Handling Black Swan Events in Deep Learning with Diversely Extrapolated Neural Networks

Maxime Wabartha[1,5], **Audrey Durand**[2,5], **Vincent Francois-Lavet**[3] and **Joelle Pineau**[1,4,5]
[1]McGill University
[2]Université Laval
[3]Université de Louvain
[4]Facebook AI Research
[5]Mila
maxime.wabartha@mail.mcgill.ca, audrey.durand@ift.ulaval.ca, vincent.francois@uclouvain.be, jpineau@cs.mcgill.ca

Abstract

By virtue of their expressive power, neural networks (NNs) are well suited to fitting large, complex datasets, yet they are also known to produce similar predictions for points outside the training distribution. As such, they are, like humans, under the influence of the Black Swan theory: models tend to be extremely "surprised" by rare events, leading to potentially disastrous consequences, while justifying these same events in hindsight. To avoid this pitfall, we introduce DENN, an ensemble approach building a set of Diversely Extrapolated Neural Networks that fits the training data and is able to generalize more diversely when extrapolating to novel data points. This leads DENN to output highly uncertain predictions for unexpected inputs. We achieve this by adding a diversity term in the loss function used to train the model, computed at specific inputs. We first illustrate the usefulness of the method on a low-dimensional regression problem. Then, we show how the loss can be adapted to tackle anomaly detection during classification, as well as safe imitation learning problems.

1 Introduction

"Black swans" are rare, surprising events that cannot be predicted by humans or statistical models. They can have huge repercussions, and we typically update our models to justify *a posteriori* the existence of such events [Taleb, 2007]. The financial crisis of 2008 is an often cited example of a black swan. While the statistical models were subsequently updated to take into account new data from the crisis, being overconfident, they would by definition still be surprised when a new black swan appears in the future (Fig. 1).

In machine learning models, being aware of the uncertainty in the predictions provided by an algorithm can help reduce the surprise coming from black swans and adopt a safe strategy. The chosen model should ideally be able to generalize, *i.e.* have low uncertainty predictions for inputs that are similar to the ones the model was trained on. At the same time, it should also be highly uncertain for completely novel inputs that do not display the same patterns as the training set.

NNs have grown to be the *de facto* model for large-scale prediction problems in the last decade, thanks to their ability to learn complex representations of high-dimensional data. Yet, while NNs are able to circumvent scaling issues, they do not come with a general and intrinsic estimation of uncertainty. In classification tasks, they tend to be overconfident about their predictions on unrelated datasets [Lakshminarayanan *et al.*, 2017]. Ensembles of NNs are a useful solution to this issue. They are a collection of several NNs, each initialized differently. For a given input, the uncertainty of the ensemble is then defined as the variance computed over each NN's predictions. It remains difficult to obtain meaningful posterior distributions over outputs that fit the training data (*in-distribution*, (ID)) while simultaneously having high variance in unseen regions of the input space (*out-of-distribution* (OOD)), where low-density inputs – akin to black swan events– are represented. In the Bayesian framework, this essentially corresponds to carrying over the high uncertainty from a prior to the posterior OOD. Specifically, ensembles can lack diversity OOD [Pearce *et al.*, 2018], resulting in a low predictive uncertainty for unexpected inputs.

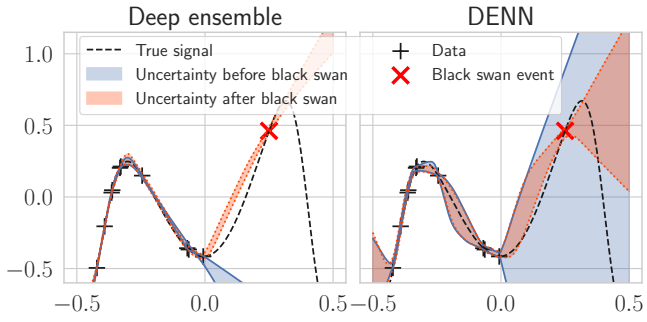

Figure 1: Illustration of the objective: the posterior predictive distribution of a method should be uncertain enough to cover the true signal, but a naive ensemble approach can fail to do so (left figure). When adding a black swan event to the training set, the ensemble rationalizes the new data point but still over-generalizes when extrapolating, on the contrary to our approach (right figure).

The main contributions of this work are:

- A novel way to enforce diversity in ensembles directly in the function space, through a diversity term applied specifically at *repulsive locations* (Sec. 3.),
- A theoretical analysis of this diversity term in the over-parameterized linear case, proving that it leads to an arbitrary diversity at repulsive locations while still fitting the observed data (Sec. 3.2),
- An empirical study of the proposed approach applied to NNs on regression, classification and anomaly detection for safe imitation learning tasks (Sec. 4).

2 Related Works

Ensemble methods are a major family of models used to estimate predictive uncertainty [Lakshminarayanan *et al.*, 2017; Pearce *et al.*, 2018; Lee and Chung, 2020; Tran *et al.*, 2020]. Training the same NN architecture with different initialization conditions, over the same training data, leads to different solutions. The predictions of the ensemble are aggregated to estimate its confidence, with higher uncertainty for unseen data [Lakshminarayanan *et al.*, 2017]. Additionally constraining their weights to stay close to their initial value increases the NNs diversity, forming an "anchored ensemble" [Pearce *et al.*, 2018]. This maintains the diversity induced by the initial weights, which otherwise tends to disappear during learning. Pearce et al. coin the term of *quasi-prior* to denote the predictor corresponding to an untrained NN. Both methods assume that the initial weights diversity is sufficient to obtain diverse predictors. However, it is neither clear how to increase or control this weights diversity, nor how it translates to the function space.

