Know Your Limits: Uncertainty Estimation with ReLU Classifiers Fails at Reliable OOD Detection

Dennis Ulmer[1]   **Giovanni Cinà**[2]

[1]ITU Copenhagen, Copenhagen, Denmark
[2]Pacmed BV, Amsterdam, Netherlands,

Abstract

A crucial requirement for reliable deployment of deep learning models for safety-critical applications is the ability to identify out-of-distribution (OOD) data points, samples which differ from the training data and on which a model might underperform. Previous work has attempted to tackle this problem using uncertainty estimation techniques. However, there is empirical evidence that a large family of these techniques do not detect OOD reliably in classification tasks.

This paper gives a theoretical explanation for said experimental findings and illustrates it on synthetic data. We prove that such techniques are not able to reliably identify OOD samples in a classification setting, since their level of confidence is generalized to unseen areas of the feature space. This result stems from the interplay between the representation of ReLU networks as piece-wise affine transformations, the saturating nature of activation functions like softmax, and the most widely-used uncertainty metrics.

1 INTRODUCTION

Notwithstanding the tremendous improvements achieved in recent years by means of novel and larger deep learning architectures, advanced models still lack certain properties that guarantee their safety in high-stakes applications like health care [He et al., 2019], autonomous driving [McDermid et al., 2019], and more. Among other traits, the capability to discern familiar data samples seen during the training stage (in-distribution) from abnormal inputs (out-of-distribution) is of paramount importance in certain contexts. Take for instance a hospital, in which an algorithm is used to predict complications for a patient. Due to factors like changing patient demographics or protocols, but also simply differ-

ent hospital environments, predictions might become less reliable and cause harm to the patient. A degradation of the model performance might only be detected much later, when the shift in the test data becomes more apparent - at which point further damage accumulates, hence the need arises to implement techniques that can detect OOD samples reliably.

Unfortunately, it is well-known that neural network classifiers tend to be overconfident in their predictions [Guo et al., 2017], i.e. exhibiting high levels of certainty when it is unwarranted, and often fail to correctly identify OOD samples [Ovadia et al., 2019, Nalisnick et al., 2019]. A recent study on medical tabular data has shown that even techniques specifically developed to quantify the model's uncertainty struggle at detecting OOD samples for a relatively simple classification task [Ulmer et al., 2020]. Crucially, it was shown that neural discriminators tend to project vast areas of high certainty far away from the training distribution - a behaviour that seems completely at odds with reliable OOD detection. These observations can easily be replicated on synthetic data, as displayed in Figure 1a, where one can observe open areas of constant certainty stretching beyond the training data. The reasons for this behavior in a classification setting are hitherto much less studied.

In this paper we propose a novel theoretical argument to explain such phenomena, showing that certainty levels are generalized on sub-spaces defined by the network (see Figure 1b and 1c). We do this by simulating covariate shift for single feature values of real variables and studying the asymptotic behavior of the model. Our contributions are as follows:

1. Our first result shows that, under mild assumptions about the network's behaviour on certain subspaces, ReLU-based neural network classifiers coupled with widely used uncertainty metrics always converge to a fixed uncertainty level on OOD samples.

2. We extend this result by proving that variational inference-based and ensembling methods in combi-
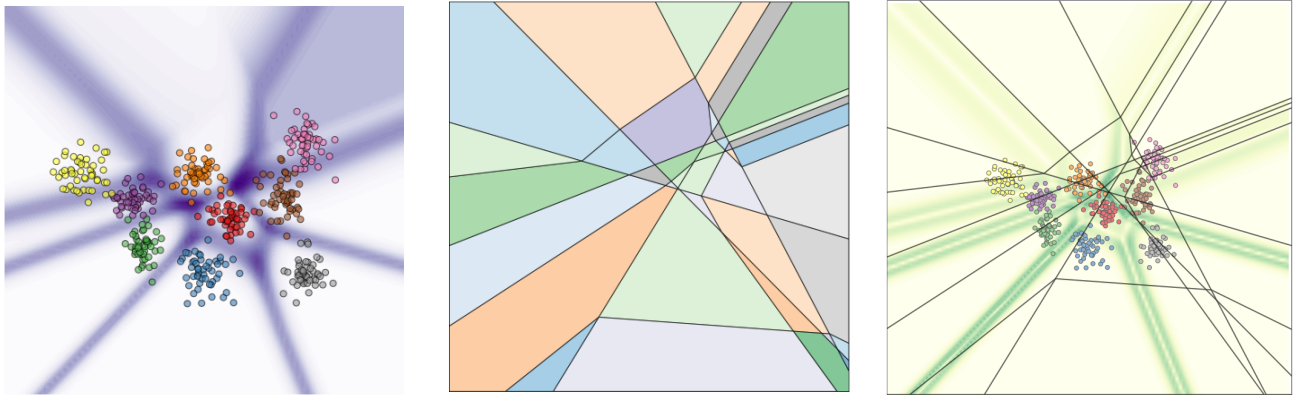
(a) Predictive entropy $\tilde{\mathbb{H}}[p_{\boldsymbol{\theta}}(y\,|\,\mathbf{x})]$ of ReLU classifier.

(b) Polytopal linear regions induced by same classifier [Arora et al., 2018].

(c) Magnitude of gradient of predictive entropy $||\nabla_{\mathbf{x}}\tilde{\mathbb{H}}[p_{\boldsymbol{\theta}}(y\,|\,\mathbf{x})]||_2$.

Figure 1: (a) Uncertainty of a neural classifier with ReLU activations measured by predictive entropy on synthetic data, illustrated by increasing shades of purple with white denoting absolute certainty. (b) Polytopal, linear regions in the feature space induced by the same classifier (as introduced by Arora et al. [2018]). (c) Norm of the gradient of the predictive entropy plotted by increasing shades of green, showing how small perturbations in the input have a decreasing influence on the uncertainty of the network as we stray away from the training data, creating large areas in which uncertainty levels are overgeneralized. Exceptions to this are the model's decision boundaries, which is discussed in Section 6. Polytopes are drawn using the code of Jordan et al. [2019].

nation with several uncertainty estimation techniques suffer from the same problem (Theorem 1). This phenomenon is illustrated and discussed on synthetic data.

These results entail that, when the conditions of the theorem are met, these models cannot be used to reliably detect OOD: since the level of certainty is generalized from seen to unseen data, the models are unable to differentiate between the two. The findings of this article have bearings on OOD detection for several critical applications using neural classifiers with ReLU activation functions.

