Know Your Limits: Uncertainty Estimation with ReLU Classifiers Fails at Reliable OOD Detection (Supplementary material)

Dennis Ulmer[1] **Giovanni Cinà**[2]

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

A ADDITIONAL PROOFS

This appendix section contains additional proofs and derivations that could not be included in the main paper due to spatial constraints.

A.1 CONNECTION BETWEEN SOFTMAX AND SIGMOID

In this section we briefly outline the connection between the softmax and the sigmoid function, which was originally shown in Bridle [1990]. Let the sigmoid function be defined as

$$\sigma(x) = \frac{\exp(x)}{1 + \exp(x)}$$

and softmax according to the definition Section 3.3. The output of $f_{\boldsymbol{\theta}}$ in a multi-class classification problem with C classes corresponds to a C-dimensional column vector that is based on an affine transformation of the network's last intermediate hidden representation $\mathbf{x}_L$, such that $f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{W}_L\,\mathbf{x}_L$.[1] Correspondingly, the output of $f_{\boldsymbol{\theta}}$ for a single class c can be written as the dot product between $\mathbf{x}_L$ and the corresponding row vector of $\mathbf{W}_L$ denoted as $\mathbf{w}_L^{(c)}$, such that $f_{\boldsymbol{\theta}}(\mathbf{x})_c \equiv \mathbf{w}_L^{(c)T}\,\mathbf{x}_L$. For a classification problem with $C = 2$ classes, we can now rewrite the softmax probabilities in the following way:[2]

$$p_{\boldsymbol{\theta}}(y = 1|\,\mathbf{x}) = \frac{\exp(\mathbf{w}_L^{(1)T}\,\mathbf{x}_L)}{\exp(\mathbf{w}_L^{(0)T}\,\mathbf{x}_L) + \exp(\mathbf{w}_L^{(1)T}\,\mathbf{x}_L)}$$

Subtracting a constant from the weight term inside the exponential function does not change the output of the softmax function. Using this property, we can show the sigmoid function to be a special case of the softmax for binary classification:

$$\begin{aligned} &p_{\boldsymbol{\theta}}(y = 1|\,\mathbf{x}) \\ &= \frac{\exp((\mathbf{w}_L^{(1)} - \mathbf{w}_L^{(0)})^T\,\mathbf{x}_L)}{\exp((\mathbf{w}_L^{(0)} - \mathbf{w}_L^{(0)})^T\,\mathbf{x}_L) + \exp((\mathbf{w}_L^{(1)} - \mathbf{w}_L^{(0)})^T\,\mathbf{x}_L)} \\ &= \frac{\exp((\mathbf{w}_L^{(1)} - \mathbf{w}_L^{(0)})^T\,\mathbf{x}_L)}{1 + \exp((\mathbf{w}_L^{(1)} - \mathbf{w}_L^{(0)})^T\,\mathbf{x}_L} = \frac{\exp(\mathbf{w}_L^{*T}\,\mathbf{x}_L)}{1 + \exp(\mathbf{w}_L^{*T}\,\mathbf{x}_L)} \end{aligned}$$

where $\mathbf{w}_L^* = \mathbf{w}_L^{(1)} - \mathbf{w}_L^{(0)}$ corresponds to the new parameter vector which is used to parametrize a single output unit for a network in the binary classification setting.

A.2 LINEARIZATION OF RELU NETWORKS

In the section we give a more detailed version of the derivation of the linearization $f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{V}(\mathbf{x})\,\mathbf{x} + \mathbf{a}(\mathbf{x})$ with

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

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

We start from Equation 6:

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

To make the steps more intuitive and to retain readability, we illustrate the necessary steps on a simple three layer network:

[1]The bias term $\mathbf{b}_L$ was omitted here for clarity.

[2]The following argument holds without loss of generality for $p_{\boldsymbol{\theta}}(y = 0|\,\mathbf{x})$.

$$
\begin{aligned}
f_{\boldsymbol{\theta}}(\mathbf{x}) &= \mathbf{W}_3\,\Phi_2(\mathbf{x})\big(\,\mathbf{W}_2\,\Phi_1(\mathbf{x})\big(\,\mathbf{W}_1\,\mathbf{x}+\mathbf{b}_1\big)+\mathbf{b}_2\big)+\mathbf{b}_3 \\
&= \mathbf{W}_3\,\Phi_2(\mathbf{x})\big(\,\mathbf{W}_2\,\Phi_1(\mathbf{x})\,\mathbf{W}_1\,\mathbf{x}+\mathbf{W}_2\,\Phi_1(\mathbf{x})\,\mathbf{b}_1\big) \\
&\quad +\mathbf{b}_2\big)+\mathbf{b}_3 \\
&= \underbrace{\mathbf{W}_3\,\Phi_2(\mathbf{x})\,\mathbf{W}_2\,\Phi_1(\mathbf{x})\,\mathbf{W}_1}_{=\mathbf{V}(x)}\,\mathbf{x} \\
&\quad + \underbrace{\mathbf{W}_3\,\Phi_2(\mathbf{x})\,\mathbf{W}_2\,\Phi_1(\mathbf{x})\,\mathbf{b}_1+\mathbf{W}_3\,\Phi_2(\mathbf{x})\,\mathbf{b}_2+\mathbf{b}_3}_{=\mathbf{a}(\mathbf{x})}
\end{aligned}
$$

which we can identify as the parts of the affine transformation above.

