OPT2023: 15th Annual Workshop on Optimization for Machine Learning

# Optimization dependent generalization bound for ReLU networks based on sensitivity in the tangent bundle

**Dániel Rácz** RACZ.DANIEL@SZTAKI.HU
*Institute for Computer Science and Control (SZTAKI) & Eötvös Loránd University, Budapest, Hungary*

**Mihály Petreczky** MIHALY.PETRECSZKY@UNIV-LILLE.FR
*CNRS CRIStAL, Université de Lille, Lille, France*

**András Csertán** CSERTAN.ANDRAS@SZTAKI.HU
*Institute for Computer Science and Control (SZTAKI) & Eötvös Loránd University, Budapest, Hungary*

**Bálint Daróczy** DAROCZYB@ILAB.SZTAKI.HU
*Institute for Computer Science and Control (SZTAKI), Budapest, Hungary & Széchenyi István University, Győr, Hungary*

## Abstract

Recent advances in deep learning have given us some very promising results on the generalization ability of deep neural networks, however literature still lacks a comprehensive theory explaining why heavily over-parametrized models are able to generalize well while fitting the training data. In this paper we propose a PAC type bound on the generalization error of feedforward ReLU networks via estimating the Rademacher complexity of the set of networks available from an initial parameter vector via gradient descent. The key idea is to bound the sensitivity of the network's gradient to perturbation of the input data along the optimization trajectory. The obtained bound does not explicitly depend on the depth of the network. Our results are experimentally verified on the MNIST and CIFAR-10 datasets.

## 1. Introduction and related work

Deep learning has started soaring in popularity during the last decade and by today it has reached unprecedented heights in terms of practical usage as well as theoretical research. As higher computational capacity is also becoming easier to access, the complexity and size of deep neural networks being used by the community is also dramatically increasing. The number of parameters contained in such networks are usually much higher than the number of training data points or the dimension of the training data. As a result, these models tend to easily interpolate the training data and according to the "classical" theory of machine learning (see e.g. [32]) they should lead to overfitting. However, it has been shown [34] empirically that this is not the case, i.e. highly over-parametrized models trained by gradient descent are capable of generalizing well while fitting the training data. This phenomenon drove the theoretical research of deep learning towards examining over-parametrized models. A big push has been given by the discovery of the Neural Tangent Kernel (NTK, [12]), leading to several convergence theorems in the infinitely wide regime [1, 8, 18] and shedding light on the connection of over-parametrized networks and kernel machines (see [3] for a comprehensive overview). While some interesting related results [4, 16, 25, 27, 31] exist, it is not entirely clear how the properties of the NTK might explain generalization.

Deep networks are almost always trained using some form of the gradient descent algorithm which seems to contain a so-called implicit bias in case of over-parametrized neural networks, i.e. finding well generalizing optima despite interpolating the training data. The conditions and properties of implicit bias is still under heavy research [33]. In [10] the authors show that under special conditions on the data, gradient descent finds a solution which generalizes well, but the resulting model is sensitive to adversarial perturbations. Analyzing the trajectories of gradient descent has also been present in the literature [22, 35].

One of the standard methods to obtain generalization bounds for models in statistical learning is the Probably Approximately Correct (PAC) framework [21], which was applied to deep networks in [14] and later in [9, 23]. These papers establish PAC-Bayesian bounds on the generalization error based on the estimation of the KL divergence of the predictor w.r.t. some prior distribution of the parameters. This was further developed in [24] resulting in a bound depending on the norm of the weights and (implicitly) on the depth of the network. PAC bounds on the generalization error are closely connected to bounding the Rademacher complexity. In [9] and [2] the authors also exploited some bounds on the Rademacher complexity of the underlying family of functions represented by ReLU networks. Several other bounds on the Rademacher complexity was derived in [11] depending on yet again the norm of the weight matrices and the width and depth of the network. In [30] a bound on the Rademacher complexity is achieved by defining the learning problem in a Reproducing Kernel Banach Space (RKBS) under some conditions on the learning algorithm. Generalization bounds for deep convolutional networks and Graph Neural Networks have been established in [17] and [20], respectively.

### 1.1. Our contribution

Informally, our main result is an upper bound on the generalization error of feedforward ReLU networks trained with gradient descent under certain conditions, depending on the optimization path. Before starting our journey to precisely state our theorem, we give a brief overview of the most important ingredients of our concept in order to make the rest of the article more traceable.

- We will make use of a classical PAC inequality for the generalization error which depends on the Rademacher complexity of the loss values on some (test) sample.
- The key step is to upper bound this Rademacher complexity of the family of functions represented by ReLU networks by examining the network's behavior along the optimization trajectory from the perspective of sensitivity to perturbation of the input data.
- The basic idea behind this sensitivity measure is the following. We look at the gradient of the network function w.r.t. the parameters as a feature map defined on the input data. What happens to this feature representation if the input data is perturbed by some Gaussian noise?
- We reinforce our theoretical bound by performing experiments on the MNIST [15] and CIFAR-10 [13] datasets. [1]

The idea of measuring the change in the network's gradients caused by Gaussian perturbation of the input data has been introduced in [6], where it was empirically shown that in case of the task of classification, the resulting measure correlates with the generalization gap of the network and can be

1. The implementation is available at https://github.com/danielracz/tansens_public

used to estimate the test loss without making use of the labels of the test data. However, previously there was no theoretical connection known to us explaining this phenomenon.

Exploiting the representation of the data via the gradient of the network as a feature map is one of the underlying ideas of the Neural Tangent Kernel and has been seen before (e.g. [5, 19]). In [26] such representation is used to induce a similarity function on the data. While the gradient is usually constant over training [19] under heavy over-parametrization, we suspect it has a vital connection to the generalization ability of the network in both the finite and infinite case.

## 2. Problem setup

### 2.1. Notations

Let us consider the framework of Empirical Risk Minimization over the task of binary classification, i.e. we are given a finite set of training data $D = \{(\mathbf{x}_i, y_i); i = \{1, \dots, n\}\}$ drawn from a probability distribution $\mathcal{X} \times \mathcal{Y}$ on $\mathbb{R}^{n_{in}} \times \{-1, 1\}$. Let $\mathcal{L}^D_{emp}(f) = \frac{1}{n}\sum_{i=1}^{n} l(f(\mathbf{x}_i), y_i)$ denote the empirical loss we want to minimize defined over a class of functions $\mathcal{F}$. We denote the true error by $\mathcal{L}(f) = \mathbf{E}_{(\mathbf{x},y)\sim\mathcal{X}\times\mathcal{Y}}[l(f(\mathbf{x}), y]$. The generalization error or gap of a model $f \in \mathcal{F}$ is defined as $|\mathcal{L}^D_{emp}(f) - \mathcal{L}(f)|$. In practice, we can approximate the generalization gap by the empirical generalization gap, i.e. the loss difference on the training data and some test data (see [7]).