2 RELATED WORK

Overconfidence in neural networks has been studied from several angles; we summarize here some of the main lines of research. One way to counteract the problem of overconfidence lies in the quantification of a network's uncertainty, which is usually divided into aleatoric and epistemic uncertainty. The former denotes *non-reducible* uncertainty, e.g. uncertainty intrinsic to the data generating process, while the latter refers to *reducible* uncertainty. This includes the knowledge about the best model class, as well as about the parameters which explain the data best [Der Kiureghian and Ditlevsen, 2009, Hüllermeier and Waegeman, 2019]. In this article, we are interested in the effect of methods that estimate uncertainty post-hoc. One way to evaluate uncertainty estimation methods is the study of their behaviour in presence of OOD samples [Ovadia et al., 2019]. Ulmer et al. [2020] specifically show how the methods used in our work

fail in practice to detect clinically relevant OOD groups of patients in a medical context. Kompa et al. [2020] conclude that many uncertainty methods produce confidence intervals which do not include the actual observations on OOD data.

A variety of articles approaches the phenomenon of overconfidence from a calibration perspective: starting with the work of Guo et al. [2017], follow-up work develops improved variants of temperature scaling [Laves et al., 2019] or new types of scaling altogether Kumar et al. [2018], Kull et al. [2019]. A separate line of enquiry investigates the effect of different activation functions [Bridle, 1990, Ramachandran et al., 2018], exploring alternatives to the sigmoid or softmax function[1] as the final component of neural discriminators [de Brébisson and Vincent, 2016, Hendrycks and Gimpel, 2017, Laha et al., 2018, Martins and Astudillo, 2016]. But to the best of our knowledge, only Hein et al. [2019] give a theoretical explanation for this behavior in ReLU-networks showing that, when a point is scaled uniformly, the network's probability mass is placed on a single class. However, they do not extend their findings to uncertainty estimation metrics nor investigate the implications for variational approaches or ensembling. Mukhoti et al. [2021] show that epistemic and aleatoric uncertainty cannot be disentangled successfully purely based on the output distribution of a single network.

---

[1]Their relationship is outlined in more detail in Appendix A.1 or Bridle [1990].

3 PRELIMINARIES

In this section we introduce some notation and relevant definitions for the rest of this work.

3.1 NOTATION

We denote sets in calligraphic letters, e.g. $\mathcal{P}$ or $\mathcal{C}$. Vectors are represented using lower-case bold letters such as $\mathbf{x}$ or $\boldsymbol{\theta}$. For functions with multiple outputs, a lowercase index refers to a specific component of the output, e.g. with a function $f : \mathbb{R} \to \mathbb{R}^N$, $f(x)_n$ denotes one of the output components with $n \in 1, ..., N$. Furthermore, we use $\odot$ to denote the Hadamard product and $|| \ldots ||_2$ for the l_2-norm.

3.2 OUT-OF-DISTRIBUTION DATA

Given a set $\mathcal{C} = \{1, \ldots, C\}$ of numbers denoting class labels, we define a data set to be a finite set $\mathcal{D} \subset \mathbb{R}^D \times \mathcal{C}$ containing N ordered pairs of D-dimensional feature vectors $\mathbf{x}_i$ and corresponding class labels y_i obtained from some (unknown) joint distribution $\mathbf{x}_i, y_i \sim p(\mathbf{x}, y)$ s.t. $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$. Although there exist many different notions of dataset shift [Shimodaira, 2000, Moreno-Torres et al., 2012], we particularly focus on *covariate shift*, in which the distribution of feature values - the covariates - differs from the original training distribution $p(\mathbf{x})$. We focus on this kind of shift as it is especially common in non-stationary environments like healthcare [Curth et al., 2019] and many other applications.

To simulate covariate shift, we obtain OOD samples by shifting points away from the training distribution by means of a scaling factor. This approach is in line with recent experiments on covariate shift and OOD detection [Ovadia et al., 2019, Ulmer et al., 2020]. We would expect a reliable OOD detection model to output increasingly higher uncertainty as points stray further and further away from the mass of $p(\mathbf{x})$, thus we study the behaviour of OOD detection models in the limit, when the scaling factor is allowed to grow indefinitely in at least one dimension.

3.3 UNCERTAINTY METRICS

We begin by defining a neural discriminator, which we assume to follow common architectural conventions, i.e. to consist of a series of affine transformations with ReLU [Glorot et al., 2011] activation functions. Together with a final softmax function [Bridle, 1990], it parametrizes a categorical distribution over classes.

Definition 1. *Let $\mathbf{x} \in \mathbb{R}^D$ be an input vector and C the number of classes in a classification problem. The unnormalized output of the network after L layers is a function $f_{\boldsymbol{\theta}} : \mathbb{R}^D \to \mathbb{R}^C$ with the final output following after an additional softmax function $\bar{\sigma}(\cdot)$ s.t. $p_{\boldsymbol{\theta}} = \bar{\sigma} \circ f_{\boldsymbol{\theta}}$, so $p_{\boldsymbol{\theta}}(y = c | \mathbf{x}) \equiv \bar{\sigma}(f_{\boldsymbol{\theta}}(\mathbf{x}))_c$. Thus the discriminator is represented by a function $p_{\boldsymbol{\theta}} : \mathbb{R}^D \to [0,1]^C$ which is parametrized by a vector $\boldsymbol{\theta}$.*

The softmax function $\bar{\sigma} : \mathbb{R}^C \to \mathbb{R}^C$ is commonly defined as

$$\bar{\sigma}(f_{\boldsymbol{\theta}}(\mathbf{x}))_c = \frac{\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_c)}{\sum_{c'=1}^C \exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{c'})} \tag{1}$$

We will consider a (non-exhaustive) set of popular uncertainty metrics in this work, which we introduce in turn. Hendrycks and Gimpel [2017] introduce a simple baseline, which is the highest probability observed for any class:

$$y_{\max} = \max_{c \in \mathcal{C}} p_{\boldsymbol{\theta}}(y = c | \mathbf{x})$$

Ideally, the model's predictive distribution would become more uniform for challenging inputs (e.g. in areas of class overlap) and thus produce a lower confidence score $y_{\max}$.[2] The other uncertainty estimation techniques introduced below try to approximate the uncertainty of the predictive distribution for a new data point $\mathbf{x}'$, which is commonly factorized as follows:

$$p(y | \mathbf{x}', \mathcal{D}) = \int p(y | \mathbf{x}', \boldsymbol{\theta}) p(\boldsymbol{\theta} | \mathcal{D}) d\boldsymbol{\theta} \tag{2}$$