A.3 CONSTRUCTION OF POLYTOPAL REGIONS

In this section, we reiterate the reasoning by Hein et al. [2019] behind the construction the polytopal regions. For this purpose, the authors define an additional diagonal matrix $\Delta_l(\mathbf{x})$ per layer l:

$$
\Delta_l(\mathbf{x}) = \begin{bmatrix} \operatorname{sign}(f^l_{\boldsymbol{\theta}}(\mathbf{x})_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \operatorname{sign}(f^l_{\boldsymbol{\theta}}(\mathbf{x})_{n_l}) \end{bmatrix}
$$

Together with the linearization of the network at $\mathbf{x}$ explained in Appendix A.2, this is used to define a set of half-spaces for every neuron in the network:

$$
\mathcal{H}_{l,i}(\mathbf{x}) = \Big\{\, \mathbf{z}\in\mathbb{R}^d \;\Big|\; \Delta_l(\mathbf{x})\big(\,\mathbf{V}_l(\mathbf{x})_i\,\mathbf{z}+\mathbf{a}_l(\mathbf{x})_i\big) \geq 0 \Big\}
$$

Here, $\mathbf{V}_l(\mathbf{x})_i$ and $\mathbf{b}_l(\mathbf{x})_i$ denote the parts of the affine transformation obtained for the i-th neuron of the l-th layer, so the i-th row vector in $\mathbf{V}_l(\mathbf{x})$ and the i-th scalar in $\mathbf{b}_l(\mathbf{x})$, respectively. Finally, the polytope Q containing $\mathbf{x}$ is obtained by taking the intersection of all half-spaces induced by every neuron in the network:

$$
Q(\mathbf{x}) = \bigcap_{l\in 1,\ldots,L} \bigcap_{i\in 1,\ldots,n_l} \mathcal{H}_{l,i}(\mathbf{x})
$$

A.4 PROOF OF PROPOSITION 1

We proceed to analyze the behaviour of gradients in the limit via two more lemmas; First, we establish the saturating property of the softmax in Lemma 9, i.e. the model doesn't change its decision anymore in the limit.

Lemma 9. *Let $c, c' \in \mathcal{C}$ be two arbitrary classes. It then holds for their corresponding output components (logits) that*

$$
\lim_{f_{\boldsymbol{\theta}}(\mathbf{x})_c\to\pm\infty} \frac{\partial}{\partial f_{\boldsymbol{\theta}}(\mathbf{x})_{c'}} \bar{\sigma}(f_{\boldsymbol{\theta}}(\mathbf{x}))_c = 0 \tag{1}
$$

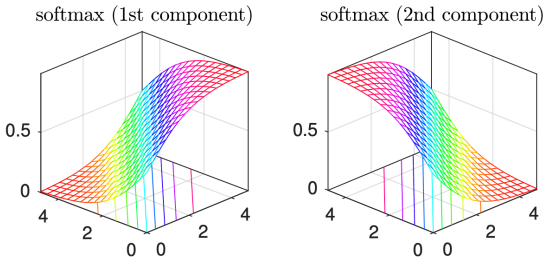

Figure 1: Illustration taken from the work of Gao and Pavel [2017], illustrating the interplay of softmax probabilities between components for $C = 2$ in $\mathbb{R}^2$.

Proof. Here, we first begin by evaluating the derivative of one component of the function w.r.t to an arbitrary component:

$$
\begin{aligned}
&\frac{\partial}{\partial f_{\boldsymbol{\theta}}(\mathbf{x})_{c'}} \bar{\sigma}(f_{\boldsymbol{\theta}}(\mathbf{x}))_c = \frac{\partial}{\partial f_{\boldsymbol{\theta}}(\mathbf{x})_{c'}} \frac{\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_c)}{\sum_{c''\in\mathcal{C}}\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{c''})} \\
&= \frac{\mathbb{1}(c=c')\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_c)}{\sum_{c''\in\mathcal{C}}\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{c''})} - \frac{\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_c)\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{c'})}{\big(\sum_{c''\in\mathcal{C}}\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{c''})\big)^2}
\end{aligned}
$$

This implies that $\frac{\partial}{\partial f_{\boldsymbol{\theta}}(\mathbf{x})_{c'}} \bar{\sigma}(f_{\boldsymbol{\theta}}(\mathbf{x}))_c =$

$$
\begin{cases}
-\dfrac{\exp(2f_{\boldsymbol{\theta}}(\mathbf{x})_c)}{\big(\sum_{c''\in\mathcal{C}}\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{c''})\big)^2} & \\
+\dfrac{\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_c)}{\sum_{c''\in\mathcal{C}}\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{c''})} & \text{If } c = c' \\
-\dfrac{\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_c + f_{\boldsymbol{\theta}}(\mathbf{x})_{c'})}{\big(\sum_{c''\in\mathcal{C}}\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{c''})\big)^2} & \text{If } c \neq c'
\end{cases} \tag{2}
$$

or more compactly:

$$
\frac{\partial}{\partial f_{\boldsymbol{\theta}}(\mathbf{x})_{c'}} \bar{\sigma}(f_{\boldsymbol{\theta}}(\mathbf{x}))_c = \bar{\sigma}(f_{\boldsymbol{\theta}}(\mathbf{x}))_c\big(\mathbb{1}(c=c') - \bar{\sigma}(f_{\boldsymbol{\theta}}(\mathbf{x}))_{c'}\big)
$$