Monte-Carlo dropout [Gal and Ghahramani, 2016] maintains dropout at inference time (instead of removing it as usual). Stochastic outputs from a model trained with dropout are obtained by running forward passes through the network. Despite an easy implementation and theoretical backing [Gal, 2016], controlling the diversity of the method's predictions is not clear, as simply tuning the dropout rate is not sufficient.

The statistical bootstrap has long been studied and enjoys statistical guarantees [Efron, 1982]. It can be leveraged by training several NN on bootstrapped data [Osband *et al.*, 2016], potentially including a quasi-prior in the loss function [Osband *et al.*, 2018], to increase diversity where few samples have been gathered. Yet, the statistical bootstrap is of restricted use when the models that generated the data do not span the space of plausible models. It is also limited by the initial NN weights diversity.

Most of the previously discussed approaches rely on a natural diversity of NNs to generate different predictions. As we show in our experiments (Sec. 4), this strategy can fall short, especially when we need some control on the diversity of the functions. This is particularly true for deep, anchored or bootstrap prior ensembles that are limited by the natural expressivity of the quasi-prior. In our understanding, training diversely NNs thus remains a challenge.

We also relate our work to the principle of maximum entropy: the best representation of the current state of knowledge is the probability distribution that is consistent with known constraints, while having the largest entropy [Jaynes, 1957]. In other words, we want to avoid assuming more knowledge than we actually have by over-generalizing OOD. Recent works have focused on the OOD detection task in the classification setting [Lee *et al.*, 2018; Malinin and Gales, 2018]; [Hendrycks *et al.*, 2019] shows for instance that one can use external datasets to access OOD samples and perform OOD detection. The authors use them with a single NN and apply their method, Outlier Exposure, to the outlier detection task in the classification and density estimation setting, but not regression. The idea of training diverse predictors using repulsion has also been developed recently [Hong *et al.*, 2018] in reinforcement learning (RL), where recent policies are constrained to be different from old ones. In path planning in robotics, agents can consider obstacles as repulsive regions that they should avoid when trying to reach a goal, by defining a *potential field* giving a high value to the obstacles and a low value to the target [Vadakkepat *et al.*, 2000]. When performing segmentation of medical images, a repulsive force can be used to separate micro-structures close to each other [Cai *et al.*, 2006].

3 Trading-off Error and Diversity

To address the shortcomings of the existing approaches, we design a loss function that enforces sufficient diversity in an ensemble, explicitly OOD and controllable through hyperparameters, while leading to accurate and confident predictions ID. In this section, we define the proposed loss function and formally introduce the central notion of repulsive locations.

Let $\mathcal{X}$ and $\mathcal{Y}$ respectively denote the input and output space and let $\mathcal{D} = \{(x, y)\}_{x\in\mathcal{X}, y\in\mathcal{Y}}$ denote the set of training samples. The space of functions mapping $\mathcal{X}$ to $\mathcal{Y}$ is denoted $\mathcal{H}$. We consider the class of over-parameterized models (OM), *i.e.* models f_θ described by more parameters than the data they are trained on. Let $\{f_{\theta_i}\}_{i\in 1\dots K}$ denote an ensemble of K models. We say that an ensemble is *diverse* if the f_{θ_i} disagree OOD, a property that existing methods may fail to achieve (see Sec. 2 and 4). As diversity in the weight space does not necessarily translate into the function space $\mathcal{H}$, we act directly in the latter. More specifically, we explicitly train towards the behavior expected from the posterior distribution, *i.e.* to generate predictors that differ more in regions with lower density of training samples. Formally, we denote $\mathcal{L}_{\text{err},\mathcal{D}}(f_\theta)$ an error loss for predictor $f_\theta \in \mathcal{H}$ on $\mathcal{D}$, and $\mathcal{L}_{\text{div},\mathbb{X}}(f_{\theta_j}, \{f_{\theta_i}\}_{i\neq j})$ a diversity loss, or penalty, when f_{θ_j} is similar to f_{θ_i} for $j \neq i$ *w.r.t.* a similarity measure that we define later. We simultaneously minimize both $\mathcal{L}_{\text{err},\mathcal{D}}$ and $\mathcal{L}_{\text{div},\mathbb{X}}$ to enforce diversity.

3.1 Training an OM with a Diversity Constraint

To obtain functions trained on the same dataset $\mathcal{D}$ but that differ elsewhere, we define a *reference function* $g : \mathcal{X} \mapsto \mathcal{Y}$. A typical choice for g is an OM minimizing the loss $\mathcal{L}_{\text{err},\mathcal{D}}$. To diversify the ensemble, we constrain each f_θ to be diverse from g by minimizing $\mathcal{L}_{\text{div},\mathbb{X}}(f_\theta, g)$. This relaxes our previous definition of diverse samples: instead of enforcing diversity in the empirical posterior distribution by requiring each trained function to differ from all the others, we require new functions to differ only from the reference function (see Sec. 3.3

for details about the relaxation). Achieving this will require a notion of similarity between predictors. Let $k : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}$ denote a similarity measure between two values in the output space $\mathcal{Y}$. We only require k to be differentiable and bounded. A straightforward example of such a similarity is a kernel, but we do not restrict ourselves in general to this class of similarities. Let $\mathbb{X}$ denote a set of inputs sampled (see Sec. 3.4) from $\mathcal{X}$, with $|\mathbb{X}| = n_{\mathbb{X}}$.