In the following, we use $p(y | \mathbf{x}, \boldsymbol{\theta}) \equiv p_{\boldsymbol{\theta}}(y | \mathbf{x})$. This equation is intractable: the weight posterior $p(\boldsymbol{\theta} | \mathcal{D})$ cannot be computed precisely using Bayes' rule. Hence the weight posterior is often replaced by a variational posterior $q(\boldsymbol{\theta})$ and the expectation formed by this expression is commonly approximated by a Monte Carlo approximation:

$$\mathbb{E}_{p(\boldsymbol{\theta} | \mathcal{D})}\left[p_{\boldsymbol{\theta}}(y | \mathbf{x})\right] \approx \frac{1}{K} \sum_{k=1}^K p_{\boldsymbol{\theta}}^{(k)}(y | \mathbf{x}); \quad \boldsymbol{\theta}^{(k)} \sim q(\boldsymbol{\theta}) \tag{3}$$

Other methods for which a similar aggregation of predictions can be used include Markov-chain Monte-Carlo procedures [Welling and Teh, 2011, Neklyudov et al., 2020] or simply ensembling [Lakshminarayanan et al., 2017], which does not require sampling. In this work, we will use the term *instance* to refer to a network characterized by $\boldsymbol{\theta}^{(k)}$ in order to group all of these methods together. We follow the reasoning of Wilson and Izmailov [2020] to interpret both variational and ensembling methods as examples of Bayesian model averaging. A very intuitive method to aggregate all instances' predictions and estimate uncertainty is to compute the variance over all classes, such as in Smith and Gal [2018]:

$$\bar{\sigma}^2 = \frac{1}{C} \sum_{c=1}^C \mathbb{E}_{p(\boldsymbol{\theta} | \mathcal{D})}\left[\left(p_{\boldsymbol{\theta}}(y = c | \mathbf{x})\right)^2\right]$$

$$- \mathbb{E}_{p(\boldsymbol{\theta} | \mathcal{D})}\left[p_{\boldsymbol{\theta}}(y = c | \mathbf{x})\right]^2$$

---

[2]Which is why we measure *un*certainty by $1 - y_{\max}$.

The more predictions disagree, the larger the average variance per class will become. Another approach lies in measuring the Shannon entropy $\mathbb{H}$ of the predictive distribution [Gal, 2016]:

$$\tilde{\mathbb{H}}\Big[p_{\boldsymbol{\theta}}(y|\,\mathbf{x})\Big] = \mathbb{H}\bigg[\mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})}\Big[p_{\boldsymbol{\theta}}(y|\,\mathbf{x})\Big]\bigg]$$

Entropy is maximal when all probability mass is centered on a single class, i.e. all aggregated predictions agree on a label, and minimum when the predictive distribution is uniform. As these two metrics capture only the total uncertainty, we finally also consider the mutual information between model parameters and a data sample [Smith and Gal, 2018], which aims to isolate epistemic uncertainty:

$$\underbrace{\mathbb{I}(y, \boldsymbol{\theta}|\mathcal{D}, \mathbf{x})}_{\text{Model uncertainty}} \approx \underbrace{\mathbb{H}\bigg[\mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})}\Big[p_{\boldsymbol{\theta}}(y|\,\mathbf{x})\Big]\bigg]}_{\text{Total uncertainty}}$$
$$- \underbrace{\mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})}\bigg[\mathbb{H}\Big[p_{\boldsymbol{\theta}}(y|\,\mathbf{x})\Big]\bigg]}_{\text{Data uncertainty}}$$

The term itself can be interpreted as the gain in information about the ideal model parameters and correct label upon receiving an input. If we can only gain a little, that implies that parameters are already well-specified and that the epistemic uncertainty is low.

3.4 ADDITIONAL DEFINITIONS

Here we introduce some concepts related to monotonicity, which will become central in the next sections. In the univariate case, we call a function strictly increasing on an interval $\mathcal{I} = [a, b]$ with $a < b$ and $a, b \in \mathbb{R}$ if it holds that

$$\forall x' \in \mathcal{I} \left(\frac{\partial}{\partial x} f(x)\Big|_{x=x'} > 0\right)$$

where $\cdot|_{x=x'}$ refers to evaluating the value of the derivative of f at x'. This definition can also be extended to multivariate functions by requiring strict monotonicity (strictly increasing or decreasing) in all dimensions:

Definition 2. *We call a multivariate function $f : \mathbb{R}^D \to \mathbb{R}$ strictly monotonic on a subspace $\mathcal{P} \subseteq \mathbb{R}^D$ if it holds that*

$$\forall d \in 1, \ldots, D, \left(\forall \mathbf{x}' \in \mathcal{P}, \left(\nabla_{\mathbf{x}} f(\mathbf{x})\Big|_{\mathbf{x}=\mathbf{x}'}\right)_d < 0\right.$$
$$\left. \vee \ \forall \mathbf{x}' \in \mathcal{P}, \left(\nabla_{\mathbf{x}} f(\mathbf{x})\Big|_{\mathbf{x}=\mathbf{x}'}\right)_d > 0\right) \quad (4)$$

where $(\cdot)_d$ refers to $\frac{\partial f(x_d)}{\partial x_d}|_{x_d=x'_d}$, the d-th component of the gradient $\nabla_{\mathbf{x}} f(\mathbf{x})$ evaluated at $\mathbf{x}'$. We call a multivariate function $f : \mathbb{R}^D \to \mathbb{R}^C$ component-wise strictly monotonic if the above definition holds for the gradient of every output component $\nabla_{\mathbf{x}} f(\mathbf{x})_c$.

We note here that the softmax function is an example for a component-wise strictly monotonic function. As later lemmas investigate the behavior of functions in the limit, it is furthermore useful to define regions of the feature space that are unbounded in at least one direction. We call a *partially-unbounded polytope* (henceforth abbreviated by PUP) a convex subspace of $\mathbb{R}^D$ that is unbounded in at least one dimension d, i.e. if the polytope's projection onto d is either left-bounded by $-\infty$ or right-bounded by ∞, or both.