Based on Equation 2, we can now investigate the asymptotic behavior for $f_{\boldsymbol{\theta}}(\mathbf{x})_c \to \infty$ more easily, starting with the $c = c'$ case:

$$
\begin{aligned}
&\lim_{f_{\boldsymbol{\theta}}(\mathbf{x})_c\to\infty} \frac{\partial}{\partial f_{\boldsymbol{\theta}}(\mathbf{x})_{c'}} \bar{\sigma}(f_{\boldsymbol{\theta}}(\mathbf{x}))_c \\
&= \underbrace{\lim_{f_{\boldsymbol{\theta}}(\mathbf{x})_c\to\infty} -\frac{\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_c)}{\sum_{c''\in\mathcal{C}}\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{c''})} \frac{\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_c)}{\sum_{c''\in\mathcal{C}}\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{c''})}}_{\text{-1}} \\
&+ \underbrace{\lim_{f_{\boldsymbol{\theta}}(\mathbf{x})_c\to\infty} \frac{\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_c)}{\sum_{c''\in\mathcal{C}}\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{c''})}}_{1} = 0
\end{aligned} \tag{3}
$$

With the numerator and denominator being dominated by the exponentiated $f_{\boldsymbol{\theta}}(\mathbf{x})_c$ in Equation 3, the first term will

tend to -1, while the second term will tend to 1, resulting in a derivative of 0. The $c \neq c'$ can be analyzed the following way:

$$\begin{aligned}
&\lim_{f_{\boldsymbol{\theta}}(\mathbf{x})_c \to \infty} \frac{\partial}{\partial f_{\boldsymbol{\theta}}(\mathbf{x})_{c'}} \bar{\sigma}(f_{\boldsymbol{\theta}}(\mathbf{x}))_c \\
&= \underbrace{\lim_{f_{\boldsymbol{\theta}}(\mathbf{x})_c \to \infty} \left(- \frac{\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_c)}{\sum_{c'' \in \mathcal{C}} \exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{c''})} \right)}_{-1} \\
&\cdot \underbrace{\lim_{f_{\boldsymbol{\theta}}(\mathbf{x})_c \to \infty} \left(\frac{\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{c'})}{\sum_{c'' \in \mathcal{C}} \exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{c''})} \right)}_{0} = 0
\end{aligned} \tag{4}$$

Again, we factorize the fraction in Equation 4 into the product of two softmax functions, one for component c, one for c'. The first factor will again tend to -1 as in the other case, however the second will approach 0, as only the sum in the denominator will approach infinity. As the limit of a product is the products of its limits, this lets the whole expression approach 0 in the limit.

When $f_{\boldsymbol{\theta}}(\mathbf{x})_c \to -\infty$, both cases approach 0 due to the exponential function, which proves the lemma. □

How to interplay between different softmax components produces zero gradients in the limit is illustrated in Figure 1. In Lemma 10, we compare the rate of growth of different components of $p_{\boldsymbol{\theta}}$. We show that for the decomposed function $p_{\boldsymbol{\theta}}$, the rate at which the softmax function converges to its output distribution in the limit outpaces the change in the underlying logits w.r.t. the network input.

Lemma 10. *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*

$$\begin{aligned}
\forall c' \in \mathcal{C}, \lim_{\alpha_d \to \infty} &\left(\frac{\partial}{\partial f_{\boldsymbol{\theta}}(\mathbf{x})_{c'}} \bar{\sigma}(f_{\boldsymbol{\theta}}(\mathbf{x}))_c \right)^{-1} \Bigg|_{\mathbf{x} = \boldsymbol{\alpha} \odot \mathbf{x}'} \\
&- \left(\frac{\partial}{\partial x_d} f_{\boldsymbol{\theta}}(\mathbf{x})_{c'} \right) \Bigg|_{\mathbf{x} = \boldsymbol{\alpha} \odot \mathbf{x}'} = \infty
\end{aligned} \tag{5}$$

Proof. We evaluate the first term of Equation 5 to show that it grows exponentially in the limit. By Lemma 2 we know that in the limit $\alpha_d \to \infty$ the vector $\boldsymbol{\alpha} \odot \mathbf{x}'$ will remain within $\mathcal{P}(\mathbf{x}', d)$. Since the matrix associated with this PUP has no zero entries, we know by Lemma 1 that the gradient of $f_{\boldsymbol{\theta}}(\mathbf{x})_c$ on dimension d is either always positive or negative, hence $f_{\boldsymbol{\theta}}(\mathbf{x})_c \to \pm\infty$. Given Lemma 9 describing the asymptotic behavior in the limit, it follows that

$$\lim_{f_{\boldsymbol{\theta}}(\mathbf{x})_c \to \pm\infty} \left(\frac{\partial}{\partial f_{\boldsymbol{\theta}}(\mathbf{x})_{c'}} \bar{\sigma}(f_{\boldsymbol{\theta}}(\mathbf{x}))_c \right)^{-1} = \infty$$

where we can see that the result is a symmetrical function displaying exponential growth in the limit of $f_{\boldsymbol{\theta}}(\mathbf{x})_c \to \pm\infty$. We now show that because we assumed $f_{\boldsymbol{\theta}}$ to be a neural network consisting of L affine transformations with ReLU activation functions, the output of the final layer is only going to be a linear combination of its inputs.[3] This can be proven by induction. Let us first look at the base case $L = 1$. In the rest of this proof, we denote $\mathbf{x}_l$ as the input to layer l, with $\mathbf{x}_1 \equiv \mathbf{x}$, and $\mathbf{W}_l, \mathbf{b}_l$ the corresponding layer parameters. $\mathbf{a}_l$ signifies the result of the affine transformation that is then fed into the activation function.