We approximate the similarity between f_θ and a reference function g as:

$$\mathcal{L}_{\text{div},\mathbb{X}}(f_\theta, g) = \frac{1}{n_{\mathbb{X}}} \sum_{\mathrm{x} \in \mathbb{X}} k(f_\theta(\mathrm{x}), g(\mathrm{x})). \tag{1}$$

f_θ minimizing Eq. 1 against g will lead to different predictions at inputs $\mathbb{X}$, thus making f_θ diverse *w.r.t.* g.

3.2 Provable Diversity for a Linear OM

We now present the guarantees derived using the diversity loss $\mathcal{L}_{\text{div},\mathbb{X}}(f_\theta, g)$. We restrict our study in this section to the linear setting. We represent the over-parameterized representation's feature maps (assumed linearly independent) for a given point x by $\Phi_x \in \mathbb{R}^{1\times k}$, such that $f_\theta(x) = \Phi_x\theta$, with $\theta \in \mathbb{R}^k$. We first notice that there exist solutions to $\mathcal{L}_{\text{err},\mathcal{D}} = 0$. We assume that $\mathcal{L}_{\text{err},\mathcal{D}}$ is the Mean Squared Error (MSE) (A1), that f_θ is linear OM (A2) and that the points in $\mathcal{D}$ are linearly independent (A3).

Lemma 1. *Assuming (A1-3), we can find infinitely many ω^* such that $\mathcal{L}_{err,\mathcal{D}}(f_{\omega^*}) = 0$.*

Proof. We consider the linear regression problem $\mathcal{L}_{\text{err},\mathcal{D}}(f_\omega) = 0$, *i.e.* $\Phi_\mathcal{D}\omega = Y$ (A1). We associate to $\mathcal{D}$ the design matrix $\Phi_\mathcal{D} \in \mathbb{R}^{n\times k}$, with $k > n$ (A2), assumed of rank n (A3), and the targets $Y \in \mathbb{R}^{n\times 1}$. Then, according to Ordinary Least Square regression, the pseudo inverse $\Phi_\mathcal{D}^\dagger = \Phi_\mathcal{D}^\top(\Phi_\mathcal{D}\Phi_\mathcal{D}^\top)^{-1}$ gives a particular solution $\omega_0 = \Phi_\mathcal{D}^\dagger Y$ of the linear regression problem: $\Phi_\mathcal{D}\omega_0 = \Phi_\mathcal{D}\Phi_\mathcal{D}^\top(\Phi_\mathcal{D}\Phi_\mathcal{D}^\top)^{-1}Y = Y$. Then, the nullspace of $\Phi_\mathcal{D}$, $\mathcal{N}(\Phi_\mathcal{D})$, is of dimension at least 1 (A2-3). Take any v in $\mathcal{N}(\Phi_\mathcal{D})$ (using *e.g.* the SVD of $\Phi_\mathcal{D}$). Then, $\forall\mu \in \mathbb{R}$, $\omega^* = \omega_0 + \mu v$ verifies $\mathcal{L}_{\text{err},\mathcal{D}}(f_{\omega^*}) = 0$: $\Phi_\mathcal{D}(\omega_0 + \mu v) = \Phi_\mathcal{D}\omega_0 + \mu\Phi_\mathcal{D}v = \Phi_\mathcal{D}\omega_0 = Y$. □

We now focus on the problem of minimizing both $\mathcal{L}_{\text{err},\mathcal{D}}$ and $\mathcal{L}_{\text{div},\mathbb{X}}$, to prove the diversity of f *w.r.t.* g. Since the space of solutions of $\mathcal{L}_{\text{err},\mathcal{D}}(f_\theta) = 0$ is of dimension at least 1 (Lemma 1), we express the minimization problem as:

$$\begin{aligned} \min_\theta \quad & \mathcal{L}_{\text{div},\mathbb{X}}(f_\theta), \\ \text{s.t.} \quad & \mathcal{L}_{\text{err},\mathcal{D}}(f_\theta) = 0 \end{aligned} \tag{2}$$

We assume that $|\mathbb{X}| = 1$ (A4), and k is a RBF kernel (A5).

Proposition 1. *Given (A1-5), the infimum of (2) is 0: $\forall\epsilon > 0$, we can find θ^* s.t. $\mathcal{L}_{div,\mathbb{X}}(f_{\theta^*}) < \epsilon$ and $\mathcal{L}_{err,\mathcal{D}}(f_{\theta^*}) = 0$.*

Proof. Let $\epsilon > 0$. Following the above notation, let $\Phi_\mathbb{X}$ the design matrix associated with $\mathbb{X}$. Using (A4-5) and applying $-\log$ (monotonically decreasing) to the objective, we maximize the equivalent problem:

$$\text{find } \theta \text{ s.t. } ||\Phi_\mathbb{X}\theta - g(\mathrm{x})||_2^2/(2\sigma^2) > \log(1/\epsilon), \Phi_\mathcal{D}\theta = Y \tag{3}$$

$\omega_0 + \mu v$ is a solution to the constraint from Eq. 3 (Lemma 1). We substitute it in the objective of (3) to yield (4):

$$\text{find } \mu \text{ s.t. } ||\Phi_\mathbb{X}\omega_0 + \mu\Phi_\mathbb{X}v - g(\mathrm{x})||_2^2 > 2\sigma^2\log(1/\epsilon) \tag{4}$$

which is a problem of maximizing a convex function of μ, for which we know there exists solutions, and that can be solved using a simple gradient ascent algorithm. □

Note that once the tolerance ϵ is set, increasing σ increases the diversity of f_{θ^*} *w.r.t.* g.