Let the function class $\mathcal{F}$ we would like to optimize over be a family of ReLU networks characterized by a parameter vector $\theta \in \mathbb{R}^P$. We will treat such networks as a function of both the input data and the parameter vectors denoted by $f : \mathbb{R}^P \times \mathbb{R}^{n_{in}} \to \mathbb{R}^{n_{out}}$, where $P$ is the number of parameters and $n_{in}$ is the input dimension. As we are dealing with the binary classification task we have $n_{out} = 1$. Such models are usually trained by using the gradient descent algorithm from the initial point $\theta_0$ defined recursively at time $T$ as $\theta_T = \theta_{T-1} - \eta(T)\nabla_\theta \mathcal{L}^D_{emp}(f(\theta_{T-1}, \cdot))$, where $\eta(T) \in \mathbb{R}^+$ is the learning rate at time $T$. We follow the convention of $\text{ReLU}'(0) = 0$. For a fixed choice of the learning rate function $\eta : \mathbb{N} \to \mathbb{R}^+$ we say $\theta = GD(\theta_0, \eta, T)$, if $\theta$ is the output of the gradient descent after $T$ steps initialized in $\theta_0$ and run with the choice of $\eta$ as the learning rate. Let $Traj(\theta_0)$ denote the set of parameter vectors $\theta$ for which there exists $\eta$ and $T$ such that $\theta = GD(\theta_0, \eta, T)$.

### 2.2. Tangent Sensitivity

The central definition of our paper is called Tangent Sensitivity. Initially it was defined in [6] motivated by the following. Consider a small enough Gaussian perturbation around $\mathbf{x} \sim \mathcal{X}$ with $\phi(\mathbf{x}) = \mathbf{x} + \delta(\mathbf{x})$ where $\delta(\mathbf{x}) \sim \mathcal{N}(0, \sigma\mathbf{I})$ is a random variable. For the expected change in the gradient mapping defined as $\mathbf{x} \to \nabla_\theta f(\theta, \mathbf{x})$ on the input space we have

$$\mathbf{E}_{\delta(\mathbf{x})}[\|\nabla_\theta f(\theta, \mathbf{x}) - \nabla_\theta f(\theta, \phi(\mathbf{x}))\|_2^2] \sim \mathbf{E}_{\delta(\mathbf{x})}\left[\left\|\frac{\partial \nabla_\theta f(\theta, \mathbf{x})}{\partial x}\delta(\mathbf{x})\right\|_2^2\right] \leq \sigma \left\|\frac{\partial \nabla_\theta f(\theta, \mathbf{x})}{\partial x}\right\|_2^2.$$

The first approximation is based on the Taylor expansion of the gradient mapping and scales with the variance $\sigma$ of the Gaussian noise. Hence the next definition.

**Definition 1** *The Tangent Sample Sensitivity of a feedforward network $f$ with output in $\mathbb{R}$ at input $\mathbf{x} \in \mathbb{R}^{n_{in}}$ is a $P \times n_{in}$ dimensional matrix, $S(\theta, \mathbf{x}) := \frac{\partial \nabla_\theta f}{\partial x}(\theta, \mathbf{x}) = \frac{\partial^2 f}{\partial x \partial \theta}(\theta, \mathbf{x})$. The Tangent Sensitivity is the expectation of tangent sample sensitivity, i.e. $S(\theta) := \mathbf{E}_{\mathbf{x}\sim\mathcal{X}}[S(\theta, \mathbf{x})]$.*

Among other interesting properties it is empirically shown in [6] that the Frobenius norm of the Tangent Sensitivity matrix has a close relationship to the generalization error of the network. Theoretical explanation of this phenomenon has been unknown to us in the literature, thus to address this experience, we will establish a PAC bound on the generalization gap in which the norm of the Tangent Sensitivity appears.
During the rest of the paper we will abuse the naming convention and shorten Tangent Sample Sensitivity to Tangent Sensitivity in some cases.

## 3. PAC bound

Our goal now is to state our main theorem. First we need to introduce a series of assumptions.

**Assumption 2** *The loss function $l$ has the form $l(f(\mathbf{x}), y) = \ell(f(\mathbf{x}) - y)$, where $\ell$ is $K_{\mathcal{L}}$-Lipschitz.*

This is a mild assumption as most of the standard loss functions are Lipschitz on a bounded domain.

**Assumption 3** *For a fixed $\theta_0 \in \mathbb{R}^P$ and $\varepsilon > 0$ let $U_{\theta_0,\varepsilon} = Traj(\theta_0) \cap B_\varepsilon(\theta_0)$, where $B_\varepsilon(\theta_0)$ is the $\varepsilon$-ball around $\theta_0$. We assume that for any $\theta \in U_{\theta_0,\varepsilon}$ the Frobenius norm of the Tangent Sensitivity is bounded on the training set, i.e. $\sup_{\mathbf{x}\sim D} \|S(\theta, \mathbf{x})\|_F \leq C_{TS}$ for some $C_{TS} > 0$.*

Empirical evidence suggests that this is a reasonable assumption around an initialization point $\theta_0$, which in practice usually contains an optimum. In light of Assumption 3 we define the family of models $\mathcal{F}_{\theta_0,C,\varepsilon} := \left\{ f(\theta, \cdot) \;\middle|\; \theta \in U_{\theta_0,\varepsilon}\ ,\ \sup_{\mathbf{x}\sim D} \|S(\theta, \mathbf{x})\|_F \leq C \right\}$.

**Assumption 4** *We assume the following upper bounds hold: $\sup_{x\sim\mathcal{X}} |f(\theta_0, \cdot)| \leq K_{\theta_0}$, $\sup_{x\sim\mathcal{X}} \|\nabla_\theta f(\theta_0, \mathbf{x})\| \leq K_{\nabla_0}$ and $\sup_{x\sim\mathcal{X}} \|\mathbf{x}\| \leq K_x$.*

Note, that the these assumptions are widely applied in the literature. The first two refer to the boundedness of the function and its gradient around the initialization, while the third one assumes the input is bounded.