$$\begin{aligned}
f_{\boldsymbol{\theta}}(\mathbf{x}) &= \phi(\mathbf{a}_1) = \phi(\mathbf{W}_1 \mathbf{x}_1 + \mathbf{b}_1) \\
\frac{\partial f_{\boldsymbol{\theta}}(\mathbf{x})}{\partial \mathbf{x}_1} &= \frac{\phi(\mathbf{a}_1)}{\partial \mathbf{a}_1} \frac{\partial \mathbf{a}_1}{\partial \mathbf{x}_1} = \mathbb{1}(\mathbf{x}_1 > \mathbf{0})^T \mathbf{W}_1 \\
\frac{\partial f_{\boldsymbol{\theta}}(\mathbf{x})}{\partial x_{1d}} &= \mathbb{1}(x_d > 0) w_{1d}
\end{aligned} \tag{6}$$

where $\mathbb{1}(\mathbf{x}_1 > \mathbf{0}) = [\mathbb{1}(x_{11} > 0), \ldots, \mathbb{1}(x_{1d} > 0)]^T$, w_{1d} denotes the d-th column of $\mathbf{W}_1$. This is a linear function, which proves the base case. Let now $\frac{\partial \mathbf{x}_l}{\partial \mathbf{x}_1}$ denote the partial derivative of the input to the l-th layer w.r.t to the input and suppose that it is linear by the inductive hypothesis. Augmenting the corresponding network by another linear adds another term akin to the second expression in Equation 6 to the chain of partial derivatives:

$$\frac{\partial \mathbf{x}_{l+1}}{\partial \mathbf{x}_1} = \frac{\partial \mathbf{x}_{l+1}}{\partial \mathbf{x}_l} \frac{\partial \mathbf{x}_l}{\partial \mathbf{x}_1} \tag{7}$$

which is also a linear function of, proving the induction step. Because we know that both terms of the product in Equation 7 are linear, the second term of the Equation 5 is as well. Together with the previous insight that the first term is exponential, this implies that it will outgrow the second in the limit, creating an infinitively-wide gap between them and thereby proving the lemma. □

Equipped with the results of Lemmas 9 and 10, we can finally prove the proposition:

Proof. We show that one scalar factor contained in the factorization of the gradient $\nabla_{\mathbf{x}} p_{\boldsymbol{\theta}}(y = c | \mathbf{x})$ tends to zero under the given assumptions, having the whole gradient become the zero vector in the limit. We begin by again factorizing the gradient $\nabla_{\mathbf{x}} p_{\boldsymbol{\theta}}(y = c | \mathbf{x})$ using the multivariate chain rule:

$$\nabla_{\mathbf{x}} p_{\boldsymbol{\theta}}(y = c | \mathbf{x}) = \sum_{c'=1}^{C} \frac{\partial}{\partial f_{\boldsymbol{\theta}}(\mathbf{x})_{c'}} \bar{\sigma}(f_{\boldsymbol{\theta}}(\mathbf{x}))_c \cdot \nabla_{\mathbf{x}} f_{\boldsymbol{\theta}}(\mathbf{x})_{c'} \tag{8}$$

[3] Here we make the argument for the whole function $f_{\boldsymbol{\theta}} : \mathbb{R}^D \to \mathbb{R}^C$, but the conclusions also applies to every output component of the function $f_{\boldsymbol{\theta}}(\mathbf{x})_c$.

By Lemma 1 and 2 we know that $f_{\boldsymbol{\theta}}$ is a component-wise strictly monotonic function on $\mathcal{P}(\mathbf{x}', d)$, which implies for the limit of $\alpha_d \to \infty$ that $\forall c \in \mathcal{C} : f_{\boldsymbol{\theta}}(\mathbf{x})_c \to \pm\infty$. Then, Lemma 9 implies that the first factor of every part in the sum of Equation 8 will tend to zero in the limit. Lemma 10 ensures that the first factor approximates zero quicker than every component of the gradient $\nabla_{\mathbf{x}} f_{\boldsymbol{\theta}}(\mathbf{x})_{c'}$ potentially approaching infinity, causing the product to result in the zero vector. As this results in a sum over C zero vectors in the limit, this proves the lemma. □

A.5 PROOF OF PROPOSITION 2

Proof. We start by rewriting the softmax probability for the c-th logit:

$$\begin{aligned}\bar{\sigma}(f_{\boldsymbol{\theta}}(\mathbf{x}))_c &= \frac{\exp(f_{\boldsymbol{\theta}}(\mathbf{x})_c)}{\sum_{c' \in \mathcal{C}} \exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{c'})} \\ &= 1 - \frac{\sum_{c'' \in \mathcal{C} \setminus \{c\}} \exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{c''})}{\sum_{c' \in \mathcal{C}} \exp(f_{\boldsymbol{\theta}}(\mathbf{x})_{c'})}\end{aligned}$$

By Lemma 1 and 2 we have shown that $f_{\boldsymbol{\theta}}$ is a component-wise strictly monotonic function on $\mathcal{P}(\mathbf{x}', d)$, so we know that $\forall c' \in \mathcal{C} : f_{\boldsymbol{\theta}}(\mathbf{x})_{c'} \to \pm\infty$ as $\alpha_d \to \infty$. We now treat the two limits $\pm\infty$ in order. Because of the assumption that d-column of $\mathbf{V}$ has no duplicate entries, this implies that there must be a $c \in \mathcal{C}$ s.t. $\forall c' \neq c : \; v_{cd} > v_{c'd}$. Thus, in the limit of $f_{\boldsymbol{\theta}}(\mathbf{x})_c \to \infty$, the sum in the *denominator* of the fraction including the logit of c will tend to infinity faster than the the sum in the *numerator* not including c's logit, and thus the fraction itself will tend to 0, proving this case. In the case of $f_{\boldsymbol{\theta}}(\mathbf{x})_c \to -\infty$, the *numerator* of the fraction will tend to 0 faster than the *denominator*, having the fraction approach 0 in the limit as well, proving the second case and therefore the lemma. □

A.6 PROOF OF LEMMA 4

Proof.