3.3 Training NNs with a Diversity Constraint

Prop. 1 essentially conveys that we can build a linear OM f_{θ^*} as diverse as desired from g at repulsive location x, while still fitting $\mathcal{D}$. This property is especially interesting as we shift our focus to the training of NNs. Due to their high number of parameters and the intrinsic low-dimensional nature of most natural datasets, we expect NNs to behave like over-parameterized models [Han *et al.*, 2015; Frankle and Carbin, 2019; Li *et al.*, 2018]. In other terms, we propose to leverage the extra capacity of NNs to boost their diversity, using the diversity term that we introduced.

To this end, we relax Eq. 2 by trading off the error term from the constraint $\mathcal{L}_{\text{err},\mathcal{D}}(f_\theta)$ and the diversity term *w.r.t.* g $\mathcal{L}_{\text{div},\mathbb{X}}(f_\theta, g)$ into our proposed training loss (Eq. 5).

$$\begin{aligned} \mathcal{L}(f_\theta; g, \mathcal{D}, \mathbb{X}) = \frac{1}{n}\sum_{(x,y)\in\mathcal{D}} d(f_\theta(x), y)) \Big\}_{\text{error loss } \mathcal{L}_{\text{err},\mathcal{D}}} \\ + \frac{\lambda}{n_\mathbb{X}}\sum_{\mathrm{x}\in\mathbb{X}} k(f_\theta(\mathrm{x}), g(\mathrm{x})) \Big\}_{\text{diversity loss } \mathcal{L}_{\text{div},\mathbb{X}}}, \end{aligned} \tag{5}$$

$\lambda \geq 0$ denotes a trade-off hyperparameter. Crucially, $\mathcal{L}_{\text{err},\mathcal{D}}$ and $\mathcal{L}_{\text{div},\mathbb{X}}$ serve opposite objectives if $\mathbb{X} \cap \mathcal{D} \neq \emptyset$, which leads to regularization [Hafner *et al.*, 2019]. Outside of $\mathcal{D}$, the diversity loss is the only one active: it induces an OOD *repulsion* between f_θ and g. Observe that for $\lambda = 0$ and $d(f_\theta(x), y) = (f_\theta(x) - y)^2$, Eq. 5 recovers the MSE loss.

Each NN of an ensemble is trained using Eq. 5. Given a stochastic initialization of θ_i and the aleas of optimization of gradient-based methods, we expect f_{θ_i} to differ from g after training in diverse ways for different i. If needed, we can use a different σ_i for each f_{θ_i}, leading to diverse levels of disagreement with g; we validate empirically this claim in Sec. 4. One could also impose a constraint on the diversity of $\{f_{\theta_i}\}_{i\in 1\ldots N}$ directly. This could ensure variety for each f_{θ_i}, but would require training them jointly.

3.4 Sampling Repulsive Locations

In this section, we explore the main ideas driving the selection of the set of repulsive locations $\mathbb{X}$, which represent the inputs where we expect the models to disagree. Ideally, $\mathbb{X}$ should be chosen in the manifold where all possible observations lie ($\mathcal{X}^\star$ in Fig. 2) and OOD, to ensure sufficient uncertainty at

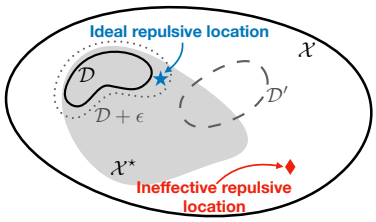

Figure 2: The different regions of the state space $\mathcal{X}$: $\mathcal{X}^\star$ represents the low-dimensional manifold where structured data are located. $\mathcal{D}$ is the training distribution. We sample the repulsive locations from a perturbed training distribution ($\mathcal{D}+\epsilon$) or a similar dataset ($\mathcal{D}'$).

the repulsive locations. However, $\mathcal{X}^\star$ is not directly accessible, and sampling randomly in $\mathcal{X}$ can result in repulsive locations arbitrarily far from $\mathcal{X}^\star$ (red diamond, Fig. 2), which can hurt uncertainty propagation and performance. We consider instead the procedure described in [Hafner *et al.*, 2019] by choosing repulsive locations at the boundary of the training distribution $\mathcal{D}$, where different models would usually agree. The enforced uncertainty at the boundary is then propagated OOD [Hafner *et al.*, 2019], including to $\mathcal{X}^\star$.

This is achievable by adding noise to the training data. Some of the noisy samples will lie, by definition, at the boundary of $\mathcal{D}$ and will be OOD, achieving the targeted goal (blue star, Fig. 2). The repulsive locations sampled inside $\mathcal{D}$ add a repulsive effect countered by the error loss attractive effect, which results in label smoothing [Hafner *et al.*, 2019]. We can also factor the structure of high-dimensional inputs such as images in the noise. With $\mathcal{X}$ being, *e.g.* , video game frames, we can switch different pixels or change their color (Sec. 4.3). We also consider choosing $\mathbb{X}$ from a dataset $\mathcal{D}'$ with similar features to $\mathcal{D}$ [Hendrycks *et al.*, 2019], with the intuition that $\mathcal{D}$ and $\mathcal{D}'$ should be represented close to each other (Sec. 4.2).