**Theorem 5** *Consider the problem setup from Section 2 and let Assumption 2 - 4 hold for a fixed initialization $\theta_0$ and $\varepsilon > 0$. Furthermore let us suppose for all $T \in \mathbb{N}$ that along all gradient descent trajectories of length $T$ starting from $\theta_0$ the quantity $\sum_{t=1}^{T-1} \eta(t) \frac{1}{n} \sum_{i=1}^{n} \frac{\partial l}{\partial f}(\theta_{t-1}, \mathbf{x}_i)$ is upper bounded by a positive constant $C_{GD}$. Then for any $\delta \in ]0, 1[$ with probability at least $1 - \delta$ over the random sample $S$ we have*

$$\forall f(\theta, \cdot) \in \mathcal{F}_{\theta_0,C_{TS},\varepsilon} : \mathcal{L}(f) - \mathcal{L}^S_{emp}(f) \leq K_{\mathcal{L}} K_{\theta_0} + \frac{C_1}{\sqrt{N}} + K_{\mathcal{L}} H(\theta) + B\sqrt{\frac{2log(\frac{4}{\delta})}{N}},$$

*where $C_1 = 2K_{\mathcal{L}} K_x K_{\nabla_0} C_{TS} C_{GD}$ and $H(\theta)$ is an error term and $N$ denotes the size of $S$. The constant $B$ is an upper bound on the loss $l(\cdot, \cdot)$. Additionally, along with a properly scaled ReLU activation if the width of the network tends to infinity, $H(\theta)$ tends to zero.*

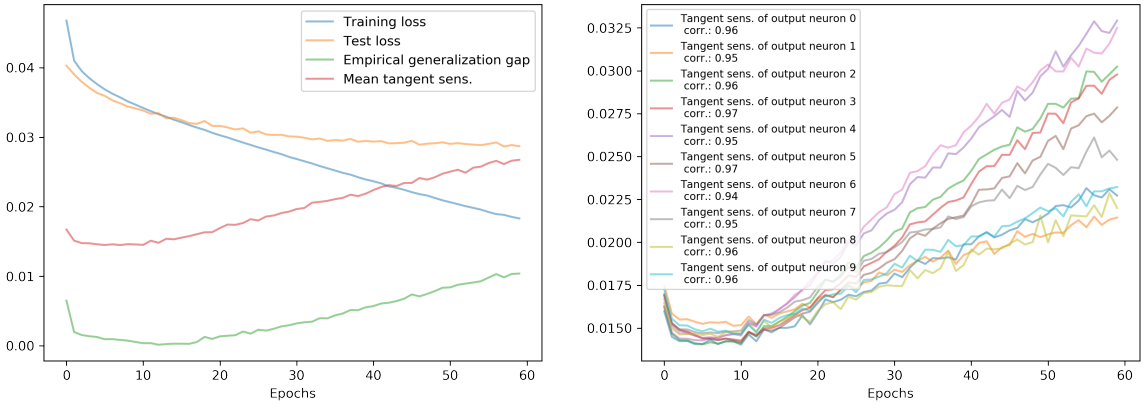


Figure 1: Pearson correlation[3] between the empirical generalization gap and the average norm of the Tangent Sensitivity on the test dataset for a $3 \times 3000$-wide fully connected ReLU net trained on CIFAR-10. The y-axes are based on the actual loss values. The norm values of tangent sensitivity were linearly scaled for presentational purposes. For more details and more experiments see Appendix C.

There are two critical terms in the bound of Theorem 5, namely $C_{TS}C_{GD}$ and $H(\theta)$. The former one highlights the connection of the generalization ability of the network and the Tangent Sensitivity along the optimization trajectory and it originates from the estimation of the Rademacher complexity (Definition 6 in Appendix A) of the network on the sample $S$. While currently we do not have a satisfying theoretical guarantee on the value of $C_{TS}$, we have strong empirical evidence that the norm of the Tangent Sensitivity is indeed correlated to the empirical generalization gap, see Fig. 1. For more details on our experiments see Appendix C.

The second term comes from the well known Taylor-approximation of the network and it is proportional to the norm of the Hessian of the network function. The main intuition behind the proof lies in the following possible approximation of a ReLU network. If $\theta = \theta_T$, then

$$f(\theta_T, \mathbf{x}) = f(\theta_0, \mathbf{x}) - \nabla_\theta f(\theta_0, \mathbf{x})^T \left( \sum_{t=1}^{T-1} \eta(t) \frac{1}{n} \sum_{i=1}^{n} \frac{\partial l}{\partial f} S(\theta_{t-1}, \mathbf{x}_i) \mathbf{x}_i \right) + h(\theta_T, \mathbf{x}),$$

where $h$ is an error term (see Appendix B) which determines the term $H(\theta)$ in Theorem 5. Let $w_\theta = \sum_{t=1}^{T-1} \eta(t) \frac{1}{n} \sum_{i=1}^{n} \frac{\partial l}{\partial f} S(\theta_{t-1}, \mathbf{x}_i) \mathbf{x}_i$. Because $w_\theta$ depends only on the training set and the optimization path, but not the actual input $\mathbf{x}$, we can approximate a ReLU network by the scalar product $< w_\theta, \nabla_\theta f(\theta_0, \mathbf{x}) >$ and bound the norm of $w_\theta$ thanks to the various assumptions. Finally, we can apply the standard techiniques for bounding the Rademacher complexity of linear classifiers. For a complete proof of Theorem 5 see Appendix B.

3. https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.pearsonr.html

## 4. Discussion and future work

In this paper we have established a PAC bound on the generalization error of feedforward ReLU networks, crucially depending on the Tangent Sensitivity along the optimization trajectory. Empirical evidence has previously shown the correlation between the two quantities, we believe the obtained bound provides a strong theoretical justification. While the established bound is not tight, we believe that the sensitivity measure might have a connection to the smoothness of the function in some appropriate function space [3], which is a promising direction for the generalization theory of deep networks. The straightforward next step seems to be the convergence analysis of the Tangent Sensitivity matrix and its norm around the initialization and optima. An interesting idea would be to incorporate it in the loss function, as presented in [28], however calculating the Tangent Sensitivity norm is computational expensive, thus it requires to find an efficient approximation. In the infinite width limit, Tangent Sensitivity can be viewed as a partial derivative-like object of the NTK w.r.t. the input data. It would be interesting to examine the connection to the generalization ability of the NTK Kernel Machine.

## 5. Acknowledgement

This research was supported by the European Union project RRF-2.3.1-21-2022-00004 within the framework of the Artificial Intelligence National Laboratory and by the C.N.R.S. E.A.I. project "Stabilité des algorithmes d'apprentissage pour les réseaux de neurones profonds et récurrents en utilisant la géométrie et la théorie du contrôle via la compréhension du rôle de la surparamétrisation". B.D. was supported by MTA Premium Postdoctoral Grant 2018.

## References


[1] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International conference on machine learning*, pages 242–252. PMLR, 2019.

[2] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pages 6240–6249, 2017.

[3] Mikhail Belkin. Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation. *Acta Numerica*, 30:203–248, 2021.

