Non-Vacuous Generalisation Bounds for Shallow Neural Networks

Felix Biggs, Benjamin Guedj

Non-Vacuous Generalisation Bounds for Shallow Neural Networks

Felix Biggs

Centre for Artificial Intelligence, Department of Computer Science
University College London and Inria London, United Kingdom
felbiggs@cs.ucl.ac.uk

Benjamin Guedj

Centre for Artificial Intelligence, Department of Computer Science
University College London and Inria London, United Kingdom
b.guedj@ucl.ac.uk

We focus on a specific class of shallow neural networks with a single hidden layer, namely those with L_2-normalised data and either a sigmoid-shaped Gaussian error function ("erf") activation or a Gaussian Error Linear Unit (GELU) activation. For these networks, we derive new generalisation bounds through the PAC-Bayesian theory; unlike most existing such bounds they apply to neural networks with deterministic rather than randomised parameters. Our bounds are empirically non-vacuous when the network is trained with vanilla stochastic gradient descent on MNIST and Fashion-MNIST.

1. Introduction

The study of generalisation properties of deep neural networks is arguably one of the topics gaining most traction in deep learning theory (see, *e.g.*, the recent surveys Kawaguchi et al., 2020; Jiang et al., 2020b). In particular, a characterisation of out-of-sample generalisation is essential to understand where trained neural networks are likely to succeed or to fail, as evidenced by the recent NeurIPS 2020 competition "Predicting Generalization in Deep Learning" (Jiang et al., 2020a). One stream of this joint effort, which the present paper contributes to, is dedicated to the study of shallow neural networks, potentially paving the way to insights on deeper architectures.

Despite numerous efforts in the past few years, non-vacuous generalisation bounds for neural networks with many more parameters than data remain generally elusive. Those few non-vacuous bounds that exist primarily report bounds for networks with randomised parameters, for example Gaussian weights, which are re-drawn for every prediction (a non-exhaustive list of references would begin with Dziugaite and Roy, 2017, 2018b; Neyshabur et al., 2017, 2018), or for compressed versions of the trained networks (Zhou et al., 2019). While these undoubtedly advanced knowledge on generalisation in deep learning theory, this is far from contemporary practice which generally focuses on deterministic networks obtained directly through stochastic gradient descent (SGD).

The PAC-Bayesian theory (we refer to the recent Guedj, 2019 and Alquier, 2021 for a gentle introduction) is thus far the only framework within which non-vacuous bounds have been provided for networks trained on common classification tasks. Given its focus on randomised or "Gibbs" predictors, the aforementioned lack of results for deterministic networks is unsurprising. However, the framework is not limited to such results: one area within PAC-Bayes where deterministic predictors are often considered lies in a range of results for the "majority vote", or the expected overall prediction of randomised predictors, which is itself deterministic.

Computing the average output of deep neural networks with randomised parameters is generally intractable: therefore most such works have focused on cases where the average output is simple to compute, as for example when considering linear predictors. Here, building on ideas from Biggs and Guedj (2021a), we show that provided our predictor structure factorises in a particular way, more complex majority votes can be constructed. In particular, we give formulations for randomised predictors whose majority vote can be expressed as a deterministic single-hidden-layer neural network. Through this, we obtain classification bounds for these deterministic predictors that are non-vacuous on the celebrated baselines MNIST (LeCun et al., 1998) and Fashion-MNIST (Xiao et al., 2017). We believe these are the first such results.

Our work fundamentally relates to the question: what kind of properties or structures in a trained network indicate likely generalisation to unseen data? It has been shown by Zhang et al. (2017) that neural networks trained by SGD can perfectly overfit large datasets with randomised labels, which would indicate a lack of capacity control, while simultaneously generalising well in a variety of scenarios. Thus, clearly any certification of generalisation must involve extracting additional information other than the train loss—for example, the specific final network chosen by SGD. How do the final parameters of a neural network trained on an "easy" data distribution as opposed to a pathological (*e.g.*, randomised label) one differ? A common answer to this has involved the return of capacity control and the norms of the weight matrices, often measured as a distance to the initialisation (as done, *e.g.*, in Dziugaite and Roy, 2017; Bartlett et al., 2017; Neyshabur et al., 2018).

We suggest, following insights from Dziugaite et al. (2020b), that a better answer lies in utilising the empirically-observed stability of SGD on easy datasets. We give bounds that are tightest when a secondary run of SGD on some subset of the training set gives final weights that are close to the full-dataset derived weights. This idea combines naturally in the PAC-Bayes framework with the requirement of perturbation-

robustness of the weights—related to the idea of flat-minima (Hinton and van Camp, 1993; Hochreiter and Schmidhuber, 1997)—to normalise the distances between the two runs. By leveraging this commonly-observed empirical form of stability we effectively incorporate information about the inherent easiness of the dataset and how adapted our neural network architecture is to it. Although it is a deep and interesting theoretical question as to when and why such stability occurs under SGD, we believe that by making the link to generalisation explicit we solve some of the puzzle.

Setting. We consider D-class classification on a set $\mathcal{X}$ with predictors returning values in $\hat{\mathcal{Y}} \subset \mathbb{R}^D$, label space $\mathcal{Y} = [D]$ and misclassification loss defined as $\ell(f(x), y) = \mathbf{1}\{\text{argmax}_{k\in[D]} f(x)[k] \neq y\}$; our predictors are "score-output" or vector-valued and their prediction is the argmaximum component. It is will prove useful that scaling does not enter into these losses and thus the outputs of classifiers can be arbitrarily re-scaled by $c > 0$ without affecting the predictions. We write $L(f) := \mathbb{E}_{(x,y)\sim\mathcal{D}}\ell(f(x), y)$ and $\hat{L}_S(f) := m^{-1}\sum_{(x,y)\in S}\ell(f(x), y)$ for the risk and empirical risk of the predictors with respect to data distribution $\mathcal{D}$ and i.i.d. m-sized sample $S \sim \mathcal{D}^m$.

Overview of our contributions. We derive generalisation bounds for a neural network $F_{U,V}$ with first and second layer weights U and V respectively taking the form

$$F_{U,V}(x) = V\,\phi\left(\beta\frac{Ux}{\|x\|_2}\right)$$

with ϕ being an element-wise activation. If the data is normalised to have $\|x\|_2 = \beta$ these are simply equivalent to one-hidden-layer neural networks with activation ϕ and the given data norm. We provide high-probability bounds on $L(F_{U,V})$ of the approximate form

$$2\mathbb{E}_{f\sim Q}\hat{L}_S(f) + \mathcal{O}\left(\frac{\beta\|U - U^n\|_F + \|V - V^n\|_F}{\sqrt{m-n}}\right),$$

where Q is a distribution over predictors f, which depends on U and V but does not necessarily take the form of a neural network. The construction of this randomised proxy Q is central to our PAC-Bayes derived proof methods. U^n and V^n are matrices constructed using some subset $n < m$ of the data. Since we consider SGD-derived weights, we can leverage the empirical stability of this training method (through an idea introduced by Dziugaite et al., 2020b) to construct U^n, V^n which are quite close to the final true SGD-derived weights U, V, essentially by training a prior on the n-sized subset in the same way.

Outline. In Section 2 we give an overview of results from previous works which we use. In Section 3 we give an abstract and general formulation of our results as arising from a "majority vote of majority votes". In Section 4 we consider the conceptually simplest case of a single hidden network with "erf" activations and in Section 5 we extend this framework to GELU activations. In Section 6 we discuss our experimental setting and give our numerical results, which we discuss along with future work in Section 7.