3.5 Algorithm

Algorithm 1 (DENN) describes the training of an ensemble of diverse NNs f_{θ_j}. Given a reference function, we sample a batch of training data and a batch of repulsive locations. We then compute and combine $\mathcal{L}_{\text{err},\mathcal{D}}$ and $\mathcal{L}_{\text{div},\mathbb{X}}$, and perform standard backpropagation. Two cases may arise; (1) g can be trained beforehand using $\mathcal{L}_{\text{err},\mathcal{D}}$: it is then possible to train simultaneously a *diverse ensemble* $\{f_{\theta_i}\}_{i\in 1\dots K}$. (2) g can be a previous diverse function $f_{\theta_{j-1}}$: then, f_{θ_j} can be trained *sequentially*. The repulsive hyperparameters λ and σ depend on the repulsive dataset we choose. They are tuned with cross-validation on a separate dataset. Repulsive locations and the associated loss (Eq. 5) should be especially useful to detect outliers, or for tasks where uncertainty should affect decisions, either as a potential source of risk (in the case of imitation learning) or gains (for exploration in RL).

In terms of computational overhead, adding the diversity loss term to the usual error loss requires sampling $\mathbb{X}$, computing $k(f(x), g(x))$ for all $x \in \mathbb{X}$, and backpropagating the gradients for the repulsive locations in $\mathbb{X}$. Overall, the added training time of f corresponds roughly to the time required to perform an additional backpropagation step due to the presence of the repulsive locations.

Algorithm 1 DENN

Require: Dataset $\mathcal{D}$, parameters λ and μ, reference function g trained on $\mathcal{D}$, and repulsive locations $\mathbb{X}$.
  for K NNs, possibly trained in parallel **do**
    for each training step **do**
      Sample training batch of size b: $\{(x_i, y_i)\}_{i\in 1\dots b}$
      Sample repulsive batch of size $n_{\mathbb{X}}$: $\{x_j\}_{j\in 1\dots n_{\mathbb{X}}}$
      Compute loss (Eq. 5) and backprop. gradients.
  Output: Ensemble $\{f_{\theta_i}\}_{i\in 1..K}$ diverse OOD *w.r.t.* g.

4 Experiments

In this section, we apply the proposed loss function to different tasks and compare the results[1] with existing approaches, to assess if using DENN can lead to the desired high uncertainty OOD. We first compare the performance of DENN with other approaches on a simple regression task to study visually the advantages brought by the diversity constraint to the posterior predictive distribution. We then illustrate how DENN can seamlessly be applied to classification, enabling the training of an ensemble having diverse predictions for unexpected datasets. Finally, we study high-dimensional, structured datasets, and show that DENN can be used to detect black swan events.

4.1 First Case: Low-dimensional Regression

We generate a training set $\mathcal{D}$ of 10 points in $[-0.5, 0]$ mapped to a bimodal function (the ground truth), such that the second mode lies outside of $\mathcal{D}$. We expect the empirical posterior predictive distribution to cover the ground truth.

We use the classical MSE loss, *i.e.* $\mathcal{L}_{\text{err},\mathcal{D}}$ with $d(f(x), y) = ||f(x) - y||_2^2$, and the Radial Basis Function (RBF) $k(f(\mathrm{x}), g(\mathrm{x})) = \exp\left(-||f(\mathrm{x}) - g(\mathrm{x})||_2^2/(2\sigma^2)\right)$ in $\mathcal{L}_{\text{div},\mathbb{X}}$, where σ controls the diversity between two predictors at location x. In the special case of a one-dimensional regression, training $f_1, f_2, \dots, f_N$ with the same σ can lead to functions highly different from g but alike one to another. To avoid this pitfall, we sample σ from $[10^{-3}; 10^{-1/2}]$ log-uniformly. We generate the repulsive locations by adding Gaussian noise (variance 0.3) to the training points (Sec. 3.4).

Experiment

We compare DENN (Alg. 1) with popular ensemble methods: anchoring [Pearce *et al.*, 2018], input bootstrap [Efron, 1982], bootstrap prior [Osband *et al.*, 2018] and MC dropout [Gal and Ghahramani, 2016]. Here, all NNs are 2-layer multilayer perceptrons (MLP) with 64 hidden units per layer and optimized with the default Adam. The reference g is trained beforehand with a MSE loss on $\mathcal{D}$ until convergence.

Fig. 3 shows the posterior predictive distribution empirically estimated by taking, for each approach, the pointwise average and 1, 2 and 3 standard deviations over ensembles of 50 NNs. While DENN successfully covers the second mode of the ground truth (Fig. 3, bottom-right), we do not manage to obtain diverse NNs with the anchoring, bootstrap prior and

[1]Code: https://github.com/maxwab/denn-ijcai

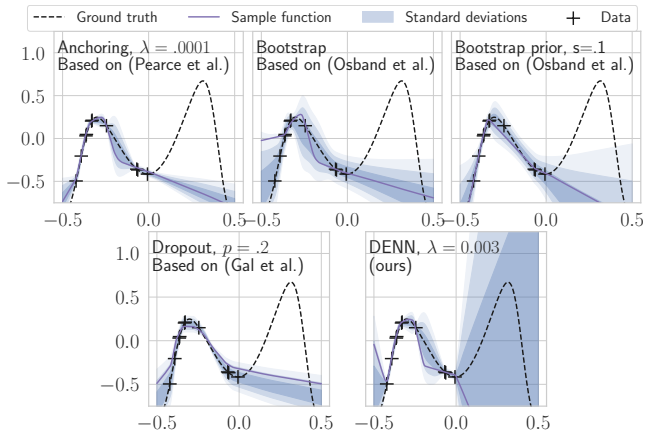

Figure 3: Empirical posterior predictive distributions (over 50 sampled functions) for 10 data points generated from a ground truth.

dropout methods for different values of their respective hyperparameters, as evidenced by their narrow posterior predictive distributions for $x > 0$. Note that MC dropout's uncertainty rises around 0 despite the presence of several training points. Input bootstrap produces NNs that span slightly better the width of outputs, but also disregards by nature certain points in $\mathcal{D}$, where we expect low uncertainty given our current knowledge (near the first peak of the ground truth, Fig. 3, top-right). Thus, all studied ensembles lack diversity, causing an inability to be highly uncertain for OOD data. While NNs should be able to anticipate any point on the ground truth [Cybenko, 1989], current methods are over-confident for unexpected inputs and fail to do so.