[4] Benjamin Bowman and Guido F Montufar. Spectral bias outside the training set for deep networks in the kernel regime. *Advances in Neural Information Processing Systems*, 35:30362–30377, 2022.

[5] Yuan Cao and Quanquan Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *Advances in Neural Information Processing Systems*, pages 10835–10845, 2019.

[6] Bálint Daróczy. Gaussian perturbations in relu networks and the arrangement of activation regions. *Mathematics*, 10(7):1123, 2022.

[7] Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*, volume 31. Springer Science & Business Media, 2013.

[8] Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

[9] Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

[10] Spencer Frei, Gal Vardi, Peter L Bartlett, and Nathan Srebro. The double-edged sword of implicit bias: Generalization vs. robustness in relu networks. *arXiv preprint arXiv:2303.01456*, 2023.

[11] Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. In *Conference On Learning Theory*, pages 297–299. PMLR, 2018.

[12] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580, 2018.

[13] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, MIT and NYU, 2009.

[14] John Langford and Rich Caruana. (not) bounding the true error. *Advances in Neural Information Processing Systems*, 14, 2001.

[15] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[16] Yicheng Li, Zixiong Yu, Guhan Chen, and Qian Lin. Statistical optimality of deep wide neural networks. *arXiv preprint arXiv:2305.02657*, 2023.

[17] Renjie Liao, Raquel Urtasun, and Richard Zemel. A pac-bayesian approach to generalization bounds for graph neural networks. *arXiv preprint arXiv:2012.07690*, 2020.

[18] Chaoyue Liu, Libin Zhu, and Mikhail Belkin. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Applied and Computational Harmonic Analysis*, 59:85–116, 2022.

[19] Chaoyue Liu, Amirhesam Abedsoltan, and Mikhail Belkin. On emergence of clean-priority learning in early stopped neural networks. *arXiv preprint arXiv:2306.02533*, 2023.

[20] Philip M Long and Hanie Sedghi. Generalization bounds for deep convolutional neural networks. *arXiv preprint arXiv:1905.12600*, 2019.

[21] David A McAllester. Pac-bayesian model averaging. In *Proceedings of the twelfth annual conference on Computational learning theory*, pages 164–170, 1999.

[22] Behnam Neyshabur, Russ R Salakhutdinov, and Nati Srebro. Path-sgd: Path-normalized optimization in deep neural networks. In *NIPS'15*, 2015.

[23] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, pages 5947–5956, 2017.

[24] Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017.

[25] Samet Oymak, Zalan Fabian, Mingchen Li, and Mahdi Soltanolkotabi. Generalization guarantees for neural networks via harnessing the low-rank structure of the jacobian. *arXiv preprint arXiv:1906.05392*, 2019.

[26] Dániel Rácz and Balint Zoltan Daroczy. Gradient representations in relu networks as similarity functions. In *ESANN 2021-European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2021.

[27] Adityanarayanan Radhakrishnan, Mikhail Belkin, and Caroline Uhler. Wide and deep neural networks achieve optimality for classification. *arXiv preprint arXiv:2204.14126*, 2022.

[28] Omar Rivasplata, Vikram M Tankasali, and Csaba Szepesvari. Pac-bayes with backprop. *arXiv preprint arXiv:1908.07380*, 2019.

[29] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[30] Alistair Shilton, Sunil Gupta, Santu Rana, and Svetha Venkatesh. Gradient descent in neural networks as sequential learning in rkbs. *arXiv preprint arXiv:2302.00205*, 2023.

[31] Sattar Vakili, Michael Bromberg, Jezabel Garcia, Da-shan Shiu, and Alberto Bernacchia. Uniform generalization bounds for overparameterized neural networks. *arXiv preprint arXiv:2109.06099*, 2021.

[32] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999.

[33] Gal Vardi. On the implicit bias in deep-learning algorithms. *Communications of the ACM*, 66 (6):86–93, 2023.

[34] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

[35] Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Gradient descent optimizes over-parameterized deep relu networks. *Machine learning*, 109:467–492, 2020.

## Appendix A. Rademacher complexity

**Definition 6** *(e.g. see definition 26.1 in [29]) The Rademacher complexity of a bounded set* $\mathcal{A} \subset \mathbb{R}^m$ *of vectors is defined as* $R(\mathcal{A}) = \mathbb{E}_\sigma \left[ \sup_{a \in \mathcal{A}} \frac{1}{m} \sum_{i=1}^{m} \sigma_i a_i \right]$*, where the random variables* $\sigma_i$ *are i.i.d such that* $\mathbb{P}[\sigma_i = 1] = \mathbb{P}[\sigma = -1] = 0.5$*. The Rademacher complexity of a set of functions* $\mathcal{F}$ *over a set of samples* $S = \{s_1, \ldots, s_m\}$ *is defined as* $R_S(\mathcal{F}) = R(\{(f(s_1), \ldots, f(s_m)) | f \in \mathcal{F}\})$.

## Appendix B. Proof of Theorem 5

In order to prove Theorem 5 we will apply the following general PAC bound to our situation.

**Theorem 7 (e.g. see Theorem 26.5 in [29])** *Let* $\mathcal{F}$ *be a compact set of hypotheses. For any* $\delta \in ]0, 1[$

$$\mathbb{P}_{\mathcal{S}} \left( \forall f \in \mathcal{F} : \mathcal{L}(f) - \mathcal{L}_{emp}(f) \leq 2R_S(L_0) + B\sqrt{\frac{2log(\frac{4}{\delta})}{N}} \right) \geq 1 - \delta,$$

*where* $R_S(L_0)$ *is the Rademacher complexity of the set* $L_0 := \{(l(f(\mathbf{x}_1, y_1)), \ldots, l(f(\mathbf{x}_N, y_N)) \mid f \in \mathcal{F})\}$*,* $B$ *is an upper bound on* $l(\,.\,,\,.\,)$ *and the probability is taken over the random samples* $S$ *of size* $N$*.*

Proof of Theorem 7 can be found in [29].
First we take the Taylor approximation of $f(\theta, \mathbf{x})$ around a random initialization point $\theta_0$.

$$f(\theta, \mathbf{x}) = f(\theta_0, \mathbf{x}) + \nabla_\theta f(\theta_0, \mathbf{x})^T(\theta - \theta_0) + h(\theta, \mathbf{x})$$

where the approximation error $h(\theta, \mathbf{x}) = O((\theta - \theta_0)^T H_\theta f(\theta, \mathbf{x})^T(\theta - \theta_0))$. We know for wide networks and a suitably scaled ReLU activation function the norm of the Hessian tends to zero as the width tends to infinity [18] in $B_\varepsilon(\theta_0)$, hence for a wide enough network the error term $h$ is sufficiently small. The radius $\varepsilon$ is proportional to the smallest eigenvalue of the NTK Gram matrix over the training data. Now consider the vanilla gradient descent for an $L_{\mathcal{L}}$-Lipschitz loss function of the form $\mathcal{L}^D_{emp}(\theta) = \frac{1}{n} \sum_{i=1}^{n} l(f(\theta, \mathbf{x}_i), y_i)$ for a fixed set of training data. Note, that $\nabla_\theta \mathcal{L}^D_{emp}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \frac{\partial l}{\partial f} \nabla_\theta f(\theta, \mathbf{x}_i)$. According to the GD update rule

$$\theta_T = \theta_{T-1} - \eta(T) \nabla_\theta \mathcal{L}^D_{emp}(\theta_{T-1})$$

where $\eta(T)$ is the learning rate at time $T$.