2. Background and Related Work

PAC-Bayesian bounds. Originated by McAllester (1998, 1999), these generally consider the expected loss or Gibbs risk $L(Q) := \mathbb{E}_{f\sim Q}L(f)$ and analogously for the empirical risk, where $Q \in \mathcal{M}_1^+(\mathcal{F})$ is a distribution over randomised predictors $f \in \mathcal{F}$. The high-probability bounds take the rough form (although numerous variations using variance terms or attaining fast rates also exist – see the aforecited Guedj, 2019 and Alquier, 2021 for a survey)

$$L(Q) \leq \hat{L}_S(Q) + \mathcal{O}\left(\sqrt{\frac{\mathrm{KL}(Q,P) + \log(1/\delta)}{m}}\right) \tag{1}$$

holding with at least $1-\delta$ probability over the draw of the dataset. Here $\mathrm{KL}(Q,P)$ is the Kullback-Leibler divergence and $P \in \mathcal{M}_1^+(\mathcal{F})$ is the PAC-Bayesian "prior" distribution, which must be chosen in a data-independent way (but is not subject to the same requirements as a standard Bayesian prior for the validity of the method). This bound holds over all "posterior" distributions Q, but a poor choice (for example, one over-concentrated on a single predictor) will lead to a vacuous bound. We note in particular the following, which we use to prove our main results.

Theorem 2.1. Langford and Seeger (2001), Maurer (2004). *Given data distribution $\mathcal{D}$, $m \in \mathbb{N}^+$, prior $P \in \mathcal{M}_1^+(\mathcal{F})$, and $\delta \in (0,1)$, with probability $\geq 1-\delta$ over $S \sim D^m$, for all $Q \in \mathcal{M}_1^+(\mathcal{H})$*

$$L(Q) \leq \mathrm{kl}^{-1}\left(\hat{L}_S(Q),\ \frac{1}{m}\left(\mathrm{KL}(Q,P) + \log\frac{2\sqrt{m}}{\delta}\right)\right)$$

where $\mathrm{kl}^{-1}(u,c) := \sup\{v \in [0,1] : \mathrm{kl}(u,v) \leq c\}$ *and* $\mathrm{kl}(q:p) := q\log(q/p) + (1-q)\log((1-q)/(1-p))$.

Data-Dependent Priors. A careful choice of the prior is essential to the production of sharp PAC-Bayesian results. A variety of works going back to Ambroladze et al. (2006) and Parrado-Hernández et al. (2012) (and further developed by Dziugaite and Roy, 2018a; Dziugaite et al., 2020b; Rivasplata et al., 2018; Perez-Ortiz et al., 2021a,b, among others) have considered dividing the training sample into two parts, one to learn the prior and another to evaluate the bound. Formally, we divide $S = S^{\mathrm{prior}} \cup S^{\mathrm{bnd}}$ and use S^{prior} to learn a prior P^n where $n = |S^{\mathrm{prior}}|$, then apply the PAC-Bayesian bound using sample S^{bnd} to a posterior Q learned on the entirety of S. The resulting bound replaces $\hat{L}_S$ by $\hat{L}_{S^{\mathrm{bnd}}}$, P by the data-dependent P^n, and m by $m-n = |S^{\mathrm{bnd}}|$; thus the KL complexity term may be reduced at the cost of a smaller dataset to apply the bound to.

Dziugaite et al. (2020b) used this when considering training neural networks by constructing a so-called "coupled" prior P^n which is trained in the same way from the same initialisation as the posterior Q by stochastic gradient descent with the first n examples from the training set forming one epoch. Due to the stability of gradient descent, the

weights of P^n and Q evolve along similar trajectories; thus stability of the training algorithm is leveraged to tighten bounds without explicit stability results being required. In many ways this can be seen as an extension of previous work such as Dziugaite and Roy (2017) relating generalisation to the distance from initialisation rather than total weight norms.

Majority Votes. Since PAC-Bayesian bounds of the above in (1) and Theorem 2.1 generally consider the risk of randomised predictors, a natural question is whether prediction accuracy can be improved by "voting" many independently drawn predictions; such a majority vote predictor takes the deterministic form $F(x) := \mathbb{E}_{f\sim Q} f(x)$ with predictions as the argmaximum of $F(x)$. Several strategies have been devised to obtain bounds for these predictors via PAC-Bayesian theorems, with the simplest (and often most successful) being the unattributed first-order bound $\ell(F(x), y) \leq 2\mathbb{E}_{f\sim Q}\ell(f(x), y)$ valid for all (x, y), called the "folk theorem" by Langford and Shawe-Taylor (2003). This can be substituted directly into PAC-Bayesian theorems such as Theorem 2.1 above to obtain bounds for F at a de-randomisation cost of a factor of two. This is the result we use since across a variety of preliminary experiments we found other strategies including the tandem bound of Masegosa et al. (2020) and the C-bound of Lacasse et al. (2006) were uniformly worse, as also discussed by Zantedeschi et al. (2021).

Gaussian Sign Aggregation. To exploit the useful relationship above, Germain et al. (2009) considered aggregating a kind of linear prediction function of the form $f(x) = \text{sign}(w \cdot x)$ with $w \sim Q = N(u, I)$. In this case the aggregation can be stated in closed form using the Gaussian error function "erf" as

$$F(x) = \mathbb{E}_{w\sim N(u,I)} \text{sign}(w \cdot x) = \text{erf}\left(\frac{u \cdot x}{\sqrt{2}\|x\|_2}\right). \tag{2}$$

This closed-form relationship has been used since by Letarte et al. (2019) and Biggs and Guedj (2021b) in a PAC-Bayesian context for neural networks with sign activation functions and Gaussian weights; Biggs and Guedj (2021a) used it to derive a generalisation bound for SHEL (single hidden erf layer) networks, which have a single hidden layer with erf activation function. We will consider deriving a different PAC-Bayesian bound for this same situation and develop this method further in this work.

Other Approaches. A wide variety of other works have derived generalisation bounds for deterministic neural networks without randomisation. We note in particular the important works of Bartlett et al. (2017), Neyshabur et al. (2017) (using PAC-Bayesian ideas in their proofs) and Arora et al. (2018), but contrary to us, they do not provide empirically non-vacuous bounds. Nagarajan and Kolter (2019a) de-randomise PAC-Bayesian bounds by leveraging the notion of noise-resilience (how much the training loss of the network changes with noise injected into the parameters), but they note that in practice their bound would be numerically large. Many of these approaches utilise uniform convergence, which may lead to shortcomings as discussed at length by

Nagarajan and Kolter (2019b); we emphasise that the bounds we give are non-uniform and avoid these shortcomings. Finally, we also highlight the works of Neyshabur et al. (2015, 2019) which specifically consider single-hidden-layer networks as we do – as in the recent study from Tinsi and Dalalyan (2021). Overall we emphasise that, to the best of our knowledge, all existing bounds for deterministic networks are vacuous when networks are trained on real-world data.

3. General Formulation

We begin with a general analysis from which later misclassification generalisation bounds will be derived, by giving bounds for a "majority vote of majority votes", or a weighted combination of predictors which can themselves be expressed as majority votes. Since certain activation functions can be expressed as majority votes with stochastic weights, our later bounds will follow from this approach. These predictors take the form

$$F(x) := \mathbb{E}_{f\sim Q} f(x) = \sum_{k=1}^{K} v_k H_k(x), \tag{3}$$

where $v_k \in \mathbb{R}^D$ are the column vectors of a matrix $V \in \mathbb{R}^{D\times K}$ and $H_k : \mathcal{X} \to \mathbb{R}$ is itself a predictor of a form expressible by a majority vote. This means that there exists a distribution on functions $Q^k \in \mathcal{M}_1^+(\mathcal{F}^k)$ such that for each $x \in \mathcal{X}$, $H_k(x) = \mathbb{E}_{h\sim Q^k}[h(x)]$.

We show how predictors of this form can be expressed as a majority vote of some underlying stochastic predictor, $f \sim Q$. These lead to the following bound.