4 CONVERGENCE OF PREDICTIONS ON OOD

In this section we will show that, moving the input to the extremes of the feature space, a ReLU classifier will converge to a fixed prediction. To demonstrate this, we must establish how the distance from the training data affects the network's logits. To this end, we utilize a known result stating that neural networks employing piecewise linear activation functions partition the input space into polytopes (such as in Figure 1b; Arora et al., 2018). Given the saturating nature of the softmax, we conclude in Proposition 1 that even for extreme feature values in the limit, the output distribution of the model will not change anymore. In order to help the reader untangle the interdependence of upcoming results, we provide a flow chart in Figure 2.

Figure 2: Dependencies between theoretical results. Information in parentheses denotes the section in the document.

We first describe how to re-write a ReLU network - or any other network with piece-wise linear activation functions - as a piece-wise affine transformation, borrowing from Croce and Hein [2018], Hein et al. [2019]. We start with the common form of $f_{\boldsymbol{\theta}}$ as a series of affine transformations, interleaved with ReLU activation functions, which we will denote by ϕ:

$$f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{W}_L\, \phi\big(\,\mathbf{W}_{L-1}\, \phi\big(\ldots \phi\big(\,\mathbf{W}_1\,\mathbf{x} + \mathbf{b}_1\,\big)\ldots\big)$$
$$+ \mathbf{b}_{L-1}\,\big) + \mathbf{b}_L \quad (5)$$

In the following, let $f_{\boldsymbol{\theta}}^l(\mathbf{x})$ denote the output of layer l before applying an activation function. We now define a layer-specific diagonal matrix $\Phi_l \in \mathbb{R}^{n_l \times n_l}$ in the following way,
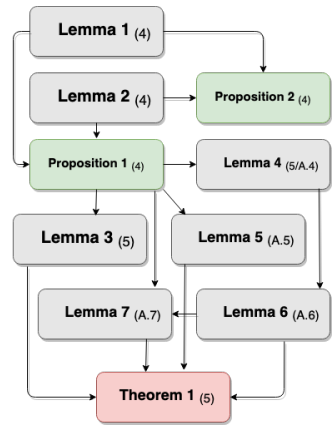
where n_l denotes the hidden units in layer l:

$$\Phi_l(\mathbf{x}) = \begin{bmatrix} \mathbb{1}(f_{\boldsymbol{\theta}}^l(\mathbf{x})_1 > 0) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbb{1}(f_{\boldsymbol{\theta}}^l(\mathbf{x})_{n_l} > 0) \end{bmatrix}$$

This allows us to rewrite Equation 5 by replacing the usage of ϕ with a matrix multiplication using Φ_l:

$$f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{W}_L \, \Phi_{L-1}(\mathbf{x}) \big(\mathbf{W}_{L-1} \, \Phi_{L-2}(\mathbf{x}) \big(\dots \\ \Phi_1(\mathbf{x}) \big(\mathbf{W}_1 \, \mathbf{x} + \mathbf{b}_1 \big) \dots \big) + \mathbf{b}_{L-1} \big) + \mathbf{b}_L \tag{6}$$

Finally, by distributing the matrix products inside-out (see Appendix A.2 for more detail), we can rewrite the network as a single affine transformation $f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{V}(\mathbf{x})\,\mathbf{x} + \mathbf{a}(\mathbf{x})$ with

$$\mathbf{V}(\mathbf{x}) = \mathbf{W}_L \left(\prod_{l=1}^{L-1} \Phi_l(\mathbf{x}) \, \mathbf{W}_{L-l} \right)$$

$$\mathbf{a}(\mathbf{x}) = \mathbf{b}_L + \sum_{l=1}^{L-1} \left(\prod_{l'=1}^{L-l} \mathbf{W}_{L+1-l'} \, \Phi_{L-l'}(\mathbf{x}) \right) \mathbf{b}_l$$

Note that the definition of $\mathbf{V}(\mathbf{x})$ corresponds to the Jacobian of $f_{\boldsymbol{\theta}}(\mathbf{x})$, meaning that $v_{cd} = \frac{\partial f_{\boldsymbol{\theta}}(\mathbf{x})_c}{\partial x_d}$ This is very useful, as it allows to quickly check whether a network $f_{\boldsymbol{\theta}}$ is component-wise strictly monotonic by checking $\mathbf{V}(\mathbf{x})$ for entries containing zeros. As Hein et al. [2019] show, this formulation can also be used to characterize a set of polytopes $\mathcal{Q} = \{Q_1, \dots, Q_M\}$ induced by $f_{\boldsymbol{\theta}}$ and that within each polytope, the function has a unique representation as an affine transformation. For this reason, we drop the dependence of $\mathbf{V}$ and $\mathbf{a}$ on $\mathbf{x}$ when we refer to a specific polytope. Such polytopes are constructed by first retrieving the half-spaces induced by each of the network's neurons and then intersecting all said half-spaces to generate convex regions or polytopes.[3] We are especially interested in polytopes that are unbounded in at least one direction. In this regard, the results of Croce and Hein [2018] and Hein et al. [2019] show that there is a finite number of polytopes corresponding to the given network, and their Lemma 3.1 proves the existence of at least one unbounded polytope. Furthermore, under a mild condition on $\mathbf{V}$, we can ascertain that $f_{\boldsymbol{\theta}}$ will be component-wise strictly monotonic on any polytope.

Lemma 1. *Suppose $f_{\boldsymbol{\theta}}$ is a ReLU network according to Definition 1. Then $f_{\boldsymbol{\theta}}$ is a component-wise strictly monotonic function on every of its polytopes $Q \in \mathcal{Q}$, as long as its corresponding matrix $\mathbf{V}$ has no zero entries.*

Proof. Let Q be one such polytope. As discussed, when restricted to Q, the network corresponds to an affine transformation $f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{V}\,\mathbf{x} + \mathbf{a}$ with $\mathbf{V} \in \mathbb{R}^{C \times D}$ and $\mathbf{a} \in \mathbb{R}^C$. $f_{\boldsymbol{\theta}}(\mathbf{x})_c$ thus corresponds to the dot product of the c-th row

of $\mathbf{V}$ and $\mathbf{x}$ plus the c-th element of $\mathbf{a}$. It follows that the partial derivative of $f_{\boldsymbol{\theta}}(\mathbf{x})_c$ with respect to a dimension d would then amount to the element v_{cd} in $\mathbf{V}$. This entails that, if $v_{cd} \neq 0$, at any point $\mathbf{x} \in Q$ the gradient will be always positive or always negative. $\qquad \square$

We note here that the component-wise strict monotonicity of $f_{\boldsymbol{\theta}}$ and softmax do not entail the same property for $p_{\boldsymbol{\theta}}$.[4] Nonetheless, the monotonic behaviour of $f_{\boldsymbol{\theta}}$ is sufficient to drive the logits to plus or minus infinity in the limit, a phenomenon that constrains the output of $p_{\boldsymbol{\theta}}$ as we scale a data sample away from training data.

