**0.006**	**0.008**
L_1	6.36% / 84.2%	6.09% / 84.8%	Failed to Converge	Failed to Converge	Failed to Converge
L_2	6.26% / 84.3%	6.1% / 83.2%	5.78% / 84.5%	5.6% / 78.8%	5.5% / 78.5%
our loss	5.88% / 85.8%	5.81% / 85%	5.51% / 87.8%	5.41% / 83.4%	5.1% / 85.1%

Convolutional Model Under PGD Weight Attack

Perturbation Radius ϵ	**0**	**0.001**	**0.0015**	**0.0017**	**0.0019**	**0.0021**	**0.0023**	**0.0025**
$\lambda = 0, \mu = 0$	58.73%	13.08%	10.15%	10.01%	10%	10%	10%	10%
$\lambda = 3.2 * 10^{-4}, \mu = 10^{-4}$	51.1%	33.9%	23.16%	19.71%	16.85%	14.25%	12.52%	11.58%
$\lambda = 4.5 * 10^{-4}, \mu = 10^{-4}$	47.28%	36.82%	27.78%	26.19%	21.6%	19.35%	17.84%	16.53%
$\lambda = 5 * 10^{-4}, \mu = 10^{-4}$	42.71%	38.68%	30.2%	22.09%	19.41%	17.32%	15.68%	11.5%
$\lambda = 5.5 * 10^{-4}, \mu = 10^{-4}$	42.79%	32.9%	25.3%	22.77%	20.78%	18.88%	18.26%	17.1%
$\lambda = 6 * 10^{-4}, \mu = 10^{-4}$	38.17%	30.21%	24.55%	23.25%	21.97%	20.68%	19.14%	17.55%

Difference Comparison

Difference between ours and related studies

- We here provide a brief table summarizing the difference between ours work on robustness against weight perturbation and other similar works.

	Generalization Settings	Types of Robustness	Measure of Generalization Gap	Methods of Evaluating Maximum(Worst Case Loss)
Ours	Robust Setting	Against **Weight**	$\mathbb{P}[\exists(w+\epsilon) s.t. \mathbb{1}(y \neq \arg\max[f_{w+\epsilon}(x)]_{y'})] \leq \frac{1}{n}\sum \mathbb{1}(M(f_w(x), y) - \psi(f_w(x) \leq \gamma) + \text{Gap}$ [See Theorem 4]	Layer Propagation (Exact Bound)
Adversarial Weight Perturbation (2020)	Robust Setting	Against Input	$L_{D,\epsilon}(w+\epsilon) \leq L_{S,\epsilon}(w+\epsilon) + \text{Gap}$ [See equation (12)]	Maximum value based on generated adversarial inputs (inexact bound)
Sharpness-Aware Minimization (2021)	Standard Setting	NA	$L_D(w) \leq \max_\epsilon L_S(w+\epsilon) + \text{Gap}$ [See Appendix A.1]	First-Order Taylor Expansion (approximation; inexact bound)

Conclusion

Thanks for your listening!