Theorem 3.1. *Fix a set of priors $P^k \in \mathcal{M}_1^+(\mathcal{F}^k)$ for $k \in [K]$, a prior weight matrix $V^0 \in \mathbb{R}^{D\times K}$, $\sigma_V > 0$, $\delta \in (0,1)$. With probability at least $1-\delta$ under the sample $S \sim \mathcal{D}^m$ simultaneously for any $V \in \mathbb{R}^{D\times K}$ and set of $Q^k \in \mathcal{M}_1^+(\mathcal{F}^k)$,*

$$L(F) \leq 2\ \mathrm{kl}^{-1}\left(\hat{L}_S(Q),\ \frac{\kappa}{m} + \frac{1}{m}\log\frac{2\sqrt{m}}{\delta}\right) \tag{4}$$

where F is the deterministic predictor given in Equation (3),

$$\kappa := \sum_{k=1}^{K} \mathrm{KL}(Q^k, P^k) + \frac{\|V - V^0\|_F^2}{2\sigma_V^2},$$

and

$$\hat{L}_S(Q) := \frac{1}{m}\sum_{(x,y)\in S} \Pr\left\{\sum_{k=1}^{K} w^k h^k(x) \neq y\right\}$$

is the stochastic predictor sample error where the probability is over independent draws of $w^k \sim N(v_k, \sigma_V^2 I), h^k \sim Q^k$ for all $k \in [K]$.

Proof. We are considering a distribution on functions of the form $\sum_k w^k h^k(x)$ where for each index $k \in [K]$ we have $w_k \sim N(\frac{1}{\sigma_V} v_k, I)$ and $h_k \sim Q_k$. This slightly different

formulation can take advantage of the scaling-invariance of the final layer to the misclassification loss when $V^0 = 0$, so we can then choose $\sigma_V > 0$ arbitrarily. The expectation of this takes the form given in Equation (3) scaled by $1/\sigma_V$ and leads to the empirical loss above.

Given another distribution P taking a similar form with $w_k \sim N(\frac{1}{\sigma_V} v_k^0, I)$ and components P_k, the KL divergence can be expressed as

$$\mathrm{KL}(Q, P) \leq \sum_{k=1}^{K} \mathrm{KL}(Q^k, P^k) + \frac{\|V - V^0\|_F^2}{2\sigma_V^2}.$$

We prove the overall bound by combining Theorem 2.1 with the first-order majority vote bound. □

4. SHEL Networks

We now present a specific example of the above leading to generalisation bounds for single-hidden-layer neural networks with an "erf" function activation.

Definition 4.1. *SHEL Network.* (Biggs and Guedj, 2021a) For $U \in \mathbb{R}^{K \times d}$, $V \in \mathbb{R}^{K \times D}$, and $\beta > 0$, a β-normalised single hidden erf layer (SHEL) network is defined by

$$F_{U,V}(x) := V \cdot \mathrm{erf}\left(\beta \frac{Ux}{\|x\|_2}\right).$$

The above is a single-hidden-layer network with a first normalisation layer, or if the data is already normalised the overall scaling $\|x\|_2$ can be absorbed into the β parameter. This parameter β could easily be absorbed into the matrix U and mainly has the effect of scaling the relative learning rate for U versus V when training by gradient descent, as shown by looking at $\frac{\partial}{\partial U} F_{U,V}(x)$, something which would normally be affected by the scaling of data. A higher β means more "feature learning" takes place as U has a relatively larger learning rate.

Using forms of the individual units as $h_k(x) = \mathrm{sign}(w_k \cdot x)$ and $w_k \sim N(u_k, (2\beta)^{-2} I)$ alongside Theorem 3.1, and defining the data-dependent prior matrices U^n, V^n from the first n train data points, we obtain the following generalisation bound in terms of the weight norms and performance of a stochastic network with sign activations, Gaussian weights and dropout.

Theorem 4.2. *Based on n data points from data distribution $\mathcal{D}$, fix data-dependent prior matrices $U^n \in \mathbb{R}^{K \times d}$ and $V^n \in \mathbb{R}^{D \times K}$, and $\sigma_V > 0$, $\beta > 0$. For $\delta \in (0, 1)$, with probability $\geq 1 - \delta$ under the sample $S^{\mathrm{bnd}} \sim \mathcal{D}^{m-n}$, for any $U \in \mathbb{R}^{K \times d}, V \in \mathbb{R}^{D \times K}$,*

$$L(F_{U,V}) \leq 2\ \mathrm{kl}^{-1}\left(\hat{L}_{S^{\mathrm{bnd}}}(Q),\ \frac{\kappa + \log \frac{2\sqrt{m-n}}{\delta}}{m-n}\right). \tag{5}$$

Here $F_{U,V}$ is a SHEL network with β-normalised activation,

$$\kappa := \beta^2 \|U - U^n\|_F^2 + \frac{\|V - V^n\|_F^2}{2\sigma_V^2},$$

and

$$\hat{L}_{S^{\text{bnd}}}(Q) := \frac{1}{m} \sum_{(x,y)\in S} \Pr\left\{W_2 \operatorname{sign}(W_1 x) \neq y\right\}$$

where the probability is over draws of $\operatorname{vec}(W_2) \sim N(\operatorname{vec}(V), \sigma_V^2 I_{DK})$, $\operatorname{vec}(W_1) \sim N(\operatorname{vec}(U), (2\beta)^{-2} I_{Kd})$. *Note that* vec *is the vectorisation operator and* sign *is applied element-wise.*

Proof. Apply the bound from Theorem 3.1 with the given choice of activation functions and Gaussian weights W_1 as described on the hidden layer. The aggregated form of the sign activation function is given in (2). The prior takes the same form as the posterior with weight means U^n, V^n and the same variances, leading to the form of KL divergence for Gaussian weights given in κ. □

Differences to Biggs and Guedj (2021a). In their Theorem 5, Biggs and Guedj (2021a) give a bound for generalisation in SHEL networks, with $L(F_{U,V})$ upper bounded under similar conditions to Theorem 4.2 by

$$\hat{L}_S^\gamma(F_{U,V}) + \tilde{O}\left(\frac{\sqrt{K}}{\gamma\sqrt{m}}\left(V_\infty \|U - U^0\|_F + \|V\|_F\right)\right),$$

where $\hat{L}_S^\gamma(g) = m^{-1}|\{(x,y) \in S : g(x)[y] - \max_{k\neq y} g(x)[k] \leq \gamma\}|$, the proportion of γ-margin errors in the training set and $V_\infty := \max_{ij} |V_{ij}|$. Thus a margin loss of the actual predictor used rather than a stochastic one appears. A tighter formulation more similar to Equation (5) is also given in an appendix and the bound could be similarly adapted to a data-dependent prior.

The derivation of the bound is quite different from ours, relying on a quite differently-constructed randomised version of Q (which is however constructed to have mean $F_{U,V}$), and a de-randomisation procedure relying on margins and concentration rather than a majority vote bound. Both the form of Q used and the de-randomisation step lead to issues which we have addressed through our alternative formulation of Q and a majority vote bound: de-randomisation requires a very low variance Q, leading to the $\sqrt{K}/\gamma$ term in the bound, which is empirically very large for low margin losses. Thus as demonstrated in their experiments, the big-O term increases with widening networks. Finally we note the most important distinction to our work: contrary to the present work, Biggs and Guedj (2021a) do not obtain non-vacuous bounds in practice.

5. GELU Networks

The Gaussian Error Linear Unit (Hendrycks and Gimpel, 2016) is a commonly-used alternative to the ReLU activation defined by $\text{GELU}(t) := \Phi(t)\, t$ where $\Phi(t)$ is the

standard normal CDF. Far from the origin, the $\Phi(t)$ is saturated at zero or one so it looks much like a smoothed ReLU or SWISH activation (defined by Ramachandran et al., 2018 as $x/(1+e^{-cx})$ for some $c>0$). It was introduced to lend a more probabilistic interpretation to activation functions, and fold in ideas of regularisation by effectively averaging the output of adaptive dropout (Ba and Frey, 2013); its wide use reflects excellent empirical results in a wide variety of settings.

Definition 5.1. *GELU Network.* For $U \in \mathbb{R}^{K\times d}$, $V \in \mathbb{R}^{K\times D}$, and $\beta > 0$, a β-normalised single hidden layer GELU network is is defined by

$$F_{U,V}(x) := V \cdot \mathrm{GELU}\left(\beta \frac{Ux}{\|x\|_2}\right)$$

where $\mathrm{GELU}(t) := \Phi(t)\, t$.