**Lemma 8** *For a biasless ReLU network* $\nabla_\theta f(\theta, \mathbf{x}) = S(\theta, \mathbf{x})\mathbf{x}$*.*

**Proof** Let us fix an input vector $\mathbf{x} \in \mathbb{R}^{n_{in}}$ and a parameter vector $\theta \in \mathbb{R}^P$. We say that a path $p$ in the network graph is active if all the nodes in the path are active, meaning that the preactivation of every node is positive. The notation $p : x_j \rightarrow$ means that $p$ is a path starting at the $j$-th input neuron. We do not specify the output neuron as we are in the binary classification setting, where $n_{out} = 1$.

Let $S_i$ be the $i$-th row of $S(\theta, \mathbf{x})$. Since we fixed the input and parameter vectors, we will omit the dependence of them in the notation of the derivatives (as we did in case of $S_i$).

The $i$-th coordinate of the LHS is

$$\frac{\partial f}{\partial \theta_i} = \sum_{\substack{x_j:\text{input} \\ \text{node}}} \sum_{\substack{p:x_j\to \\ \text{active path}, \\ \theta_i\in p}} \mathbf{x}_j \prod_{\substack{\theta_p\in p \\ \theta_p\neq\theta_i}} \theta_p = \sum_{\substack{x_j:\text{input} \\ \text{node}}} \Big( \sum_{\substack{p:x_j\to \\ \text{active path}, \\ \theta_i\in p}} \prod_{\substack{\theta_p\in p \\ \theta_p\neq\theta_i}} \theta_p \Big) \mathbf{x}_j = S_i \mathbf{x}$$

■

Let us fix an initial parameter $\theta_0$ and a positive radius $\varepsilon$ and let $U_{\theta_0,\varepsilon} = Traj(\theta_0) \cap B_\varepsilon(\theta_0)$. For any $\theta \in U_{\theta_0,\varepsilon}$ there is a $T$ such that $\theta = GD(\theta_0, \eta, T)$ for some $\eta$. Under these circumstances we may emphasize the relationship between $\eta$, $\theta$ and $T$ by the notations $\theta = \theta_T$ and $\eta = \eta_{\theta_T}$. Using the GD update rule for such $\theta_T$ and the Taylor approximation of $f$ we obtain

$$\begin{aligned} f(\theta_T, \mathbf{x}) &= f(\theta_0, \mathbf{x}) - \nabla_\theta f(\theta_0, \mathbf{x})^T \left( \sum_{t=1}^{T-1} \eta(t) \nabla_\theta \mathcal{L}^D_{emp}(\theta_{t-1}) \right) + h(\theta_T, \mathbf{x}) \\ &= f(\theta_0, \mathbf{x}) - \nabla_\theta f(\theta_0, \mathbf{x})^T \left( \sum_{t=1}^{T-1} \eta(t) \frac{1}{n} \sum_{i=1}^{n} \frac{\partial l}{\partial f} \nabla_\theta f(\theta_{t-1}, \mathbf{x}_i) \right) + h(\theta_T, \mathbf{x}) \qquad (1) \\ &= f(\theta_0, \mathbf{x}) - \nabla_\theta f(\theta_0, \mathbf{x})^T \left( \sum_{t=1}^{T-1} \eta(t) \frac{1}{n} \sum_{i=1}^{n} \frac{\partial l}{\partial f} S(\theta_{t-1}, \mathbf{x}_i) \mathbf{x}_i \right) + h(\theta_T, \mathbf{x}). \end{aligned}$$

The last equality follows from Lemma 8. Note, that the first term and $\nabla_\theta f(\theta_0, \mathbf{x})$ depend only on $\theta_0$. In order to use Theorem 7 we need to upper bound the Rademacher complexity of $\left\{ (l(f(\theta, z_1)), \dots, l(f(\theta, z_N)))^T \middle| f(\theta, .) \in \mathcal{F}_{\theta_0, C_{TS}, \varepsilon} \right\}$ for which it is enough to upper bound $R_S(\left\{ (f(\theta, z_1), \dots, f(\theta, z_N))^T \middle| f(\theta, .) \in \mathcal{F}_{\theta_0, C_{TS}, \varepsilon} \right\})$ by Lemma 26.9 in [29] and Assumption 2. Here we denoted the elements of the random sample $S$ from Theroem 5 by $z_1, \dots, z_N$ and incorporated the labels into the loss function $l$. Using the notation $R_S(\mathcal{F}_{\theta_0, C_{TS}, \varepsilon}) = R_S(\left\{ (f(\theta, z_1), \dots, f(\theta, z_N))^T \middle| f(\theta, .) \in \mathcal{F}_{\theta_0, C_{TS}, \varepsilon} \right\})$, Theorem 5 immediately follows from the following proposition.

**Proposition 9** *Under Assumptions 2-4 and the additional assumptions stated in Theorem 5 we have*

$$\begin{aligned} R_S(\mathcal{F}_{\theta_0, C_{TS}, \varepsilon}) \leq K_{\mathcal{L}} R_S(\left\{ (f(\theta_0, z_1) \dots f(\theta_0, z_M))^T \right\}) + K_{\mathcal{L}} \frac{K_x K_{\nabla_0} C_{TS} C_{GD}}{\sqrt{N}} + \\ + K_{\mathcal{L}} R_S(\left\{ (h(\theta, z_1), \dots, h(\theta, z_M))^T \middle| f(\theta, .) \in \mathcal{F}_{\theta_0, C_{TS}, \varepsilon} \right\}). \end{aligned}$$

Before starting the proof we need another lemma.