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

Linearity of gradient:

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

Utilize Jensen's inequality $\phi(\mathbb{E}[\mathbf{x}]) \leq \mathbb{E}[\phi(\mathbf{x})]$ as l_2-norm is a convex function and Proposition 1:

$$\leq \lim_{\alpha \to \infty} \mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})} \Bigg[\underbrace{\left|\left| \nabla_{\mathbf{x}} p_{\boldsymbol{\theta}}(y = c|\,\mathbf{x}) \Big|_{\mathbf{x} = \boldsymbol{\alpha} \odot \mathbf{x}'} \right|\right|_2}_{=0 \text{ (Proposition 1)}} \Bigg] = 0$$

Because the last expression is an upper bound to the original expression and the l_2 norm is lower-bounded by 0, this proves the lemma. □

A.7 PROOF OF LEMMA 5

Lemma 5. *(Asymptotic behavior with softmax variance) 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 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*

$$\lim_{\alpha_d \to \infty} \Bigg|\Bigg| \nabla_{\mathbf{x}} \frac{1}{C} \sum_{c=1}^{C} \mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})} \left[\Big(p_{\boldsymbol{\theta}}(y = c|\,\mathbf{x}) \Big)^2 \right] - \mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})} \left[p_{\boldsymbol{\theta}}(y = c|\,\mathbf{x}) \right]^2 \Big|_{\mathbf{x} = \boldsymbol{\alpha} \odot \mathbf{x}'} \Bigg|\Bigg|_2 = 0$$

Proof.

$$\lim_{\alpha \to \infty} \Bigg|\Bigg| \nabla_{\mathbf{x}} \frac{1}{C} \sum_{c=1}^{C} \mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})} \left[\Big(p_{\boldsymbol{\theta}}(y = c|\,\mathbf{x}) \Big)^2 \right] - \mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})} \left[p_{\boldsymbol{\theta}}(y = c|\,\mathbf{x}) \right]^2 \Big|_{\mathbf{x} = \boldsymbol{\alpha} \odot \mathbf{x}'} \Bigg|\Bigg|_2$$

Linearity of gradient:

$$= \lim_{\alpha_d \to \infty} \Bigg|\Bigg| \frac{1}{C} \sum_{c=1}^{C} \nabla_{\mathbf{x}} \mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})} \left[\Big(p_{\boldsymbol{\theta}}(y = c|\,\mathbf{x}) \Big)^2 \right] - \nabla_{\mathbf{x}} \mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})} \left[p_{\boldsymbol{\theta}}(y = c|\,\mathbf{x}) \right]^2 \Big|_{\mathbf{x} = \boldsymbol{\alpha} \odot \mathbf{x}'} \Bigg|\Bigg|_2$$

Apply triangle inequality $||x + y|| \leq ||x|| + ||y||$ to sum over all c:

$$\leq \lim_{\alpha_d \to \infty} \frac{1}{C} \sum_{c=1}^{C} \Bigg|\Bigg| \nabla_{\mathbf{x}} \mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})} \left[\Big(p_{\boldsymbol{\theta}}(y = c|\,\mathbf{x}) \Big)^2 \right] - \nabla_{\mathbf{x}} \mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})} \left[p_{\boldsymbol{\theta}}(y = c|\,\mathbf{x}) \right]^2 \Big|_{\mathbf{x} = \boldsymbol{\alpha} \odot \mathbf{x}'} \Bigg|\Bigg|_2$$

On the first term use linearity of gradients and apply chain rule, do it in the reverse order on the second term:

$$\begin{aligned} &= \lim_{\alpha_d \to \infty} \\ &\frac{1}{C} \sum_{c=1}^{C} \Bigg|\Bigg| \mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})} \Bigg[2 p_{\boldsymbol{\theta}}(y = c|\,\mathbf{x}) \underbrace{\nabla_{\mathbf{x}} p_{\boldsymbol{\theta}}(y = c|\,\mathbf{x})}_{=\mathbf{0} \text{ (Proposition 1)}} \Big|_{\mathbf{x} = \boldsymbol{\alpha} \odot \mathbf{x}'} \Bigg] \\ &- \left(2 \mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})} \left[p_{\boldsymbol{\theta}}(y = c|\,\mathbf{x}) \right] \right) \\ &\cdot \mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})} \Bigg[\underbrace{\nabla_{\mathbf{x}} p_{\boldsymbol{\theta}}(y = c|\,\mathbf{x})}_{=\mathbf{0} \text{ (Proposition 1)}} \Big|_{\mathbf{x} = \boldsymbol{\alpha} \odot \mathbf{x}'} \Bigg] \Bigg|\Bigg|_2 = 0 \end{aligned}$$

We can see that due to an intermediate result of Proposition 1, i.e. that $\nabla_{\mathbf{x}} p_{\boldsymbol{\theta}}(y = c|\,\mathbf{x})$ approaches the zero vector in the limit, the innermost gradients tend to zero, bringing the whole expression to 0.