Theorem 5.2. *Under the setting of Theorem 4.2, with $\sigma_U > 0$ and $\sigma_V > 0$,*

$$L(F_{U,V}) \leq 2\ \mathrm{kl}^{-1}\left(\hat{L}_{S^{\mathrm{bnd}}}(Q),\ \frac{\kappa + \log\frac{2\sqrt{m-n}}{\delta}}{m-n}\right), \tag{6}$$

for $F_{U,V}$ is a single-hidden-layer GELU network with β-normalised activation,

$$\kappa := \left(\beta^2 + \frac{1}{\sigma_U^2}\right)\frac{\|U-U^n\|_F^2}{2} + \frac{\|V-V^n\|_F^2}{2\sigma_V^2},$$

and

$$\hat{L}_{S^{\mathrm{bnd}}}(Q) := \frac{1}{m}\sum_{(x,y)\in S} \Pr\left\{W_2(\mathbf{1}_{W_1 x} \otimes (W_1' x)) \neq y\right\}$$

with the probability is over draws of $\mathrm{vec}(W_2) \sim N(\mathrm{vec}(V), \sigma_V^2 I_{DK})$, $\mathrm{vec}(W_1) \sim N(\mathrm{vec}(U), \beta^{-2} I_{Kd})$ and $\mathrm{vec}(W_1') \sim N(\mathrm{vec}(V), \sigma_U^2 I_{Kd})$ where vec is the vectorisation operator and the indicator function $\mathbf{1}_y$ is applied element-wise.

Proof. The proof takes the same form as that of Theorem 4.2. We note that the expectation under the given probability distributions of $W_2(\mathbf{1}_{W_1 x} \otimes (W_1' x)) = \|x\|_2 F_{U,V}(x)$, but since the misclassification loss is scaling-invariant this gives equivalent results. Choosing appropriate prior forms as in Theorem 4.2 gives the KL divergence which we substitute into Theorem 3.1. □

Although the proof method for Theorem 5.2 and the considerations around the hyper-parameter β are the same as for Theorem 4.2 and SHEL networks, one notable difference is the inclusion of the σ_U parameter. When this is very small, the stochastic predictions are effectively just a linear two-layer network with adaptive dropout providing the non-linearity. The ability to adjust the variability of the stochastic network hidden layer and thus $\hat{L}_S(Q)$ is a major advantage over the SHEL network; in SHEL networks this variability can only be changed through β, which is a fixed parameter related to the deterministic network, not just a quantity appearing only in the bound.

6. Numerical Experiments

For numerical evaluation and the tightest possible values of bounds, a few further ingredients are needed, which are here described. We also give the specific way these are evaluated in our later experiments.

Bounding the empirical error term. We note that there is rarely a closed form expression for $\hat{L}_S(Q)$. This term must be estimated and bounded by making many independent draws of the parameters and using the fact that the quantity is bounded in $[0,1]$ to provide a concentration bound through for example Hoeffding's inequality. This adds a penalty to the bound which reduces with the number of independent draws and thus the amount of computing time invested in calculating the bound, but this is not a theoretical drawback of the bound. We give here a form which is useful in the neural network setting, where it computationally efficient to re-draw predictors for every prediction, but we make T passes through the dataset to ensure a tight bound. This formulation is considerably more computationally efficient than drawing a single h for every pass of the dataset.

Theorem 6.1 (Train Set Bound). *Let Q be some distribution over predictors and $h^{i,t} \sim Q$ be i.i.d. draws for $i \in [m], t \in [T]$. Then with probability $\geq 1-\delta'$,*

$$\hat{L}_S(Q) \leq \frac{1}{mT}\sum_{i=1}^{m}\sum_{t=1}^{T}\ell(h^{i,t}(x_i), y_i) + \sqrt{\frac{\log\frac{1}{\delta'}}{2mT}}.$$

Proof. Define $\xi = \sum_{i=1}^{m}\sum_{t=1}^{T}\frac{1}{mT}\ell(h^{i,t}(x_i), y_i)$ which has expectation $\mathbb{E}_Q\xi = \hat{L}_S(Q)$. Since this quantity is a sum of mT independent random variables in $\{0, 1/mT\}$, application of Hoeffding's inequality gives the result. □

In our results, we will set $\delta' = 0.01$, $T = 20$, and the generalisation bound $\delta = 0.025$; combining them our overall results will hold with probability at least $\delta + \delta' = 0.035$, as in Dziugaite and Roy (2017).

Variance Parameters β and σ. The parameters β, σ_V and σ_U control the variances of the weights in the stochastic estimator defined by Q, but fulfil different functions. The β parameter appears in the non-stochastic shallow network $F_{U,V}$ and thus affects the final predictions made and the training by SGD, and can be related to data normalisation as discussed above. We therefore set it to the fixed value of $\beta = 5$ in our experiments.

However the σ parameters appear only on the right hand side of the bounds for SHEL and GELU, and can be tuned to provide the tightest bounds—as they grow the KL term reduces but the performance of Q will degrade. We therefore optimise the final bounds over a grid of σ values as follows: choose a prior grid of σ_V values, $\sigma_V \in \{\sigma_V^1, \ldots, \sigma_V^r\}$, and combine via a union bound argument to add a $\log(r)$ term to κ where r is the number of grid elements. The same practice is applied to σ_U in the GELU case. In practice we use a grid $\sigma \in \{0.05, 0.06, \ldots, 0.2\}$ for both.

Coupling Procedure. We adopt a 60%-prefix coupling procedure for generating the prior weights U^n, V^n as in Dziugaite et al. (2020b). This works by taking the first 60% of training examples used in our original SGD run and looping them in the same order for up to 4000 epochs. These are used to train a prior model of the same architecture with the same learning rate from the same initialisation (this is valid because the initialisation is data-independent). The best bound from the generated prior weights was chosen (with a small penalty for this choice added to the bound via a union argument).

Numerical Results. In order to evaluate the quality of the bounds provided, we made many evaluations of the bound under many different training scenarios. In particular we show that the bound behaves in similar ways to the test error on changes of the width, learning rate, training set size and random relabelling of the data.

The following results follow by training β-normalised SHEL and GELU networks with stochastic gradient descent on the cross-entropy loss to a fixed cross entropy value of 0.3 for Fashion-MNIST and 0.1 for MNIST. We trained using SGD with momentum $= 0.9$ (as suggested by Hendrycks and Gimpel, 2016 and following Biggs and Guedj, 2021a) and a batch size of 200, or without momentum and a batch size of 1000 (with this larger batch size stabilising training).

We evaluated for ten different random seeds, a grid search of learning rates $\in \{0.1, 0.03, 0.01\}$ without momentum, and additionally $\in \{0.003, 0.001\}$ with momentum (where small learning rate convergence was considerably faster), and widths $\in \{50, 100, 200, 400, 800, 1600\}$ to generate the bounds in Table 1.

Best Coupled Bounds with Momentum

	Data	Test Err	Full Bnd	Coupled Bnd
SHEL	MNIST	0.046	0.772	0.490
GELU	MNIST	0.043	0.693	0.293
SHEL	Fashion	0.150	0.984	0.727
GELU	Fashion	0.153	0.976	0.568

Best Coupled Bounds without Momentum

	Data	Test Err	Full Bnd	Coupled Bnd
SHEL	MNIST	0.038	0.821	0.522
GELU	MNIST	0.036	0.742	0.317
SHEL	Fashion	0.136	1.109	0.844
GELU	Fashion	0.135	1.100	0.709

Table 1: Results for β-normalised (with $\beta = 5$) SHEL and GELU networks trained with and without momentum SGD on MNIST and Fashion-MNIST after a grid search of learning rates and widths as described above. Results shown are those obtaining the tightest coupled bound, with the accompanying full train set bound and test error for the same hyper-parameter settings.

From these results we also show plots in Figure 1 of the test error, stochastic error $\hat{L}_{S^{\text{bnd}}}(Q)$ and best prior bound versus width for the different dataset/activation combinations, with more plots given in the appendix. We also note here that in all except the width $= 50$ case, our neural networks have more parameters than there are train data points (60000).