**Lemma 10 (Lemma 26.10 from [29])** *Let $\mathbf{v}_1, \dots, \mathbf{v}_m$ be vectors in a Hilbert space. Define $S' = \{ (\langle \mathbf{w}, \mathbf{v}_1 \rangle, \dots, \langle \mathbf{w}, \mathbf{v}_m \rangle) \mid \|\mathbf{w}\|_2 \leq 1 \}$. Then,*

$$R(S') \leq \frac{\max_i \|\mathbf{v}_i\|_2}{\sqrt{m}}$$

Note that by omitting the condition $\|\mathbf{w}\|_2 \leq 1$ the lemma remains true with the modification of the upper bound multiplied by $\|\mathbf{w}\|_2$.

**Proof of Proposition 9**

We remind the reader that the training dataset is considered constant and is denoted by $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. The three terms in the RHS of Proposition 9 respectively originates from the three terms in equation 1 together with the fact that for a set of the form $T = \{A(\theta) + B(\theta) \mid \theta \in \Theta\}$ the Rademacher complexity is upper bounded by $R_S(T) \leq R_S(\{A(\theta) \mid \theta \in \Theta\}) + R_S(\{B(\theta) \mid \theta \in \Theta\})$.

The first and third terms are straightforward. The second term comes from the application of Lemma 10 along with the roles $\mathbf{v}_i = \nabla_\theta f(\theta_0, z_i)$ and $\mathbf{w}_\theta = \frac{1}{n} \sum_{t=1}^{T-1} \eta(t) \sum_{i=1}^{n} \frac{\partial l}{\partial f}(\theta_{t-1}, \mathbf{x}_i) \nabla_\theta f(\theta_{t-1}, \mathbf{x}_i)$. Assuming that the Tangent Sensitivity is bounded for every input along the optimization path by $C_{TS}$ (Assumption 3) and the input is also bounded by $K_x$ (Assumption 4), we have

$$\|\mathbf{w}_\theta\| \leq C_{TS} K_x \sum_{t=1}^{T-1} \eta(t) \frac{1}{n} \sum_{i=1}^{n} \frac{\partial l}{\partial f}(\theta_{t-1}, \mathbf{x}_i) \leq C_{TS} K_x C_{GD},$$

where we used the condition in Theorem 5. ■

The term $H(\theta)$ in the Theorem is any upper bound on the Rademacher complexity of $R_S(\{(h(\theta, z_1), \ldots, h(\theta, z_M))^T | f(\theta, .) \in \mathcal{F}_{\theta_0, C_{TS}, \varepsilon}\})$ . Additionally, we have $R_S(\{(f(\theta_0, z_1) \ldots f(\theta_0, z_M))^T\}) \leq K_{\theta_0}$ and $\|\mathbf{v}_i\| = \|\nabla_\theta f(\theta_0, z_i)\| \leq K_{\nabla_0}$, hence the Theorem follows.

## Appendix C. Experiments

We performed experiments to verify the correlation between the norm of Tangent Sensitivity and the empirical generalization gap. We considered the task of image classification on the well-known MNIST and CIFAR-10 datasets. In case of the MNIST problem we trained a fully connected ReLU network, a Multi-layer Percpetron (MLP) of the layer structure (784, 3000, 3000, 3000, 10). We optimized the network for squared loss with Adam optimizer. The resulted correlation between the empirical generalization gap and the average norm of the Tangent Sensitivity on the test dataset can be seen in Fig. 2. Note, as in Fig. 1 on all figures the norm values of tangent sensitivity were linearly scaled for presentational purposes. Typically, a division by a constant with magnitude $10^5$ was enough. The class gaps are computed by considering the $i$-th output neuron and the binary classificaiton problem for the $i$-th class.

As for the CIFAR-10 dataset, we trained similarly structured MLPs where the three hidden layers have widths of 100, 700, 1500 and 3000. Besides the results of 3000-width network in Fig. 1, the rest of the curves are shown in Fig. 3, Fig. 4, and Fig. 5.

Interestingly, in case of wide networks (3000-wide) the tangent sensitivity of different output neurons correlate with the mean generalization gap while if the network has lower width this correlation remains but only for matching per class output neuron and per class loss.