4.2 Second Case: Classification

We now evaluate the uncertainty provided by DENN on a classification task. In this context, a principled notion of uncertainty exists in the form of the entropy of the prediction probability vector. We expect a good approach to have confident predictions ID, which shows generalization, while having high uncertainty OOD, effectively detecting outliers.

Eq. 5 extends naturally to classification tasks with d the usual cross-entropy loss and $k(f(\mathrm{x}), g(\mathrm{x})) = \exp(-|H(g(\mathrm{x}), f(\mathrm{x})) - H(f(\mathrm{x}), f(\mathrm{x}))|^2)/(2\sigma^2))$ in $\mathcal{L}_{\text{div},\mathbb{X}}$, where $H(g(\mathrm{x}), f(\mathrm{x}))$ denotes the cross-entropy between probability vectors $f(\mathrm{x})$ and $g(\mathrm{x})$.Thus, $k(f(\mathrm{x}), g(\mathrm{x})) = 1$ if $f(\mathrm{x}) = g(\mathrm{x})$, while $k(f(\mathrm{x}), g(\mathrm{x})) \approx 0$ if $f(\mathrm{x})$ and $g(\mathrm{x})$ significantly differ. We randomly sample the repulsive locations x from other structurally similar datasets (Sec. 3.4): here, the FashionMNIST[2], notMNIST and EMNIST datasets[3] [Cohen *et al.*, 2017].

Experiment

We replicate the benchmark proposed in [Lakshminarayanan *et al.*, 2017] using the same 2 hidden layers MLP setting for all NNs. We train DENN, a deep ensemble[4], Bootstrap prior

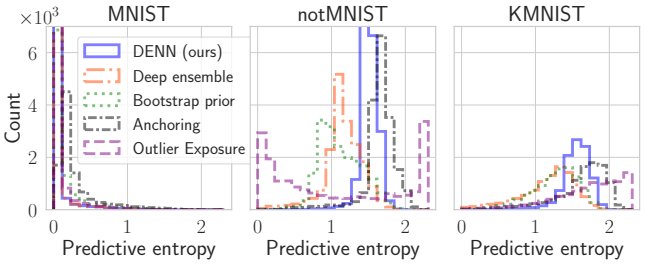

Figure 4: Histograms of predictive entropy on 3 different datasets. Training is performed on MNIST. More mass on the right means more uncertainty on the inputs from the corresponding dataset.

that we adapt to classification and Anchoring on the MNIST dataset[5]. We also compare ourselves to Outlier Exposure (OE) [Hendrycks *et al.*, 2019]: we use the authors' code to train a single MLP constrained to output uniform predictions OOD (also accessed using an external dataset). We evaluate the methods' accuracy and predictive uncertainty on the MNIST test set to ensure generalization, and their uncertainty on OOD datasets: letters with notMNIST[6] and Japanese characters with KMNIST[7]. DENN and OE are trained with repulsive locations drawn from FashionMNIST. The reference g is a MLP trained only with the cross-entropy loss. All hyperparameters are chosen by cross-validation by computing the histogram of predictive entropies over the MNIST validation set (generalization) and EMNIST (OOD uncertainty). We then perform a Z-test between both histograms and choose the best hyperparameters. The final probability is averaged over each NN predictions.

The middle and right plots of Fig. 4 show that DENN produces highly uncertain predictions on the OOD datasets, as desired. It performs favorably compared to the deep and bootstrap prior ensembles. The average entropy on notMNIST of each NN of DENN taken alone is 0.04 (each one is extremely confident) while the entropy of the ensembled OOD predictions is 1.56, implying that the NNs have on average diverse predictions OOD. Anchoring has marginally higher entropy predictions than DENN for OOD inputs, but also displays a higher uncertainty ID (Fig. 4, left plot). We relate this to the fact that the diversity mechanism from anchoring does not include a notion of OOD, whereas DENN's diversity loss is specifically tailored for OOD inputs. Therefore, DENN predictions on the MNIST test set have very low entropy, barely above the deep ensemble (Fig. 4, left plot) and are accurate despite the repulsive constraint (98.7% accuracy, compared to the deep ensemble's 98.6% and anchoring's 98.2%). OE performs the best on KMNIST (despite having some confident predictions for certain inputs, on the contrary to DENN) but is extremely confident for half the notMNIST dataset.

This experiment shows DENN's ability to have high predictive entropy OOD while maintaining generalization ID.

[2]http://github.com/zalandoresearch/fashion-mnist

[3]http://www.nist.gov/itl/products-and-services/emnist-dataset

[4]without optional adversarial training as it brings marginal gains for predicting uncertainty [Lakshminarayanan *et al.*, 2017]

[5]http://yann.lecun.com/exdb/mnist/

[6]http://yaroslavvb.blogspot.com/2011/09/notmnist-dataset.html

[7]http://github.com/rois-codh/kmnist

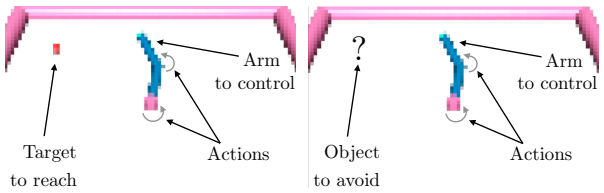

Figure 5: The modified Reacher task. The agent should reach the target, a red sphere (left image), while also adopting a safe strategy if the target is unknown (right image).