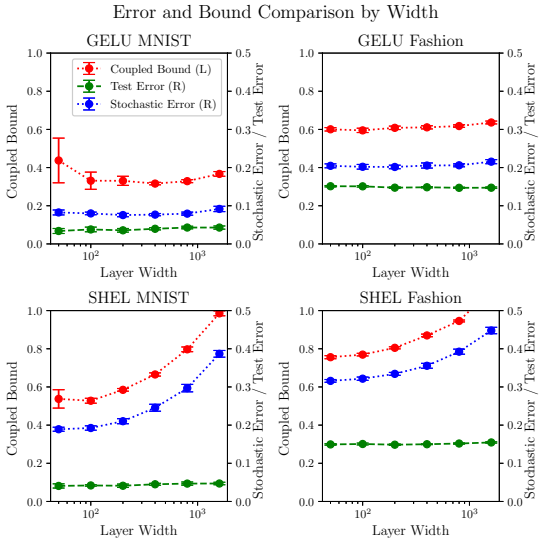

Figure 1: Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\hat{L}_{S^{\text{bnd}}}(Q)$ on the **right** (R) axis versus width for SHEL and GELU networks trained with momentum SGD and learning rate 0.01 on Fashion-MNIST and MNIST. Error bars show 1 standard deviation from ten different random seeds. The different scales are chosen so the trade-off between $\hat{L}_{S^{\text{bnd}}}(Q)$ and complexity terms can be seen more easily by neglecting the overall factor of 2, and the trends can be seen more clearly.

7. Discussion

In Table 1 we have given the first non-vacuous bounds for two types of deterministic neural networks trained on MNIST and Fashion-MNIST through a standard SGD learning algorithm, both with and without momentum. The coupled bounds are in all cases far from vacuous, with even the full bounds being non-vacuous in most cases, particularly on the easier MNIST task. Further, Figures 1 and 2 show that the bounds are robustly non-vacuous across a range of widths and learning rates. Since these are direct bounds on $L(F_{U,V})$ rather than the usual PAC-Bayes $L(Q)$, we emphasise that (for fixed hyper-parameters) no trade off is made between the tightness of the bound and the real test set performance, which is usually worse for a higher-variance (and thus more tightly bounded) Q.

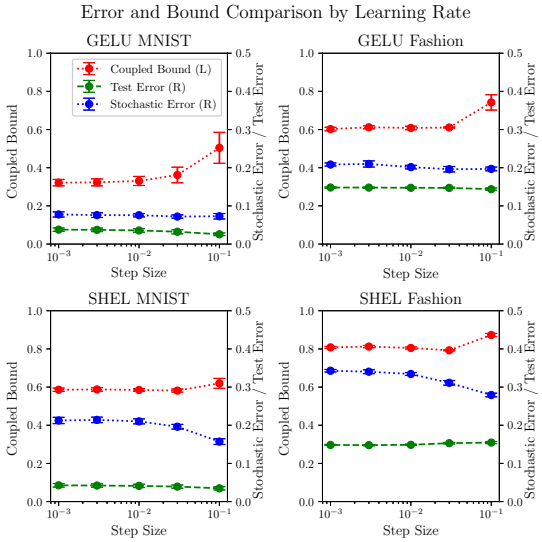

Figure 2: Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\hat{L}_{S^{\text{bnd}}}(Q)$ on the **right** (R) axis versus learning rate for width 200 SHEL and GELU networks trained with momentum SGD on Fashion-MNIST and MNIST. Scales are as in Figure 1.

Stability and Robustness Trade-Off. The two main contributions to the bound are the empirical error $\hat{L}(Q)$ and the KL divergence incorporated in κ. $\hat{L}(Q)$ can be seen roughly as measuring a combination of the difficulty of the task for our predictor $F_{U,V}$ combined with some kind of perturbation resistance of its weights (like the idea of a flat minimum originated in Hinton and van Camp, 1993 and discussed at length by Dziugaite and Roy, 2017); while κ is here an empirical measure of the stability of the training method, scaled by the inverse width of the perturbation robustness.

When optimising the trade-off between these terms through a choice of σ_U, σ_V values, we find that the complexity contribution to the bound remains relatively consistent across datasets and architectures, while it is the stochastic error that varies. This is especially true of SHEL networks as seen in Figure 1, perhaps since there is no easy way to set the stochastic error small by adjusting the variability of the Q hidden layer. This is in direct contrast to many works (Jiang et al., 2020b; Dziugaite et al., 2020a) evaluating the predictive ability of PAC-Bayesian bounds for generalisation on hyper-parameter changes, which fix the weight variances as the largest leading to a bound on $\hat{L}_S(Q)$ of a fixed value, say 0.1. Our results show that this approach may be sub-optimal for predicting generalisation, if as in our results the optimal trade-off tends to fix the κ term and trade off the size of $\hat{L}_S(Q)$ instead of the reverse[1].

Width Comparison. For the width comparisons we note that it is difficult to discern the real trend in the out-of-sample error of our trained networks. The test sets only

[1]The use of bi-criterion plots as suggested by Neyshabur et al. (2017) may therefore offer an better alternative when comparing vacuous bounds.

have 10000 examples and thus any test-set estimate of $L(F_{U,V})$ is subject to error; if the differences between test errors of two networks of different widths is smaller than about 0.02 (obtained through a Hoeffding bound) it is not possible to say if generalisation is better or worse. It is therefore possible that the pattern of weaker bounds for wider SHEL networks seen is a strong amplification of an existing trend, but it seems more likely it is an artefact of the bound shared with that of Biggs and Guedj (2021a).

Assuming the latter conclusion that the trained network true error really is relatively width-independent, the GELU bound does better matching this prediction (with this also being true in the momentum-free case, see appendix). The value of $\hat{L}_{S^{\text{bnd}}}(Q)$ stays roughly constant as width increases, while we observe that the optimal bound σ_U tends to decrease with increasing width. We attribute to this the tighter bounds for wide GELU networks, since the SHEL network has no comparable way to reduce the randomness of the hidden layer in Q, as we discuss at the end of Section 5.

Lower-Variance Stochastic Predictions. Following from the above, we note that in general $\hat{L}_{S^{\text{bnd}}}(Q)$ is smaller for comparably-trained GELU networks than the SHEL networks. We speculate that this arises from the increased randomness of the hidden layer of Q in Theorem 4.2: the sign activation is only $\{+1, -1\}$-valued and the amount of information coming through this layer is therefore more limited; and a $\{+1, -1\}$-valued random variable has maximum variance among $[+1, -1]$-bounded variables of given mean. In future work we will explore whether variance reduction techniques such as averaging multiple samples for each activation can improve the tightness of the bounds, but we also emphasise both that the bounds are still non-vacuous across a range of widths, and that the ability to adjust this variability is a central advantage of our new GELU formulation.

Learning Rate Comparison and Stability. In the case of training with momentum SGD we see that a very large learning rate leads to weaker and higher-variance bounds, with significantly larger norm contribution in κ. We speculate this arises because of the reduced stability at such high rates: we found in general that small batch sizes (particularly under vanilla SGD) and fast learning rates caused the training trajectory of U^n, V^n to diverge more greatly from that of U, V.

Improving Prior Coupling. With the instability of high learning rates and the empirical observation that in many cases $\hat{L}_S(Q)$ was very close to $L(Q)$ (as estimated from the test set), we see that there is a degree of slackness in the bound arising from the κ term. We speculate that it may be possible to make more efficient use of the sample S in constructing U^n, V^n to reduce this term further. This might be possible through an improved coupling scheme, or through extra side-channel information from S^{bnd} which can be compressed (as per Zhou et al., 2019) or is utilised in a differentially-private manner (as by Dziugaite and Roy, 2018a).

Majority Votes. In our results we rely on the novel idea of randomised single-hidden-layer neural networks as majority votes for de-randomisation of our PAC-Bayes bound.