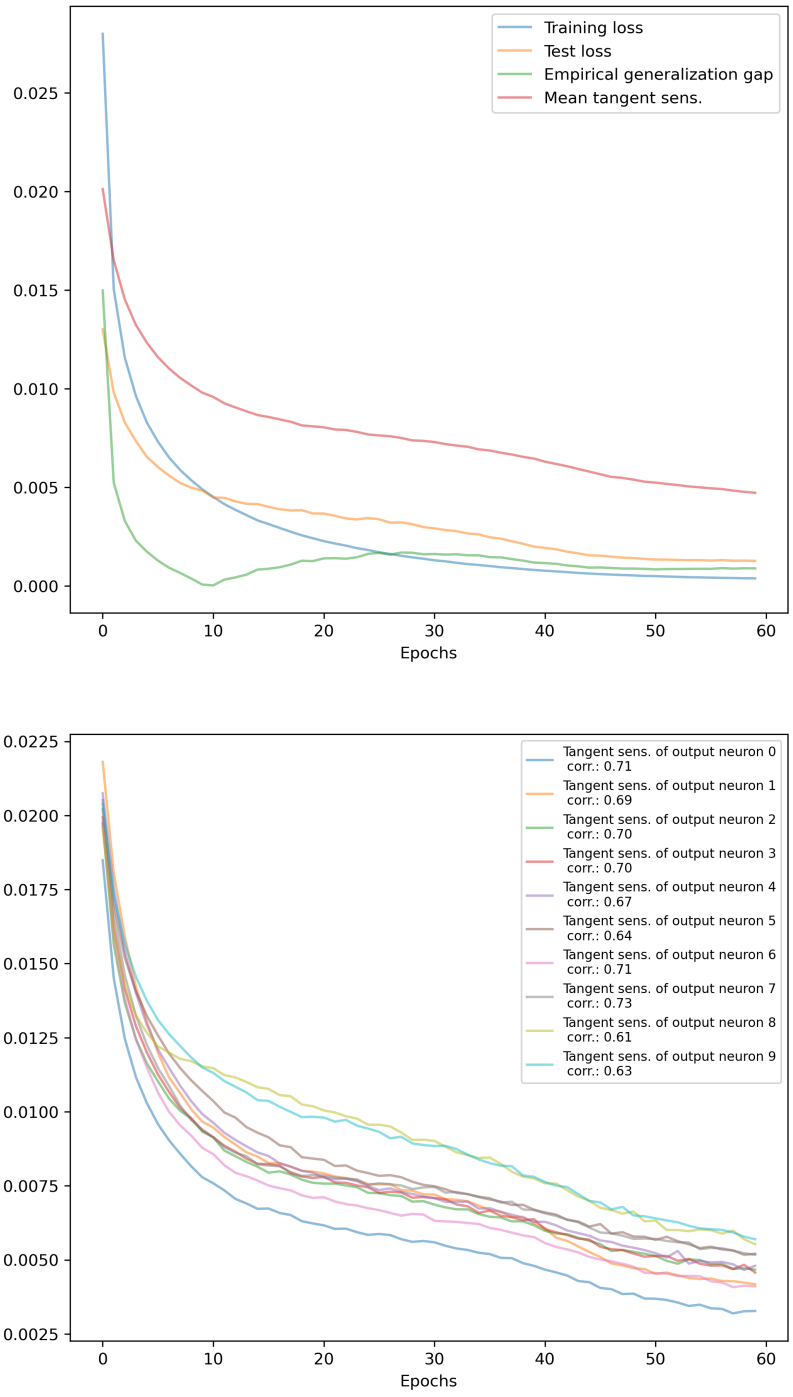


Figure 2: Correlation between the empirical generalization gap and the average norm of the Tangent Sensitivity on the test dataset for a $3 \times 3000$-wide fully connected ReLU trained on MNIST.

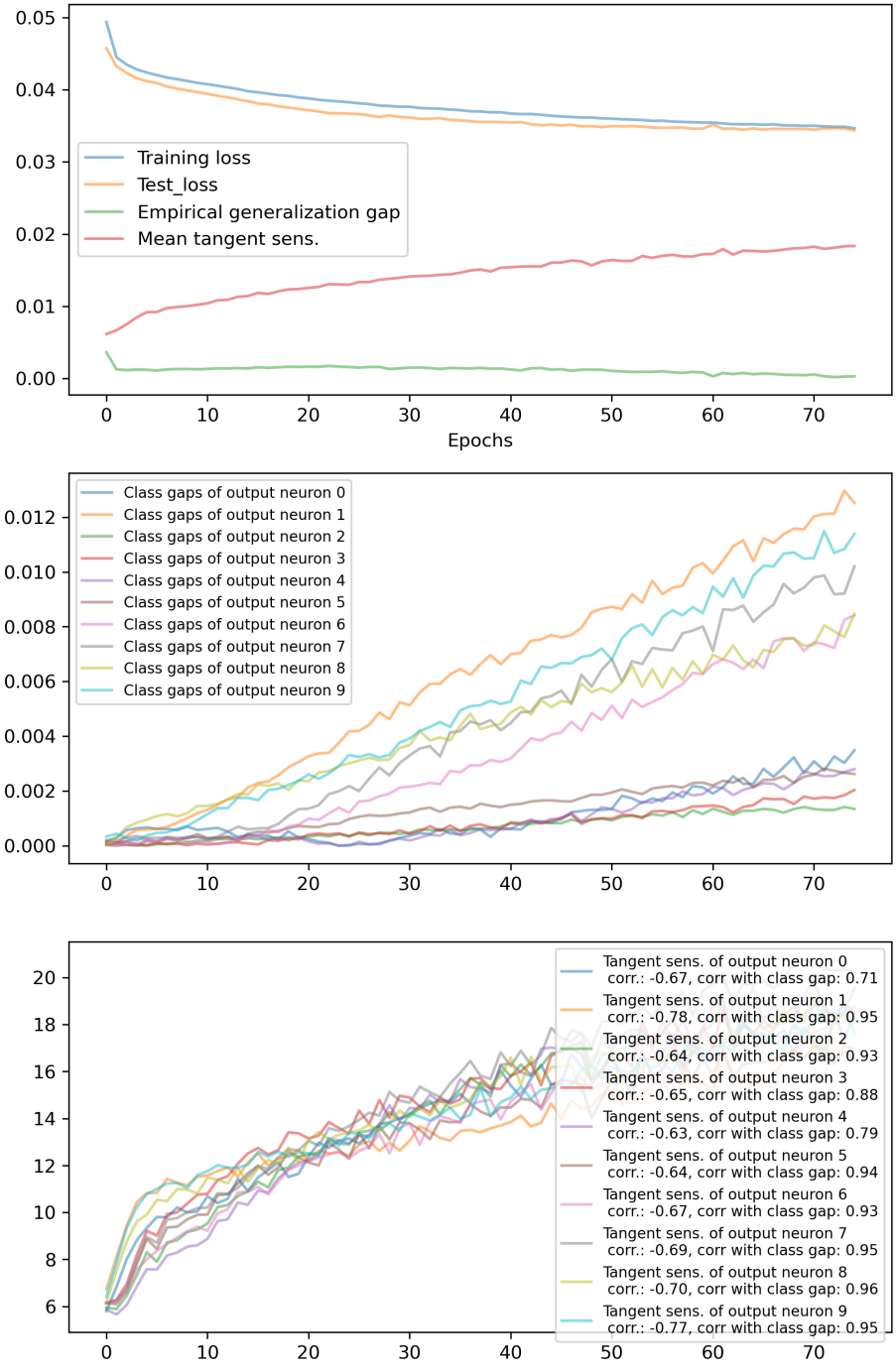


Figure 3: Correlation between the empirical generalization gap and the average norm of the Tangent Sensitivity on the test dataset for a $3 \times 100$-wide fully connected ReLU trained on CIFAR-10.