We begin our investigation of behaviour in the limit by showing that if we scale a vector only in a single dimension, we eventually always remain within a unique PUP.

Lemma 2. *Let $\mathbf{x}' \in \mathbb{R}^D$ and $\mathcal{Q} = \{Q_1, \dots, Q_M\}$ be the finite set of polytopes generated by a network $f_{\boldsymbol{\theta}}$. Let $\boldsymbol{\alpha} \in \mathbb{R}^D$ be a vector s.t. $\forall d' \neq d, \alpha_{d'} = 1$. There exist a value $\beta > 0$ and $m \in 1, \dots, M$ such that for all $\alpha_d > \beta$, the product $\mathbf{x}' \odot \boldsymbol{\alpha}$ lies within Q_m.*

Proof. The proof mirrors the proof of Lemma 3.1 in Hein et al. [2019], so we only provide the intuition. By contradiction, suppose that there is no unique polytope and thus the point $\mathbf{x}' \odot \boldsymbol{\alpha}$ must traverse different polytopes as we scale up α_d. Since there are finitely many polytopes, eventually the same polytope Q_m will have to be traversed twice. Since the polytopes are convex, all the points on the line connecting the locations of where the boundary of Q_m was crossed the first and second time must lie within Q_m, but this contradicts the fact that the scaled point traverses different polytopes. $\qquad \square$

From here onward, we adapt the following shorthand to simplify notation: Given a scaling vector $\boldsymbol{\alpha} \in \mathbb{R}^D$ s.t. $\forall d' \neq d, \alpha_{d'} = 1$, we use $\mathcal{P}(\mathbf{x}', d)$ to denote the PUP that $\mathbf{x}'$ lands in when scaling it with α_d in the limit. This definition implies that we can only scale parallel to the basis vectors (for a discussion on how restrictive this is, see Section 6).

Finally, in the next lemma we establish that the output distribution converges to a fixed point using the l_2-norm of the gradient $\nabla_{\mathbf{x}} p_{\boldsymbol{\theta}}(y = c \,|\, \mathbf{x})$. Generally, in regions of the feature space where the classifier is predicting the same probability distribution over classes, small perturbations in the input $\mathbf{x}$ will not change the prediction. Therefore, the gradient in these regions w.r.t. the input will be small and

---

[3]We refer the reader to Appendix A.3 or Hein et al. [2019] for details on the construction, since it is not central to our reasoning.

[4]To see a counterexample, the reader can check that even assuming component-wise strict monotonicity for $f_{\boldsymbol{\theta}}$, if the matrix $\mathbf{V}$ associated to $f_{\boldsymbol{\theta}}$ on a specific polytope has a column d filled with the same value a, then the resulting $p_{\boldsymbol{\theta}}$ will have a gradient of 0 at dimension d, regardless of what class we are considering. This is because the partial derivatives of the softmax, when all multiplied by the same constant a, add up to zero.

potentially even correspond to the zero vector, with a norm of (or close to) zero.

Proposition 1 (Convergence of predictions in the limit). *Suppose that $f_{\boldsymbol{\theta}}$ is a ReLU-network. Let $\mathbf{x}' \in \mathbb{R}^D$, suppose $\boldsymbol{\alpha}$ is a scaling vector and that the associated PUP $\mathcal{P}(\mathbf{x}', d)$ has a corresponding matrix $\mathbf{V}$ with no zero entries. Then it holds that*

$$\forall c \in \mathcal{C}, \ \lim_{\alpha_d \to \infty} \left\| \nabla_{\mathbf{x}} p_{\boldsymbol{\theta}}(y = c \,|\, \mathbf{x}) \Big|_{\mathbf{x} = \boldsymbol{\alpha} \odot \mathbf{x}'} \right\|_2 = 0$$

The whole proof exceeds the available space and can be found in Appendix A.4, so we present the main intuitions here. Because of Lemma 2, we know the scaled point $\boldsymbol{\alpha} \odot \mathbf{x}'$ will end in a unique PUP. The assumption on $f_{\boldsymbol{\theta}}$ then triggers Lemma 1, from which we can infer that scaling the input in a single dimension leads all logits to $\pm\infty$. Because of the saturating property of the softmax, this will in turn provoke the output of $p_{\boldsymbol{\theta}}$ to converge to a fixed point.

As an aside, we now recast Theorem 3.1 in Hein et al. [2019] in our framework, showing that the model becomes increasingly certain in a single class, placing all its probability mass on it in the limit. The proof of this additional Proposition is in Appendix A.5.

Proposition 2. *Let $f_{\boldsymbol{\theta}}$ be ReLU network. Let $\mathbf{x}' \in \mathbb{R}^D$, suppose $\boldsymbol{\alpha}$ is a scaling vector and that the associated PUP $\mathcal{P}(\mathbf{x}', d)$ has a corresponding matrix $\mathbf{V}$ with no zero entries. Assume the d-th column of $\mathbf{V}$ has no duplicate entries. Then there exists a class c such that*

$$\lim_{\alpha_d \to \infty} \bar{\sigma}(f_{\boldsymbol{\theta}}(\boldsymbol{\alpha} \odot \mathbf{x}'))_c = 1$$

In conclusion, we have shown in this section that the output probabilities of ReLU networks are less and less sensitive to small perturbations of the input in the limit and, under the assumptions of Proposition 2, will converge to favor a single class, in which they appear to be very confident in. In the next section we prove that all other uncertainty metrics also converge to fixed values in the limit.

5 CONVERGENCE OF UNCERTAINTY ESTIMATION METRICS

In Proposition 1, we have established how the prediction of a model converges to a fixed point when feature values become extreme. We now want to show a similar property about the uncertainty estimation techniques introduced in Section 3.3. To this end, we have to establish how the predictions coming from multiple model instances interact, a point we analyze in Lemma 4. Then, we demonstrate how the uncertainty metrics also converge to a fixed value in the limit by proving the case for each of them in turn, before

bundling our results in Theorem 1. We start with the easiest metric, which also applies to a single model.