We found empirically (similarly to many PAC-Bayesian works) that $L(Q) > L(F_{U,V})$, in other words the majority vote was better than the stochastic version on the test set. By de-randomising through the first order bound, we introduce a factor of 2 which cannot be tight in such cases – removal of this term would lead to considerably tighter bounds and even (based on preliminary experiments) non-vacuous bounds for CIFAR-10 (Krizhevsky, 2009). Improved bounds for the majority vote have been the focus of a wide variety of PAC-Bayesian works (Lacasse et al., 2006; Masegosa et al., 2020), and can theoretically give tighter results for $L(F_{U,V})$ than $L(Q)$, but these are not yet competitive. They universally led to inferior or vacuous results in preliminary experiments. However, there is still much scope for exploration here: alternative formulations of the oracle C-bound lead to different empirical bounds, and improvement of the KL term (which appears more times in an empirical C-bound than Theorem 2.1) may improve these bounds more than the first order one. We also hope that offering this new perspective on one-hidden-layer networks as majority votes can lead to better understanding of their properties, and perhaps even of closely-related Gaussian processes (Neal, 1996).

Summary. We have provided non-vacuous generalisation bounds for shallow neural networks through a novel method that makes a promising new link to majority votes. Although some aspects of our approach have recently appeared in the PAC-Bayesian literature on neural networks, we note that all previous results obtaining non-vacuous generalisation bounds only apply to randomised versions of neural networks. This often leads to degraded test set performance versus a deterministic predictor. By providing bounds directly on the deterministic networks we provide a setting through which the impact of robustness, flat-minima and stability on generalisation can be explored directly, without making potentially sub-optimal trade-offs or invoking stringent assumptions.

In future work we intend to address two main potential sources of improvement: through progress in majority votes to tighten the step from stochastic to deterministic predictor; and through development of the prior (perhaps thorough improved utilisation of data), a strand running parallel to much PAC-Bayesian research on neural networks.

References

Pierre Alquier. User-friendly introduction to PAC-Bayes bounds. 2021. URL https://www.arxiv.org/abs/2110.11216.

Amiran Ambroladze, Emilio Parrado-Hernández, and John Shawe-Taylor. Tighter pac-bayes bounds. In Bernhard Schölkopf, John C. Platt, and Thomas Hofmann, editors, *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, pages 9–16. MIT Press, 2006. URL https://proceedings.neurips.cc/paper/2006/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstra

Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. In Jennifer G. Dy and Andreas

Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 254–263. PMLR, 2018. URL http://proceedings.mlr.press/v80/arora18b.html.

Lei Jimmy Ba and Brendan J. Frey. Adaptive dropout for training deep neural networks. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3084–3092, 2013. URL https://proceedings.neurips.cc/paper/2013/hash/7b5b23f4aadf9513306bcd59afb6e4c9-Abstra

Peter L. Bartlett, Dylan J. Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6240–6249, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/b22b257ad0519d4500539da3c8bcf4dd-Abstra

Felix Biggs and Benjamin Guedj. On margins and derandomisation in PAC-Bayes. *CoRR*, abs/2107.03955, 2021a. URL https://arxiv.org/abs/2107.03955.

Felix Biggs and Benjamin Guedj. Differentiable PAC-Bayes objectives with partially aggregated neural networks. *Entropy*, 23(10):1280, 2021b. doi: 10.3390/e23101280. URL https://doi.org/10.3390/e23101280.

Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *Conference on Uncertainty in Artificial Intelligence 33.*, 2017.

Gintare Karolina Dziugaite and Daniel M. Roy. Data-dependent PAC-Bayes priors via differential privacy. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 8440–8450, 2018a. URL https://proceedings.neurips.cc/paper/2018/hash/9a0ee0a9e7a42d2d69b8f86b3a0756b1-Abstra

Gintare Karolina Dziugaite and Daniel M Roy. Data-dependent PAC-Bayes priors via differential privacy. In *Advances in Neural Information Processing Systems 31*, pages 8430–8441. Curran Associates, Inc., 2018b. URL http://papers.nips.cc/paper/8063-data-dependent-pac-bayes-priors-via-differential-priv

Gintare Karolina Dziugaite, Alexandre Drouin, Brady Neal, Nitarshan Rajkumar, Ethan Caballero, Linbo Wang, Ioannis Mitliagkas, and Daniel M. Roy. In search of

robust measures of generalization. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020a. URL https://proceedings.neurips.cc/paper/2020/hash/86d7c8a08b4aaa1bc7c599473f5dddda-Abstra

Gintare Karolina Dziugaite, Kyle Hsu, Waseem Gharbieh, and Daniel M. Roy. On the role of data in PAC-Bayes bounds. *CoRR*, abs/2006.10929, 2020b. URL https://arxiv.org/abs/2006.10929.

Pascal Germain, Alexandre Lacasse, François Laviolette, and Mario Marchand. PAC-Bayesian learning of linear classifiers. In *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*, pages 1–8, Montreal, Quebec, Canada, 2009. ACM Press. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553419.

Benjamin Guedj. A Primer on PAC-Bayesian Learning. In *Proceedings of the second congress of the French Mathematical Society*, 2019. URL https://arxiv.org/abs/1901.05353.

Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with Gaussian error linear units. *CoRR*, abs/1606.08415, 2016. URL https://arxiv.org/abs/1606.08415.

Geoffrey E. Hinton and Drew van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In Lenny Pitt, editor, *Proceedings of the Sixth Annual ACM Conference on Computational Learning Theory, COLT 1993, Santa Cruz, CA, USA, July 26-28, 1993*, pages 5–13. ACM, 1993. doi: 10.1145/168304.168306. URL https://doi.org/10.1145/168304.168306.

Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Comput.*, 9(1):1–42, 1997. doi: 10.1162/neco.1997.9.1.1. URL https://doi.org/10.1162/neco.1997.9.1.1.

Yiding Jiang, Pierre Foret, Scott Yak, Daniel M. Roy, Hossein Mobahi, Gintare Karolina Dziugaite, Samy Bengio, Suriya Gunasekar, Isabelle Guyon, and Behnam Neyshabur. NeurIPS 2020 competition: Predicting generalization in deep learning. *CoRR*, abs/2012.07976, 2020a. URL https://arxiv.org/abs/2012.07976.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020b. URL https://openreview.net/forum?id=SJgIPJBFvH.

Kenji Kawaguchi, Leslie Pack Kaelbling, and Yoshua Bengio. Generalization in deep learning. To appear in Mathematics of Deep Learning, Cambridge University Press, 2020. URL https://www.arxiv.org/abs/1710.05468.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Alexandre Lacasse, François Laviolette, Mario Marchand, Pascal Germain, and Nicolas Usunier. PAC-Bayes bounds for the risk of the majority vote and the variance of the Gibbs classifier. In Bernhard Schölkopf, John C. Platt, and Thomas Hofmann, editors, *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, pages 769–776. MIT Press, 2006. URL https://proceedings.neurips.cc/paper/2006/hash/779efbd24d5a7e37ce8dc93e7c04d572-Abstra

John Langford and Matthias Seeger. Bounds for averaging classifiers. 2001. URL http://www.cs.cmu.edu/~jcl/papers/averaging/averaging_tech.pdf.

John Langford and John Shawe-Taylor. PAC-Bayes & margins. In *Advances in Neural Information Processing Systems*, pages 439–446, 2003.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Gaël Letarte, Pascal Germain, Benjamin Guedj, and Francois Laviolette. Dichotomize and generalize: PAC-Bayesian binary activated deep neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 6872–6882. Curran Associates, Inc., 2019.

Andrés R. Masegosa, Stephan Sloth Lorenzen, Christian Igel, and Yevgeny Seldin. Second order PAC-Bayesian bounds for the weighted majority vote. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/386854131f58a556343e056f03626e00-Abstra

Andreas Maurer. A note on the PAC-Bayesian theorem. *CoRR*, cs.LG/0411099, 2004. URL https://arxiv.org/abs/cs.LG/0411099.

David A McAllester. Some PAC-Bayesian theorems. In *Proceedings of the eleventh annual conference on Computational Learning Theory*, pages 230–234. ACM, 1998.

David A McAllester. PAC-Bayesian model averaging. In *Proceedings of the twelfth annual conference on Computational Learning Theory*, pages 164–170. ACM, 1999.