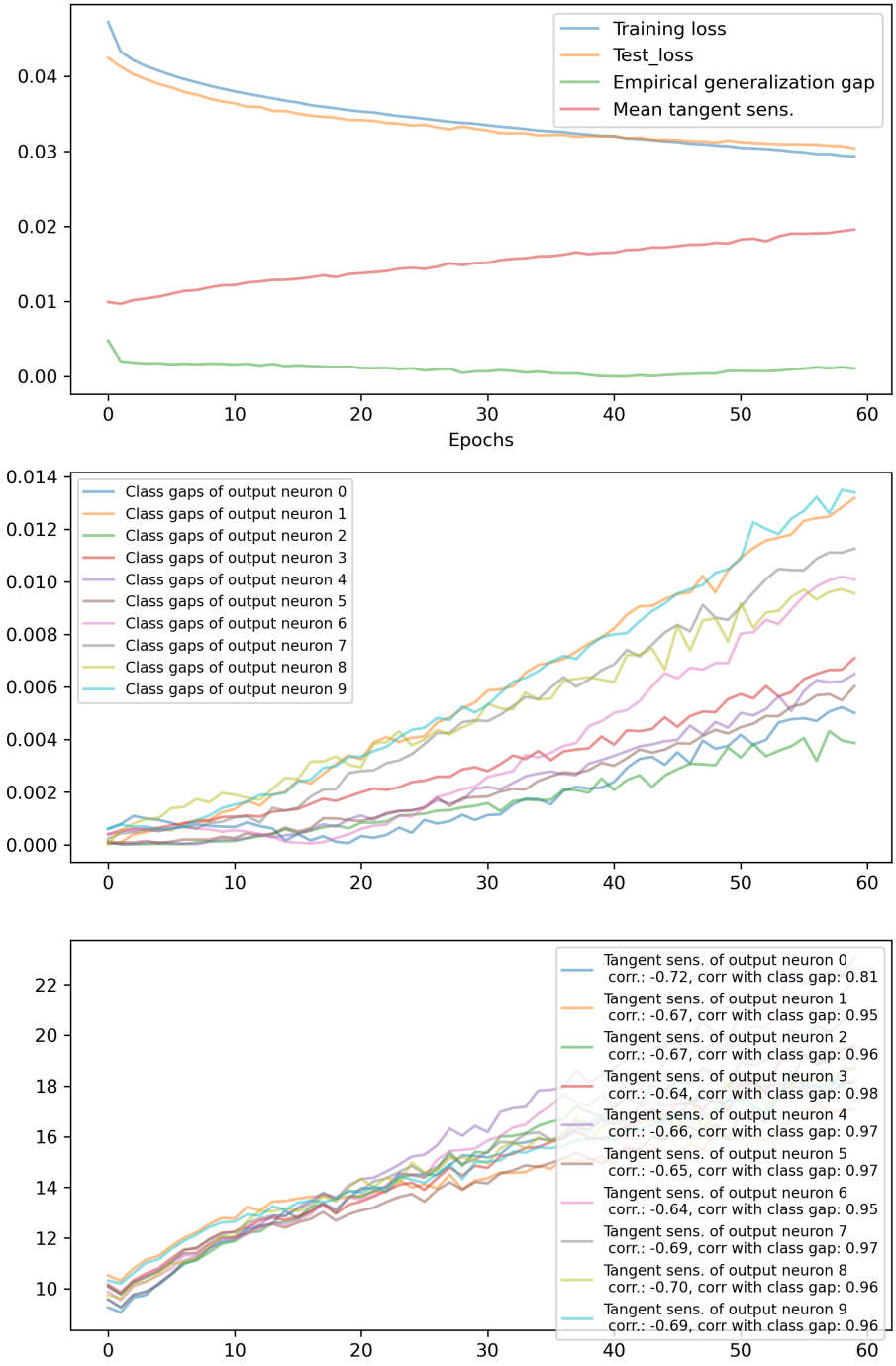


Figure 4: Correlation between the empirical generalization gap and the average norm of the Tangent Sensitivity on the test dataset for a $3\times700$-wide fully connected ReLU trained on CIFAR-10.

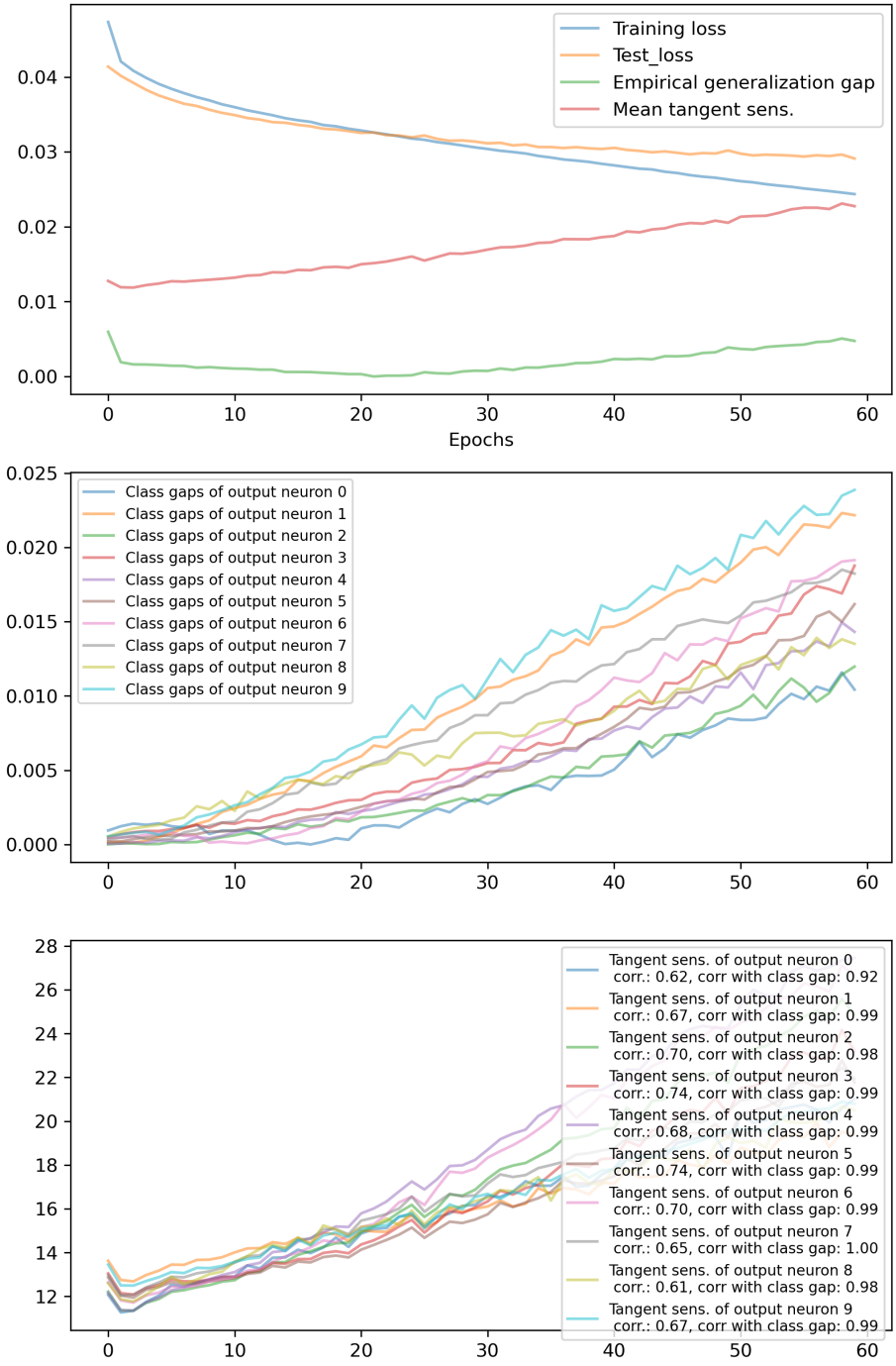


Figure 5: Correlation between the empirical generalization gap and the average norm of the Tangent Sensitivity on the test dataset for a $3 \times 1500$-wide fully connected ReLU trained on CIFAR-10.