Lemma 3. *(Max. softmax probability) Suppose that $f_{\boldsymbol{\theta}}$ is a ReLU-network. Let $\mathbf{x}' \in \mathbb{R}^D$, suppose $\boldsymbol{\alpha}$ is a scaling vector and that the associated PUP $\mathcal{P}(\mathbf{x}', d)$ has a corresponding matrix $\mathbf{V}$ with no zero entries. It holds that*

$$\lim_{\alpha_d \to \infty} \left\| \nabla_{\mathbf{x}} \max_{c \in \mathcal{C}} \left(p_{\boldsymbol{\theta}}(y = c \,|\, \mathbf{x}) \right) \Big|_{\mathbf{x} = \boldsymbol{\alpha} \odot \mathbf{x}'} \right\|_2 = 0$$

Proof. The gradient of the $\max$ function will be a specific $\nabla_{\mathbf{x}} p_{\boldsymbol{\theta}}(y = c \,|\, \mathbf{x})$, which reduces this to the case already proven in Proposition 1. $\square$

Note that for this metric, the combination with Proposition 2 shows that the model be fully confident in a single class in the limit. For our following lemmas, we want to consider uncertainty scores that are based on multiple instances, e.g. different ensemble members or forward passes using re-sampled dropout masks. What all of these approaches have in common is that for every k, the network parameters $\boldsymbol{\theta}^{(k)}$ will differ, and thus also the polytopal tesselation of the feature space. Hence, we have to adjust our assumptions accordingly. For every instance k, let us denote the affine function on a polytope $Q^{(k)}$ as $f_{\boldsymbol{\theta}}^{(k)}(\mathbf{x}) = \mathbf{V}^{(k)} \mathbf{x} + \mathbf{a}^{(k)}$. In order for our previous strategy to hold, we now assume $\forall k \in 1, \ldots, K$ that $\mathcal{P}^{(k)}(\mathbf{x}', d)$ has a matrix $\mathbf{V}^{(k)}$ which does not have any zero entries. Note that even though this assumption has to hold for all k, this does not mean that the matrices have to be identical.

Lemma 4 (Convergence of aggregated predictions in the limit). *Suppose that $f_{\boldsymbol{\theta}}^{(1)}, \ldots, f_{\boldsymbol{\theta}}^{(K)}$ are ReLU networks. Let $\mathbf{x}' \in \mathbb{R}^D$, suppose $\boldsymbol{\alpha}$ is a scaling vector and that the for all k, the associated PUP $\mathcal{P}^{(k)}(\mathbf{x}', d)$ has a corresponding matrix $\mathbf{V}^{(k)}$ with no zero entries. It holds that*

$$\lim_{\alpha_d \to \infty} \left\| \nabla_{\mathbf{x}} \, \mathbb{E}_{p(\boldsymbol{\theta} \,|\, \mathcal{D})} \left[p_{\boldsymbol{\theta}}(y = c \,|\, \mathbf{x}) \right] \Big|_{\mathbf{x} = \boldsymbol{\alpha} \odot \mathbf{x}'} \right\|_2 = 0$$

The full proof of this lemma can be found in appendix section A.6. The analogous lemmas for the remaining uncertainty metrics are stated and proved in appendix sections A.7, A.8 and A.9. The proof strategy for all further metrics is to simplify and reduce the uncertainty metrics such that Lemma 4 or Proposition 1 can be applied. All of these result combined now pave the way for our central theorem.

Theorem 1 (Convergence of uncertainty level in the limit). *Suppose that $f_{\boldsymbol{\theta}}^{(1)}, \ldots, f_{\boldsymbol{\theta}}^{(K)}$ are ReLU networks. Let $\mathbf{x}' \in \mathbb{R}^D$, suppose $\boldsymbol{\alpha}$ is a scaling vector and that the for all k, the associated PUP $\mathcal{P}^{(k)}(\mathbf{x}', d)$ has a corresponding matrix $\mathbf{V}^{(k)}$ with no zero entries. Then, whenever uncertainty is measured via either of the following metrics*

1. *Maximum softmax probability*

2. *Class variance*

3. *Predictive entropy*

4. *Approximate mutual information*

the network(s) will converge to fixed uncertainty scores for $\mathbf{x}' \odot \boldsymbol{\alpha}$ in the limit of $\alpha_d \to \infty$

Proof. The four parts of the theorem are proven separately by lemmas 3, 5 (A.7), 6 (A.8) and 7 (A.9). □

What follows from this result is that methods based on multiple instances of ReLU classifiers will suffer from the aforementioned problem as long as uncertainty is estimated with one of the techniques listed above. Next we demonstrate how these assumptions and results apply on synthetic data.

6 SYNTHETIC DATA EXPERIMENTS

To illustrate our findings, we plot the uncertainty surfaces and the gradient magnitudes of different models and uncertainty metric pairings on the half moons dataset, which we generate using the corresponding function in the `scikit-learn` package [Pedregosa et al., 2011]. Detailed information about the procedure can be found in Appendix B along with additional plots.[5]

For a single network, we can observe in Figure 3a) that there exist vast open-ended regions of stable confidence, confirming the findings of Theorem 1. However, in the bottom part of Figure 3a) we can observe green regions with high gradient magnitude which do not seem to comply with our findings. In this case, we can see that these regions follow the decision boundaries. Due to the exponential function in the softmax, it is intuitive that small perturbation in these areas would have a large impact on the uncertainty score, resulting in a high gradient magnitude. But why does the magnitude not decrease in the limit as predicted by Theorem 1? We formulated our scaling vector $\boldsymbol{\alpha}$ in way that only allows scaling along one of the coordinate axes. Therefore, if the decision boundaries are not parallel to the axes, by scaling we eventually escape the green areas and arrive at an

area with gradient of magnitude zero. If the green regions were parallel to the axes then this would result in a violation of our main assumption. Traversing the input space parallel to a decision boundary in direction d will not influence the prediction within the polytope, meaning that there will be entries $v_{cd} = 0$.[6]

Turning to predictions aggregated from multiple network instances in Figures 3b-3d, we again observe large regions of constant uncertainty. The high-confidence region in the plots using mutual information displays a different behaviour from the others. As this metric aims to isolate epistemic uncertainty, it makes sense that uncertainty would be lowest around the training data, i.e. where the model is best specified. The character of the green regions in the bottom part of Figures 3c and 3d can again be explained by decision boundaries: In these cases, we have multiple instances with parameters $\boldsymbol{\theta}^{(k)}$, all with their own polytopal structure. When they overlap, the regions of the feature space where the assumption of our theorem is violated can either extend (Figure 3c) or shrink (Figure 3d), based on the diversity among instances. The fact that the anchored ensemble in Figure 3d does not exhibit such uniform regions of uncertainty like the vanilla ensemble could be explained by the fact that its training procedure encourages diversification between members. In turn, the difference between MC Dropout and ensemble models can be elucidated using recent insights that variational methods tend to only explore a single mode of the weight posterior $p(\boldsymbol{\theta} \,|\, \mathcal{D})$, while ensemble members often spread across multiple modes [Wilson and Izmailov, 2020].