Because the final is an upper bound to the original expression and because the l_2 norm has a lower bound of 0, this proves the lemma. □

A.8 PROOF OF LEMMA 6

Lemma 6. *(Asymptotic behavior for predictive entropy) 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 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*

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

Proof.

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

$$= \lim_{\alpha_d \to \infty} \left|\left| \nabla_{\mathbf{x}} \left(\sum_{c=1}^{C} \mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})}\left[p_{\boldsymbol{\theta}}(y = c|\,\mathbf{x})\right] \cdot \log\left(\mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})}\left[p_{\boldsymbol{\theta}}(y = c|\,\mathbf{x})\right]\right)\right)\Big|_{\mathbf{x}=\boldsymbol{\alpha}\,\odot\,\mathbf{x}'} \right|\right|_2$$

Linearity of gradient:

$$= \lim_{\alpha_d \to \infty} \left|\left| \sum_{c=1}^{C} \nabla_{\mathbf{x}} \left(\mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})}\left[p_{\boldsymbol{\theta}}(y = c|\,\mathbf{x})\right] \cdot \log\left(\mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})}\left[p_{\boldsymbol{\theta}}(y = c|\,\mathbf{x})\right]\right)\right)\Big|_{\mathbf{x}=\boldsymbol{\alpha}\,\odot\,\mathbf{x}'} \right|\right|_2$$

Apply product rule:

$$= \lim_{\alpha_d \to \infty} \left|\left| \left(\sum_{c=1}^{C} \mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})}\left[p_{\boldsymbol{\theta}}(y = c|\,\mathbf{x})\right] \cdot \left(\mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})}\left[p_{\boldsymbol{\theta}}(y = c|\,\mathbf{x})\right]\right)^{-1} \cdot \nabla_{\mathbf{x}} \left(\mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})}\left[p_{\boldsymbol{\theta}}(y = c|\,\mathbf{x})\right]\right) + \nabla_{\mathbf{x}} \left(\mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})}\left[p_{\boldsymbol{\theta}}(y = c|\,\mathbf{x})\right]\right) \cdot \log\left(\mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})}\left[p_{\boldsymbol{\theta}}(y = c|\,\mathbf{x})\right]\right)\right)\Big|_{\mathbf{x}=\boldsymbol{\alpha}\,\odot\,\mathbf{x}'} \right|\right|_2$$

Factor out gradient:

$$= \lim_{\alpha_d \to \infty} \left|\left| \sum_{c=1}^{C} \nabla_{\mathbf{x}} \mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})}\left[p_{\boldsymbol{\theta}}(y = c|\,\mathbf{x})\right] \cdot \left(1 + \log\left(\mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})}\left[p_{\boldsymbol{\theta}}(y = c|\,\mathbf{x})\right]\right)\right)\Big|_{\mathbf{x}=\boldsymbol{\alpha}\,\odot\,\mathbf{x}'} \right|\right|_2$$

Apply triangle inequality to sum over all c:

$$\leq \lim_{\alpha_d \to \infty} \sum_{c=1}^{C} \left|\left| \nabla_{\mathbf{x}} \mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})}\left[p_{\boldsymbol{\theta}}(y = c|\,\mathbf{x})\right] \cdot \left(1 + \log\left(\mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})}\left[p_{\boldsymbol{\theta}}(y = c|\,\mathbf{x})\right]\right)\right)\Big|_{\mathbf{x}=\boldsymbol{\alpha}\,\odot\,\mathbf{x}'} \right|\right|_2$$

As the log expectation just evaluates to a scalar, it can be pulled out of the norm and we can apply Lemma 4

$$= \lim_{\alpha_d \to \infty} \sum_{c=1}^{C} \underbrace{\left(1 + \log\left(\mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})}\left[p_{\boldsymbol{\theta}}(y = c|\,\mathbf{x})\right]\right)\right)}_{\text{Scalar}} \cdot \underbrace{\left|\left| \nabla_{\mathbf{x}} \mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})}\left[p_{\boldsymbol{\theta}}(y = c|\,\mathbf{x})\right]\Big|_{\mathbf{x}=\boldsymbol{\alpha}\,\odot\,\mathbf{x}'} \right|\right|_2}_{=0 \text{ (Lemma 4)}} = 0$$

As the final result is an upper bound to the original expression and is lower-bounded by 0 due to the l_2 norm, this proves the lemma. □

A.9 PROOF OF LEMMA 7

Lemma 7. *(Asymptotic behavior for approximate mutual information) 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 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*

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

Proof.