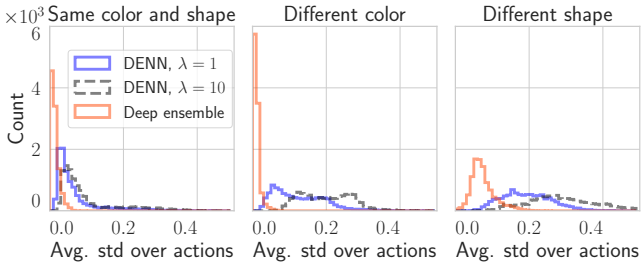

Figure 6: Histograms of the predictive standard deviations averaged over action predictions. Higher values indicate higher uncertainty over the action to take. DENN is clearly more uncertain than the deep ensemble on OOD datasets (middle and right), as desired, while maintaining confident predictions on ID data (left).

4.3 Third Case: Safe Imitation Learning

Finally, we illustrate an application of DENN in an imitation learning setting with high-dimensional inputs. In imitation learning, it is well-known that the approximation error of a model grows with the length of trajectories [Ross *et al.*, 2011]. It is thus interesting to know at which states to get new demonstrations [Kim and Pineau, 2013]. We show how the increased diversity of DENN, this time composed of convolutional NNs, can help detect such states.

In imitation learning, the agent aims to solve a control task while having access to optimal actions, called demonstrations, from an oracle. Imitation learning can be framed as successive supervised learning problems [Ross *et al.*, 2011] by regressing, for an input s, the action to take $\pi(s)$. However, demonstrations are expensive (they can require, *e.g.* , a human intervention). One can use the agent's confidence over its actions to only request demonstrations for novel inputs, using for instance the kernel-based Maximum Mean Discrepancy [Borgwardt *et al.*, 2006; Kim and Pineau, 2013]. To scale easily to high-dimensional problems, we use DENN to learn π and compute the uncertainty over the policy predictions. If this value exceeds a certain threshold, we consider the current state an outlier and ask for a new demonstration. We focus on a pixel version of MuJoCo Reacher [Todorov *et al.*, 2012], with RGB frames (Fig. 5) as inputs to the models ($x \in [0; 1]^{84\times84\times3}$). The agent uses a policy π to control an articulated arm to reach a target, by default a red sphere.

We generate OOD datasets by changing the color of the target and training a PPO [Schulman *et al.*, 2017] agent to reach them. We then generate trajectories that we store. Note that we place ourselves in the worst-case scenario so that the color we are evaluating the predictive uncertainty on (blue) is not a linear combination of the training color (red) and the target color of the repulsive frames (green) in the RGB space. We train DENN with the MSE for d and a RBF for k. We first evaluate the ability of the models to ask for demonstrations when confronted with a target of a different color. We then observe that the trained model behaves favorably even when evaluated on frames with targets of different shape.

Results

We first generate the demonstrations with a red sphere target as $\mathcal{D}$, as well as the repulsive frames, stored in $\mathbb{X}$, with a green sphere target. Then, we train in a supervised learning fashion a policy $\pi : \mathcal{X} \mapsto \mathbb{R}^2$ using a deep ensemble and our DENN approach (each composed of 10 convolutional NNs) using the green repulsive dataset. We find the value of (λ, σ) with cross-validation on a yellow sphere targets dataset. We then evaluate both methods on red sphere targets for generalization and blue sphere targets for outlier detection.

While the deep ensemble is more confident than DENN on the generalization dataset (Fig. 6, left plot), it is also more confident on the OOD dataset (middle plot) and thus cannot detect outliers as well. Critically, the deep ensemble is more confident OOD than ID, conversely to DENN which displays the desired behaviour. Interpreting the target color change as a black swan event (*e.g.* with an extremely negative reward) illustrates the failure of regular ensembles to anticipate unlikely events.

Additionally, we evaluate how confident the previously trained DENN and the deep ensemble are when the target shape, now a rectangle, instead of the color has changed. We observe that DENN again outperforms the deep ensemble. Notice that changing λ from 1 to 10 results in a higher uncertainty on the unseen blue sphere and red rectangle datasets, as desired, but also on the red sphere dataset. This is consistent with the definition of λ as a trade-off parameter: the diversity of DENN increases with λ at the expense of generalization.

We believe that setting a repulsive constraint on one attribute – the color – of the target led to modify the whole representation of the target, including its shape. This opens up several ways of choosing repulsive locations: having a frame where a single characteristic is different may be sufficient to increase uncertainty for OOD frames, to be applied in a context of safe decision making.

5 Conclusion

In this work, we described a method for training convolutional and regular NNs more diverse OOD by using a modified loss function enacting directly in the function space. We explored various methods to sample the repulsive locations used in the proposed loss function, and discussed how choosing judiciously the repulsive locations can modify the learnt representation to be more uncertain when confronted with "surprising" data points, thus offering a solution to handle black swan events in deep learning. Moreover, our approach does not over-generalize after seeing such an event, conversely to deep ensembles. We compared our technique with existing methods providing NNs with uncertainty, and illustrated the importance of training diverse predictors on both

low and high-dimensional regression and classification problems. Applied to imitation learning, we studied how DENN can detect outliers more efficiently than a usual ensemble, requiring so fewer demonstrations.