Overall, we have seen that our theorem can explain why an overgeneralization of uncertainty scores beyond the training data results in failure in OOD detection. We also explored the cases in which our assumptions are violated, i.e. by multiple, diverse model instances. In such scenarios, identification of OOD samples could in theory succeed, but often fails to do so reliably, see e.g. Ovadia et al. [2019], Ulmer et al. [2020]. These insights can also help explain many other empirical findings in this regard on a variety of real-world datasets, e.g. Smith and Gal [2018], Kompa et al. [2020].

7 DISCUSSION

In the past sections, we have proven that a single model will produce very confident softmax probabilities and that even for models using multiple network instances like ensembling and MC Dropout, all the uncertainty metrics analyzed will tend to fixed scores on far away samples, generalizing along

---

[6]A decision boundary in a polytope is not the only way in which this assumption can be broken, but it still appears to hold reasonably often. For instance, just around 6.3% of plotted points in Figure 1 possess a matrix $\mathbf{V}$ with at least one zero entry - all located in the PUP in the top right corner.
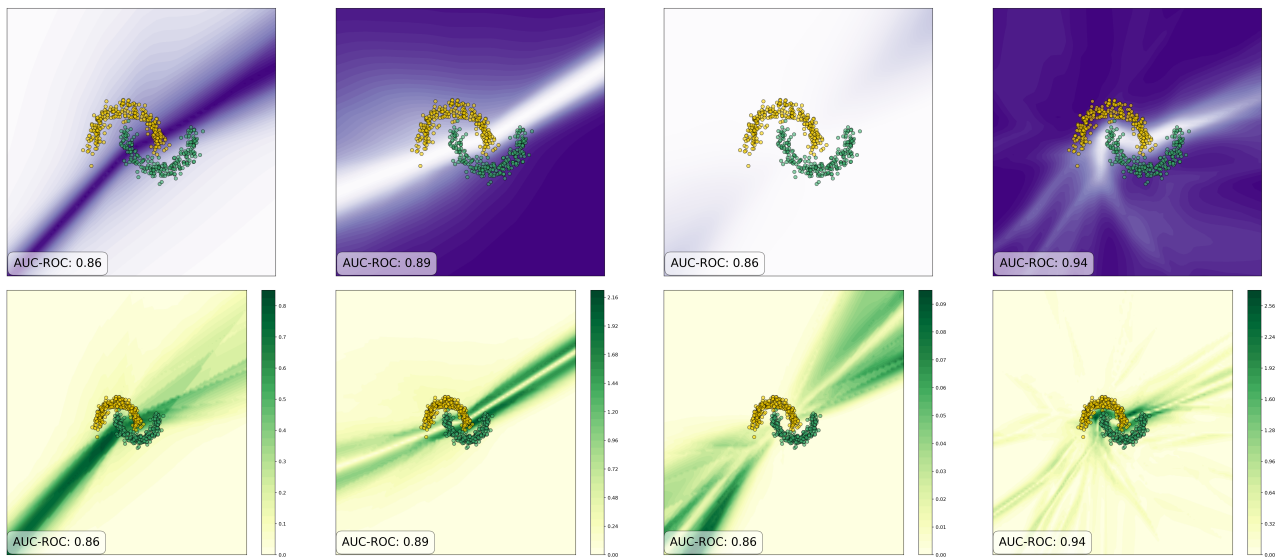
(a) Neural discriminator with maximum probability [Hendrycks and Gimpel, 2017].

(b) MC Dropout [Gal and Ghahramani, 2016] with mutual info. [Smith and Gal, 2018].

(c) Neural ensemble [Lakshminarayanan et al., 2017] with class variance.

(d) Anchored ensemble [Pearce et al., 2020] with mutual information [Smith and Gal, 2018].

Figure 3: Uncertainty on the half-moon dataset, including the binary classification AUC-ROC. (Top row) The uncertainty surface is represented with increasingly darker shades of purple, with white being the lowest uncertainty. Open-ended regions of static certainty appear across different models and metrics, bein extrapolated to unseen data (see 3a-3c); this phenomenon is less apparent in some instances (3d). (Bottom row) Increasing shades of green indicate the magnitude of the gradient of the uncertainty score w.r.t. the input. All metrics show open ended regions where the magnitude approaches zero.

open-ended polytopes induced by their architecture. The significance of this observation is that, albeit some modest success at OOD detection that might take place locally, the models we analyzed have an inherent overgeneralization bias: by extrapolating their level of uncertainty beyond the seen data, they hinder their ability to discern between in-distribution and OOD data.

For a single network, it can even be proven that the predictions attained on OOD samples is not just fixed but also unreasonably high in one class, as shown in our Proposition 2, which adapts the main result in Hein et al. [2019] to our framework. Our formulation shows that this result is also obtained when scaling a point along a single dimension.

Most of our results depend on an assumption on the matrix **V** corresponding to the PUP containing the OOD sample. Like discussed in Section 6, except for the case of decision boundaries that run parallel to a basis vector, this assumptions should rarely be broken for ReLU networks, as their are known to be resistant to the problem of vanishing gradients [Glorot et al., 2011]. We thus expect the behaviour described in our theoretical results to be very common, as confirmed by our experiments.

It remains to be explored how the stable level of certainty that we derive for all metrics relates to the degree of diversity of the underlying set of model instances, but given our experimental results it appears to be dependent on the level of

"disagreement" among them, i.e. when polytopes of different instances overlap to an increasing degree. Since it is hard to flag instances of OOD reliably this way, the aforementioned methods run a concrete risk of missing OOD samples, with potentially unintended, negative side-effects. These investigations bring us closer to a theoretically-motivated explanation to observations such as reported in Smith and Gal [2018], Ovadia et al. [2019], Kompa et al. [2020], Ulmer et al. [2020] and to enable the discovery of more effective methods.