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

Linearity of gradients:

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

Linearity of gradients on second part of difference:

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

Applying chain rule and intermediate result of Proposition 1:

$$= \lim_{\alpha_d \to \infty} \Bigg|\Bigg| \nabla_{\mathbf{x}} \mathbb{H}\Big[\mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})}\Big[p_{\boldsymbol{\theta}}(y|\,\mathbf{x})\Big]\Big]\Big|_{\mathbf{x}=\boldsymbol{\alpha}\,\odot\,\mathbf{x}'} - \mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})}\Big[\sum_{c=1}^{C}\Big(1+\log p_{\boldsymbol{\theta}}(y=c|\,\mathbf{x})\Big)\underbrace{\nabla_{\mathbf{x}} p_{\boldsymbol{\theta}}(y=c|\,\mathbf{x})}_{=\mathbf{0}\text{ Proposition 1)}}\Big] \Big|_{\mathbf{x}=\boldsymbol{\alpha}\,\odot\,\mathbf{x}'} \Bigg|\Bigg|_2$$

Because this lets the entire second term become the zero vector in the limit, the remaining part reduces to the case proven in Lemma 6:

$$= \underbrace{\lim_{\alpha_d \to \infty} \Bigg|\Bigg| \nabla_{\mathbf{x}} \mathbb{H}\Big[\mathbb{E}_{p(\boldsymbol{\theta}\,|\mathcal{D})}\Big[p_{\boldsymbol{\theta}}(y|\,\mathbf{x})\Big]\Big]\Big|_{\mathbf{x}=\boldsymbol{\alpha}\,\odot\,\mathbf{x}'} \Bigg|\Bigg|_2}_{\text{Lemma 6}} = 0$$

As the final result is an upper bound to the original expression and the l_2 norm provides a lower bound of 0, this proves the lemma. □

B SYNTHETIC DATA EXPERIMENTS

We perform our experiments on the half-moons dataset, using the corresponding function to generate the dataset in `scikit-learn` [Pedregosa et al., 2011], producing 500 samples for training and 250 samples for validation using a noise level of .125.

We do hyperparameter search using the ranges listed in Table 2, settling on the values given in Table 1 after 200 evaluation runs per model (for `NN` and `MCDropout`; the hyperparameters found for `NN` were then used for `PlattScalingNN`, `AnchoredNNEnsemble`, `NNEnsemble` as well). We also performed a similar hyperparameter search for the Bayes-by-backprop [Blundell et al., 2015] model, which seemed to not have yielded a suitable configuration even after extensive search, which is why results were omitted here. All models were trained with a batch size of 64 and for 20 epochs at most using early stopping with a patience of 5 epochs and the Adam optimizer.

All of the plots produced can be found in Figure 2 and 3, where uncertainty values where plotted for different ranges depending on the metric (variance: 0-0.25; (negative) entropy: 0-1; mutual information: $4-5$; (1 -) max. prob: $0-0.5$), with deep purple signifying high uncertainty and white signifying low uncertainty / high certainty.

Table 1: Best hyperparameters found on the half-moon dataset.

Model	Hyperparameter	Value
`NN`	`hidden_sizes`	$[25, 25, 25]$
`NN`	`dropout_rate`	0.014552
`NN`	`lr`	0.000538
`MCDropout`	`hidden_sizes`	$[25, 25, 25, 25]$
`MCDropout`	`dropout_rate`	0.205046
`MCDropout`	`lr`	0.000526

Table 2: Distributions or options that hyperparameters were sampled from during the random hyperparameter search.

Hyperparameter	Description	Chosen from
`hidden_sizes`	Hidden layers	1-5 layers of 15, 20, 25
`lr`	Learning rate	$\mathcal{U}(\log(10^{-4}), \log(0.1))$
`dropout_rate`	Dropout rate	$\mathcal{U}(0, 0.5)$

We can see in Figure 2 that maximum probability and predictive entropy behave quite similarly, forming a tube-like region of high uncertainty along what appear to be the decision boundary. In both cases, the region appears to be sharper in the case of maximum probability (right column) and also more defined after additional temperature scaling (bottom row). For all models and metrics, we see that the gradient magnitude decreases and approaches zero away from the training data (yellow / green plots), except for the cases discussed in Section 6.

In the next figure, Figure 3, we observe the uncertainty surfaces for models using multiple network instances. For the remaining models it is interesting to see that class variance (left column) didn't seem to produce significantly different values across the feature space except for the anchored ensemble. For predictive entropy (central column), we can see a similar behaviour compared to the single-instances models. Interestingly, the "fuzziness" of the high-uncertainty region increases with the ensemble and becomes increasing large with its anchored variant. Nevertheless, regions with static levels of certainty still exist in this case. For the mutual information plots (right column), epistemic uncertainty is lowest around the training data, where the model is best specified, which creates another tube-like region of high confidence even where there is no training data, an effect that is reduced with the neural ensemble and almost completely solved by the anchored ensemble. For all metrics, we see a magnitude close to zero for the uncertainty gradient away from the training data, except for the decision boundaries, as discussed in Section 6.