Vaishnavh Nagarajan and J. Zico Kolter. Deterministic PAC-Bayesian generalization bounds for deep networks via generalizing noise-resilience. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019a. URL https://openreview.net/forum?id=Hygn2o0qKX.

Vaishnavh Nagarajan and J. Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11611–11622, 2019b. URL https://proceedings.neurips.cc/paper/2019/hash/05e97c207235d63ceb1db43c60db7bbb-Abstra

Radford M. Neal. *Priors for Infinite Networks*, pages 29–53. Springer New York, New York, NY, 1996. ISBN 978-1-4612-0745-0. doi: 10.1007/978-1-4612-0745-0_2. URL https://doi.org/10.1007/978-1-4612-0745-0_2.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In Peter Grünwald, Elad Hazan, and Satyen Kale, editors, *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, volume 40 of *JMLR Workshop and Conference Proceedings*, pages 1376–1401. JMLR.org, 2015. URL http://proceedings.mlr.press/v40/Neyshabur15.html.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5947–5956, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/10ce03a1ed01077e3e289f3e53c72813-Abstra

Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A PAC-Bayesian approach to spectrally-normalized margin bounds for neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=Skz_WfbCZ.

Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. The role of over-parametrization in generalization of neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=BygfghAcYX.

Emilio Parrado-Hernández, Amiran Ambroladze, John Shawe-Taylor, and Shiliang Sun. PAC-Bayes bounds with data dependent priors. *J. Mach. Learn. Res.*, 13:3507–3531, 2012. URL http://dl.acm.org/citation.cfm?id=2503353.

Maria Perez-Ortiz, Omar Rivasplata, Benjamin Guedj, Matthew Gleeson, Jingyu Zhang, John Shawe-Taylor, Miroslaw Bober, and Josef Kittler. Learning PAC-Bayes priors for probabilistic neural networks. 2021a. URL https://arxiv.org/abs/2109.10304.

Maria Perez-Ortiz, Omar Rivasplata, John Shawe-Taylor, and Csaba Szepesvari. Tighter risk certificates for neural networks. *Journal of Machine Learning Research*, 22(227): 1–40, 2021b. URL http://jmlr.org/papers/v22/20-879.html.

Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=Hkuq2EkPf.

Omar Rivasplata, Csaba Szepesvári, John Shawe-Taylor, Emilio Parrado-Hernández, and Shiliang Sun. PAC-Bayes bounds for stable algorithms with instance-dependent priors. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 9234–9244, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/386854131f58a556343e056f03626e00-Abstra

Laura Tinsi and Arnak S. Dalalyan. Risk bounds for aggregated shallow neural networks using gaussian priors. 2021. URL https://arxiv.org/abs/2112.11086.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017. URL https://arxiv.org/abs/1708.07747.

Valentina Zantedeschi, Paul Viallard, Emilie Morvant, Rémi Emonet, Amaury Habrard, Pascal Germain, and Benjamin Guedj. Learning stochastic majority votes by minimizing a PAC-Bayes generalization bound. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS*, 2021. URL https://arxiv.org/abs/2106.12535.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Sy8gdB9xx.

Wenda Zhou, Victor Veitch, Morgane Austern, Ryan P. Adams, and Peter Orbanz. Non-vacuous generalization bounds at the ImageNet scale: A PAC-Bayesian compression approach. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BJgqqsAct7.

A. Additional Results and Code

Code for the paper at is included at https://github.com/biggs/shallow-nets.

Here we also provide Figures 3 and 4 similar to Figures 1 and 2 for GELU and SHEL networks trained without momentum and with a batch size of 1000, as described in Section 6. We then also provide further similar plots for networks trained with momentum and a batch size of 200 as in Section 6 with different learning rates and widths, to show the similar behaviour across a variety of regimes.

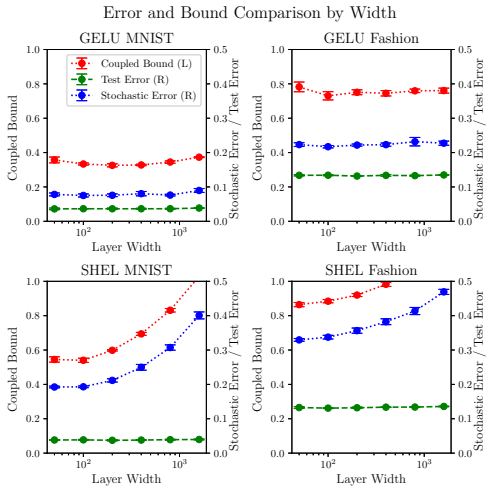

Figure 3: Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\hat{L}_{S^{\text{bnd}}}(Q)$ on the **right** (R) axis versus width for SHEL and GELU networks trained with vanilla SGD and learning rate 0.01 on Fashion-MNIST and MNIST. Scales are as in Figure 1.

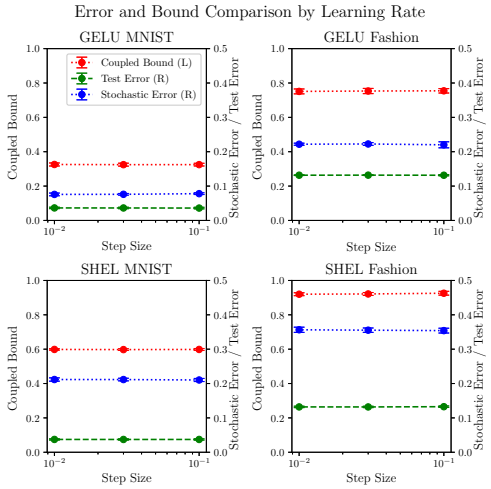

Figure 4: Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\hat{L}_{S^{\text{bnd}}}(Q)$ on the **right** (R) axis versus learning rate for width 200 SHEL and GELU networks trained with vanilla SGD on Fashion-MNIST and MNIST. Scales are as in Figure 1.

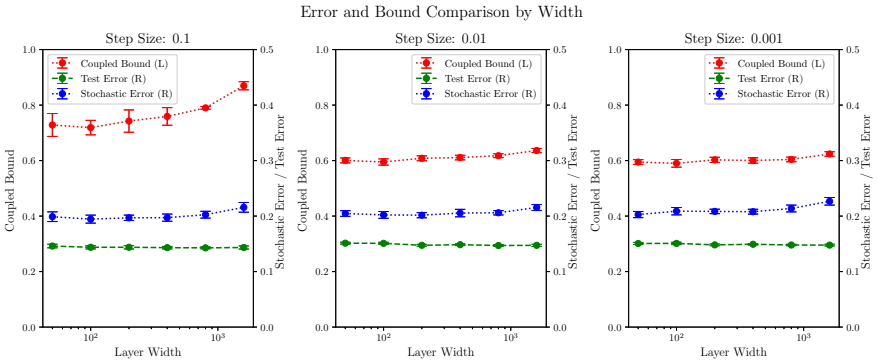

Figure 5: Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\hat{L}_{S^{\text{bnd}}}(Q)$ on the **right** (R) axis versus width under fixed other hyperparameters, for a GELU network trained with momentum on Fashion-MNIST.

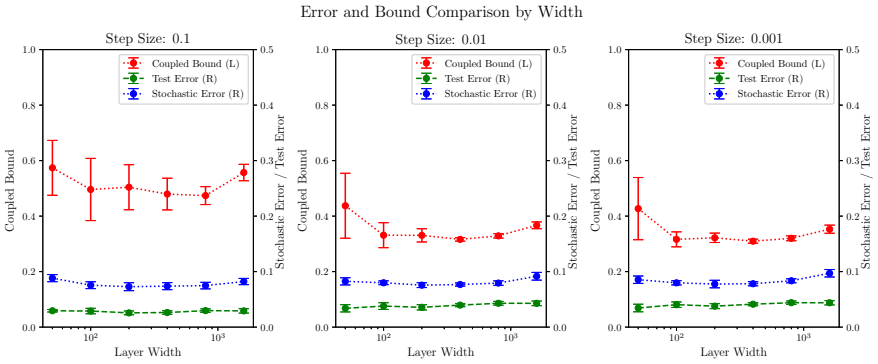

Figure 6: Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\hat{L}_{\mathcal{S}^{\text{bnd}}}(Q)$ on the **right** (R) axis versus width under fixed other hyperparameters, for a GELU network trained with momentum on MNIST.