Future research might be divided in two categories: efforts to solve the problem of OOD detection, and attempts at sharpening our theoretical understanding of the issue. The following approaches fall into the first category. One way consists of complementing neural discriminators with density-based approaches such as in Grathwohl et al. [2020]. Other lines of research try to have neural discriminators parametrize Dirichlet instead of categorical distributions [Malinin and Gales, 2018, Joo et al., 2020, Charpentier et al., 2020] or making models distance-aware [Liu et al., 2020] or supplementing the network architecture with Bayesian capabilities [Kristiadi et al., 2020]. In the second category we mention that more work is needed to cover the case of discrete features. Reasoning in the limit is not available for categorical or ordinal variables and one would probably have to resort to techniques that address directly the difference between the training and the new distribution. Furthermore, it remains

an open question whether the results presented here can be extended be extended to GELU activations [Hendrycks and Gimpel, 2016], which represent a continuous approximation of the ReLU function with the same asymptotic behavior. The GELU has recently become a very popular alternative in deep neural networks [Devlin et al., 2019, Lan et al., 2020, Brown et al., 2020]. Lastly, the similarity between lemmas used for Theorem 1 suggest that similar results could be derived for a whole family of uncertainty metrics. These investigations are left to future work.

Author Contributions

Both authors contributed equally in writing the paper and deriving theoretical results. Dennis Ulmer implemented the code to run experiments and create plots and illustrations in this work.

Acknowledgements

References

Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

John S Bridle. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neurocomputing*, pages 227–236. Springer, 1990.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. 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/`
`1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html`.

Bertrand Charpentier, Daniel Zügner, and Stephan Günnemann. Posterior network: Uncertainty estimation without OOD samples via density-based pseudo-counts. 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.

Francesco Croce and Matthias Hein. A randomized gradient-free attack on relu networks. In *German Conference on Pattern Recognition*, pages 215–227. Springer, 2018.

Alicia Curth, Patrick Thoral, Wilco van den Wildenberg, Peter Bijlstra, Daan de Bruin, Paul Elbers, and Mattia Fornasa. Transferring clinical prediction models across hospitals and electronic health record systems. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 605–621. Springer, 2019.

Alexandre de Brébisson and Pascal Vincent. An exploration of softmax alternatives belonging to the spherical loss family. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? *Structural safety*, 31(2):105–112, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1423. URL `https://doi.org/10.18653/v1/n19-1423`.

Yarin Gal. Uncertainty in deep learning. *University of Cambridge*, 1(3), 2016.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International conference on Machine Learning*, pages 1050–1059, 2016.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011.

Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 1321–1330, 2017.

Jianxing He, Sally L Baxter, Jie Xu, Jiming Xu, Xingtao Zhou, and Kang Zhang. The practical implementation of artificial intelligence technologies in medicine. *Nature medicine*, 25(1):30–36, 2019.

Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 41–50, 2019.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: A tutorial introduction. *arXiv preprint arXiv:1910.09457*, 2019.

Taejong Joo, Uijung Chung, and Min-Gwan Seo. Being bayesian about categorical probability. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 4950–4961. PMLR, 2020.

Matt Jordan, Justin Lewis, and Alexandros G. Dimakis. Provable certificates for adversarial examples: Fitting a ball in the union of polytopes. 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 14059–14069, 2019.

Benjamin Kompa, Jasper Snoek, and Andrew Beam. Empirical frequentist coverage of deep learning uncertainty quantification procedures. *arXiv preprint arXiv:2010.03039*, 2020.

Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Being bayesian, even just a bit, fixes overconfidence in relu networks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 5436–5446. PMLR, 2020. URL `http://proceedings.mlr.press/v119/kristiadi20a.html`.

Meelis Kull, Miquel Perelló-Nieto, Markus Kängsepp, Telmo de Menezes e Silva Filho, Hao Song, and Peter A. Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. 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 12295–12305, 2019. URL `https://proceedings.neurips.cc/paper/2019/hash/8ca01ea920679a0fe3728441494041b9-Abstract.html`.

Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pages 2810–2819, 2018. URL `http://proceedings.mlr.press/v80/kumar18a.html`.

Anirban Laha, Saneem Ahmed Chemmengath, Priyanka Agrawal, Mitesh Khapra, Karthik Sankaranarayanan, and Harish G Ramaswamy. On controllable sparse alternatives to softmax. In *Advances in Neural Information Processing Systems*, pages 6422–6432, 2018.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pages 6402–6413, 2017.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL `https://openreview.net/forum?id=H1eA7AEtvS`.

Max-Heinrich Laves, Sontje Ihler, Karl-Philipp Kortmann, and Tobias Ortmaier. Well-calibrated model uncertainty with temperature scaling for dropout variational inference. *4th workshop on Bayesian Deep Learning (NeurIPS 2019), Vancouver, Canada*, 2019.

Jeremiah Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in Neural Information Processing Systems*, 33, 2020.

Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. In *Advances in Neural Information Processing Systems*, pages 7047–7058, 2018.

Andre Martins and Ramon Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International Conference on Machine Learning*, pages 1614–1623, 2016.

John Alexander McDermid, Yan Jia, and Ibrahim Habli. Towards a framework for safety assurance of autonomous systems. In *Artificial Intelligence Safety 2019*, pages 1–7. CEUR Workshop Proceedings, 2019.

Jose G Moreno-Torres, Troy Raeder, RocíO Alaiz-RodríGuez, Nitesh V Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern recognition*, 45(1):521–530, 2012.

Jishnu Mukhoti, Andreas Kirsch, Joost van Amersfoort, Philip HS Torr, and Yarin Gal. Deterministic Neural Networks with Appropriate Inductive Biases Capture Epistemic and Aleatoric Uncertainty. *arXiv preprint arXiv:2102.11582*, 2021.

Eric T. Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Görür, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Kirill Neklyudov, Max Welling, Evgenii Egorov, and Dmitry P. Vetrov. Involutive MCMC: a unifying framework. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, pages 7273–7282, 2020.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, pages 13991–14002, 2019.

Tim Pearce, Felix Leibfried, and Alexandra Brintrup. Uncertainty in neural networks: Approximately bayesian ensembling. In *International Conference on Artificial Intelligence and Statistics*, pages 234–244, 2020.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

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.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

Lewis Smith and Yarin Gal. Understanding measures of uncertainty for adversarial example detection. In *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pages 560–569, 2018.

Dennis Ulmer, Lotta Meijerink, and Giovanni Cinà. Trust issues: Uncertainty estimation does not enable reliable ood detection on medical tabular data. In *Machine Learning for Health Workshop (NeurIPS 2020)*, pages 341–354. PMLR, 2020.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688, 2011.

Andrew Gordon Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective 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*, 2020.