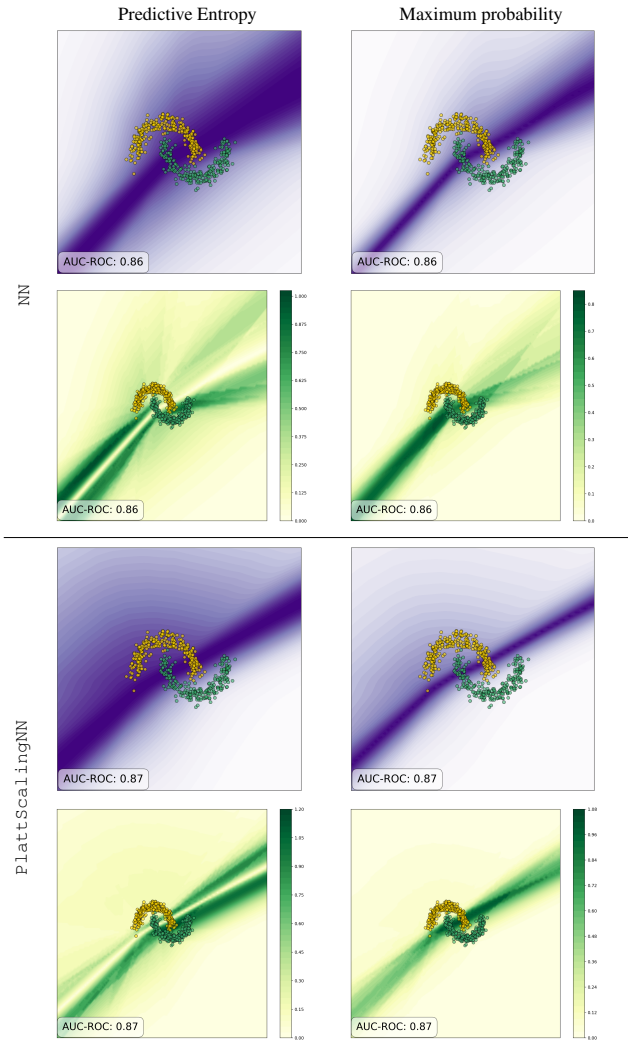

Figure 2: Uncertainty measured by different metrics for single-instance models (purple plots) and their gradient magnitude (yellow / green plots).

References

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

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

Bolin Gao and Lacra Pavel. On the properties of the softmax function with application in game theory and reinforcement learning. *arXiv preprint arXiv:1704.00805*, 2017.

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.

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.

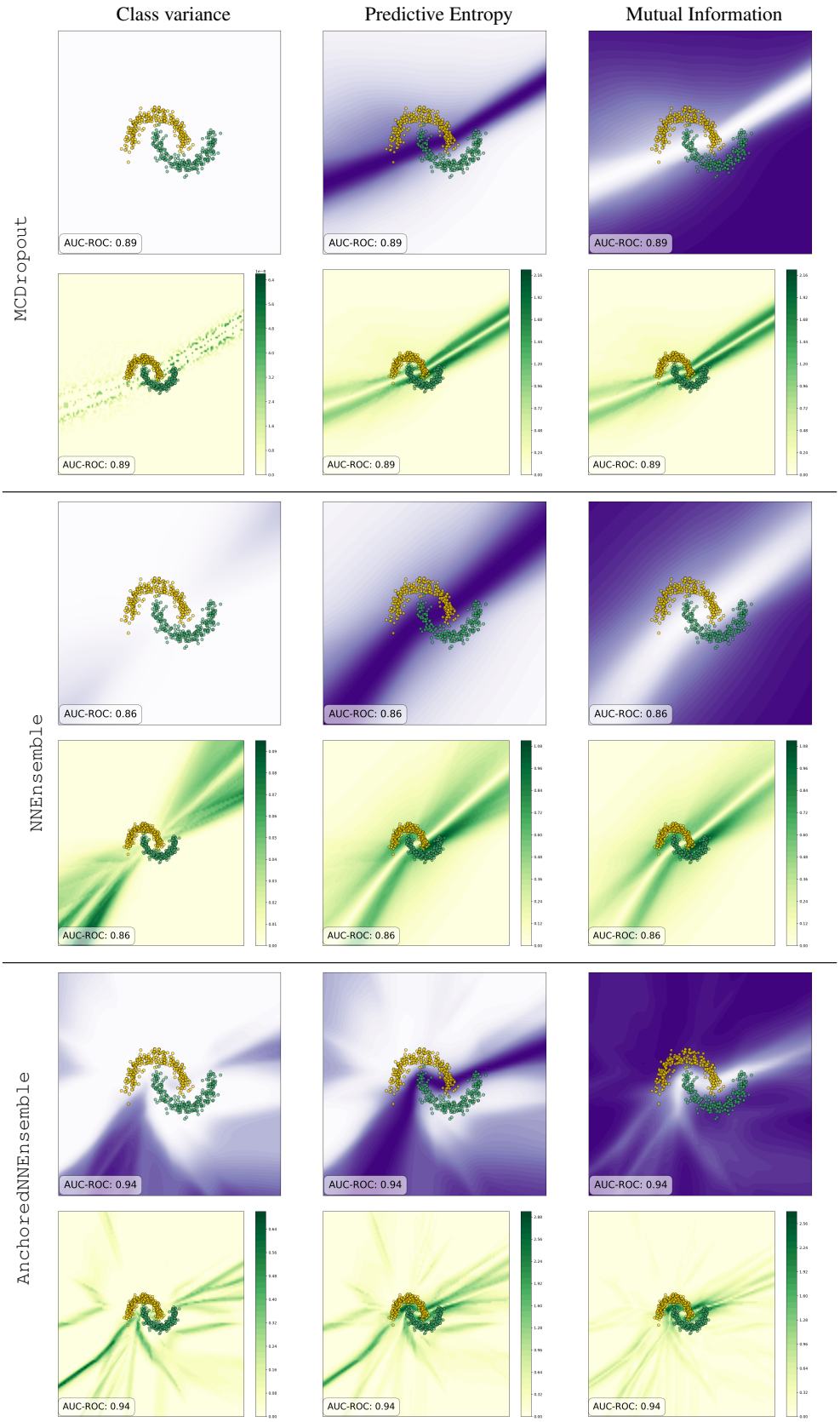

Figure 3: Uncertainty measured by different metrics for multi-instance models (purple plots) and the gradient of the uncertainty score w.r.t to the input (yellow / green plot).