This work could be extended in a few ways. The method introduces hyperparameters that necessitate tuning, and three distinct datasets: for model training, for producing the repulsive locations, and for hyperparameter selection. This can be restrictive when the problem offers limited sources of data. Finally, working in the latent space using a variational autoencoder [Kingma and Welling, 2014] could help sampling repulsive locations independent of the input space nature, as recent works have focused on the repulsive datasets themselves [Abbasi *et al.*, 2019; Sensoy *et al.*, 2020].

We believe that having the ability to train diverse functions to have uncertain predictions OOD is promising; future research includes extending DENN to RL and developing more principled ways of choosing repulsive locations.

Acknowledgements

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Canada CIFAR AI chairs program. The authors would like to thank specifically Guillaume Rabusseau for his help with the proofs, Jonathan Lebensold for coming up with the "Black Swan" analogy, Patrick Nadeem Ward, Clara Lacroce, Thang Doan, Alex Lamb, and more generally the several people that provided fruitful discussions and insights on earlier versions of this paper.

References

[Abbasi *et al.*, 2019] Mahdieh Abbasi, Changjian Shui, Arezoo Rajabi, Christian Gagne, and Rakesh Bobba. Toward metrics for differentiating out-of-distribution sets. In *Advances in Neural Information Processing Systems Workshop on Safety and Robustness in Decision Making*, 2019.

[Borgwardt *et al.*, 2006] Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57, 2006.

[Cai *et al.*, 2006] Hongmin Cai, Xiaoyin Xu, Ju Lu, Jeff W Lichtman, SP Yung, and Stephen TC Wong. Repulsive force based snake model to segment and track neuronal axons in 3d microscopy image stacks. *NeuroImage*, 32(4):1608–1620, 2006.

[Cohen *et al.*, 2017] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. EMNIST: an extension of MNIST to handwritten letters. *CoRR*, abs/1702.05373, 2017.

[Cybenko, 1989] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[Efron, 1982] Bradley Efron. *The jackknife, the bootstrap, and other resampling plans*. Siam, 1982.

[Frankle and Carbin, 2019] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, Conference Track Proceedings*, 2019.

[Gal and Ghahramani, 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.

[Gal, 2016] Yarin Gal. *Uncertainty in deep learning*. PhD thesis, University of Cambridge, 2016.

[Hafner *et al.*, 2019] Danijar Hafner, Dustin Tran, Timothy P. Lillicrap, Alex Irpan, and James Davidson. Noise contrastive priors for functional uncertainty. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, 2019.

[Han *et al.*, 2015] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in neural information processing systems*, pages 1135–1143, 2015.

[Hendrycks *et al.*, 2019] Dan Hendrycks, Mantas Mazeika, and Thomas G Dietterich. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, Conference Track Proceedings*, 2019.

[Hong *et al.*, 2018] Zhang-Wei Hong, Tzu-Yun Shann, Shih-Yang Su, Yi-Hsiang Chang, Tsu-Jui Fu, and Chun-Yi Lee. Diversity-driven exploration strategy for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 10510–10521, 2018.

[Jaynes, 1957] Edwin T Jaynes. Information theory and statistical mechanics. *Physical review*, 1957.

[Kim and Pineau, 2013] Beomjoon Kim and Joelle Pineau. Maximum mean discrepancy imitation learning. In *Robotics: Science and systems*, 2013.

[Kingma and Welling, 2014] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[Lakshminarayanan *et al.*, 2017] 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.

[Lee and Chung, 2020] Jisoo Lee and Sae-Young Chung. Robust training with ensemble consensus. In *International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia , April 26- May 1, 2020, Conference Track Proceedings*, 2020.

[Lee *et al.*, 2018] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *International*

Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30- May 3, 2018, Conference Track Proceedings, 2018.

[Li *et al.*, 2018] Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30- May 3, 2018, Conference Track Proceedings*, 2018.

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

[Osband *et al.*, 2016] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. In *Advances in Neural Information Processing Systems*, pages 4026–4034, 2016.

[Osband *et al.*, 2018] Ian Osband, John Aslanides, and Albin Cassirer. Randomized prior functions for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 8626–8638, 2018.

[Pearce *et al.*, 2018] Tim Pearce, Nicolas Anastassacos, Mohamed Zaki, and Andy Neely. Bayesian inference with anchored ensembles of neural networks, and application to reinforcement learning. In *International Conference on Machine Learning Workshop on Exploration in Reinforcement Learning*, 2018.

[Ross *et al.*, 2011] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics*, pages 627–635, 2011.

[Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv:1707.06347*, 2017.

[Sensoy *et al.*, 2020] Murat Sensoy, Lance Kaplan, Federico Cerutti, and Maryam Saleki. Uncertainty-aware deep classifiers using generative models. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

[Taleb, 2007] Nassim Nicholas Taleb. *The black swan: The impact of the highly improbable*, volume 2. Random house, 2007.

[Todorov *et al.*, 2012] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

[Tran *et al.*, 2020] Linh Tran, Bastiaan S. Veeling, Kevin Roth, Jakub Swiatkowski, Joshua V. Dillon, Jasper Snoek, Stephan Mandt, Tim Salimans, Sebastian Nowozin, and Rodolphe Jenatton. Hydra: Preserving ensemble diversity for model distillation. *arXiv:2001.04694*, 2020.

[Vadakkepat *et al.*, 2000] Prahlad Vadakkepat, Kay Chen Tan, and Wang Ming-Liang. Evolutionary artificial potential fields and their application in real time robot path planning. In *Congress on evolutionary computation. CEC00 (Cat. No. 00TH8512)*, volume 1, pages 256–263. IEEE, 2000.