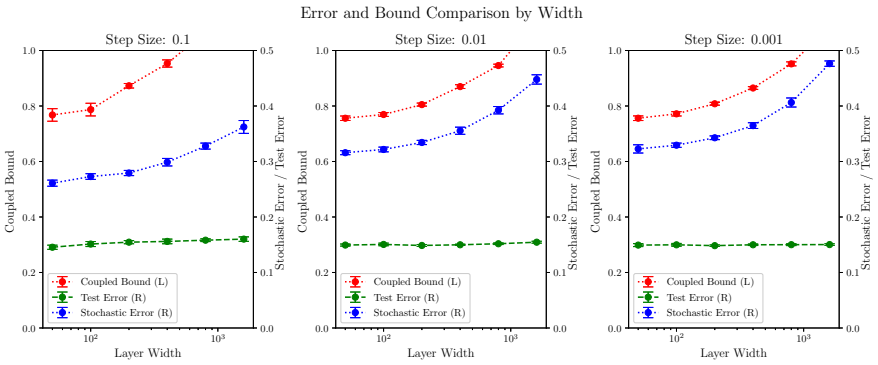

Figure 7: Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\hat{L}_{\mathcal{S}^{\text{bnd}}}(Q)$ on the **right** (R) axis versus width under fixed other hyperparameters, for a SHEL network trained with momentum on Fashion-MNIST.

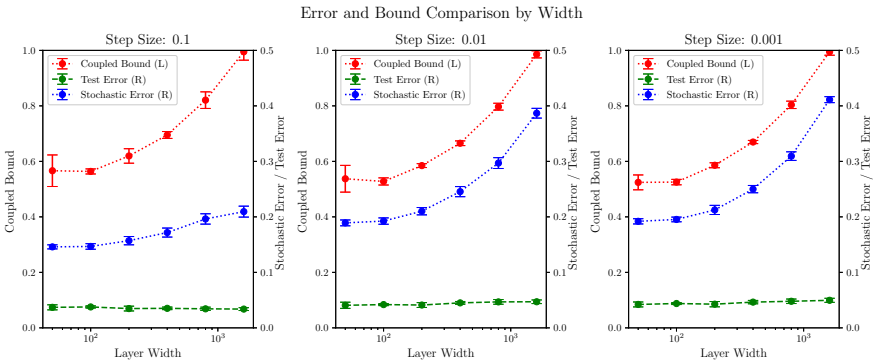

Figure 8: Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\hat{L}_{\mathcal{S}^{\text{bnd}}}(Q)$ on the **right** (R) axis versus width under fixed other hyperparameters, for a SHEL network trained with momentum on MNIST.

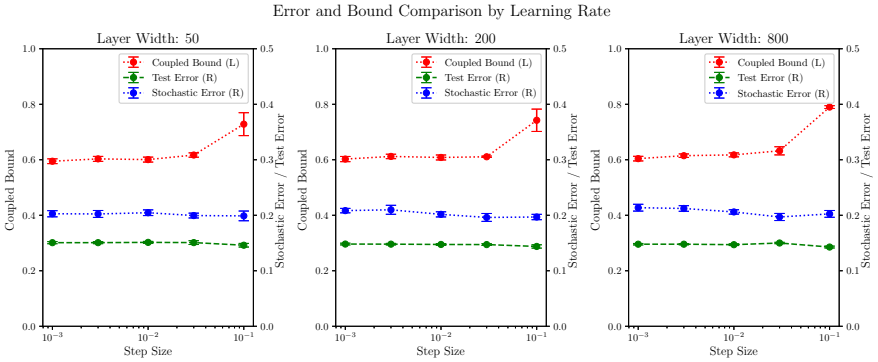

Figure 9: Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\hat{L}_{S^{\text{bnd}}}(Q)$ on the **right** (R) axis versus learning rate under fixed other hyperparameters, for a GELU network trained with momentum on Fashion-MNIST.

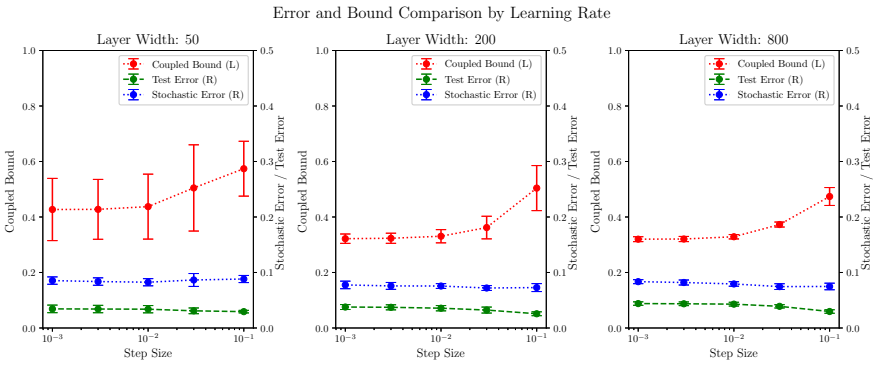

Figure 10: Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\hat{L}_{S^{\text{bnd}}}(Q)$ on the **right** (R) axis versus learning rate under fixed other hyperparameters, for a GELU network trained with momentum on MNIST.

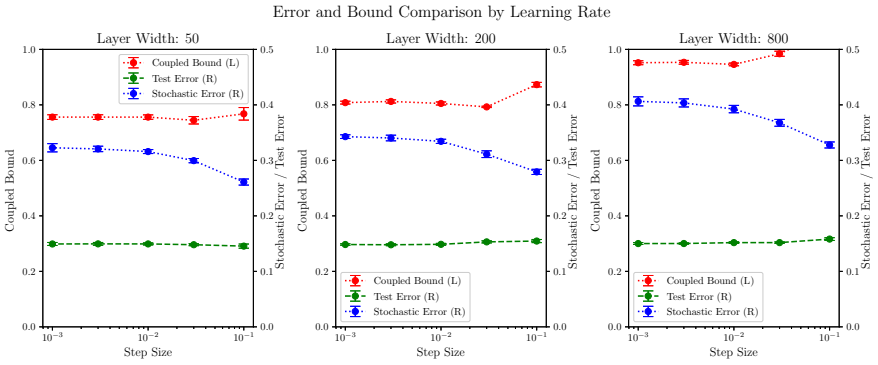

Figure 11: Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\hat{L}_{S^{\text{bnd}}}(Q)$ on the **right** (R) axis versus learning rate under fixed other hyperparameters, for a SHEL network trained with momentum on Fashion-MNIST.

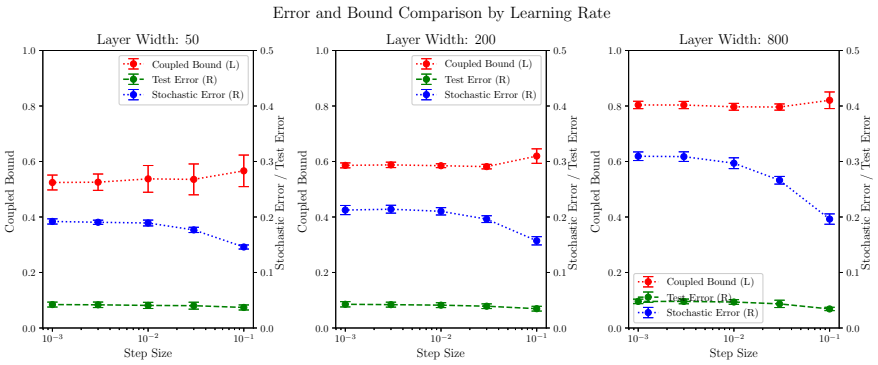

Figure 12: Changes in bound on **left** (L) hand axis, and test error and stochastic bound error $\hat{L}_{S^{\text{bnd}}}(Q)$ on the **right** (R) axis versus learning rate under fixed other hyperparameters, for a SHEL network trained with momentum on MNIST.