Out-of-distribution detection for computational pathology with multi-head ensembles

by
JIM WINKENS
10003592

February 8, 2019

36 ECTs
April 2018 - February 2019

Conducted during a research internship at the Diagnostic Image Analysis Group of the Radboudumc in Nijmegen, The Netherlands.
Abstract

Distribution shift is a common phenomenon in real-life safety-critical situations that is detrimental to the performance of current deep learning models. Constructing a principled method to detect such a shift is critical to building safe and predictable automated image analysis pipelines for medical imaging. In this work, we interpret the problem of out-of-distribution detection for computational pathology in an epistemic uncertainty estimation setting. Given the difficulty of obtaining a sufficiently multi-modal predictive distribution for uncertainty estimation, we present a multiple heads topology in CNNs as a highly diverse ensembling method. We empirically prove that the method exhibits greater representational diversity than various popular ensembling methods, such as MC dropout and Deep Ensembles. The fast gradient sign method is repurposed and we show that it separates the softmax scores of in-distribution samples and out-of-distribution samples. We identify the challenges for this task in the domain of computational pathology and extensively demonstrate the effectiveness of the proposed method on two clinically relevant tasks in this field.
Contents

1 Introduction ................................................. 4
  1.1 Contributions of this thesis ............................. 5

2 Preliminaries ................................................. 7
  2.1 Computational pathology .................................. 7
    2.1.1 Lymph nodes metastases .......................... 7
    2.1.2 Prostate biopsy .................................. 7
  2.2 Uncertainty in machine learning ......................... 8
  2.3 Uncertainty via ensembles ................................ 8
    2.3.1 MC dropout ..................................... 9
    2.3.2 Deep Ensembles .................................. 9
    2.3.3 Reconstruction-based and density-based methods 10

3 Multiple hypotheses model ................................. 11
  3.1 Network architecture and training procedure ........... 11
    3.1.1 Depth of heads .................................. 12

4 Representational diversity in ensembles .................. 14
  4.1 Canonical correlation analysis on representations in neural networks 14
    4.1.1 Mathematical interpretation of CCA ................ 15
    4.1.2 CCA distance ................................... 15
  4.2 Representational diversity in ensembles on CIFAR-10 16
    4.2.1 Diminishing diversity of M-Heads ................ 17

5 Input pre-processing using FGSM .......................... 18
  5.1 Adversarial examples .................................. 18
    5.1.1 Model linearity .................................. 18
    5.1.2 A simple attack: Fast Gradient Sign Method ....... 18
  5.2 FGSM for detection of OOD samples ...................... 19
    5.2.1 FGSM for M-Heads ................................ 20
  5.3 Validation of $\epsilon$ ................................ 20

6 Distance-based OOD detection .............................. 22
  6.1 Mathematical formulation of distance-based detection 23
    6.1.1 Linear discriminant analysis ..................... 23
    6.1.2 Mahalanobis distance-based classification ....... 23
  6.2 LDA for M-Heads ...................................... 24
# Experiments

## 7.1 Datasets

## 7.2 Experimental setup

- 7.2.1 Detection of metastases in sentinel lymph node
- 7.2.2 Detection of epithelium in prostate
- 7.2.3 Slide-level out-of-distribution measure

## 7.3 Out-of-distribution detection performance

- 7.3.1 Lymphoma in sentinel lymph node
- 7.3.2 Colon mucosa in prostate gland

## 7.4 Effectiveness of distance-based classifier

## 7.5 Comparison of validation methods for FGSM

# Limitations & Future Work

# Conclusions
Chapter 1

Introduction

Over the past few years, machine learning has seen rapid progress, predominantly due to increases in computational power and the availability of new large datasets. Notably, healthcare and medical imaging stand to gain tremendously from machine learning because of the increasing usage of medical devices and digital health records as well as the immense volume of data being generated.

Medical imaging, specifically, can greatly benefit from recent advances in image classification, segmentation and object detection. Numerous studies have demonstrated promising results in medical diagnostics covering radiology, pathology, ophthalmology and dermatology. Clinical studies have shown that AI systems can improve diagnostic quality by providing a second-opinion as well as provide cost-saving by for example performing menial routine tasks in the diagnostic pipeline. Although a small number of these studies have already been translated and deployed as autonomous agents in a clinical setting [1], these studies have also brought legitimate concerns about autonomous systems in safety-critical settings.

It is vital to understand the limitations of automated image analysis pipelines and evaluate the quality of the results being reported. This is especially an issue in the digital pathology field, where it is common to have a limited amount of training data and sample from an incredibly large amount of possible anomalies in images, i.e. the ”long tail” of rare cases is often present. This means many rare samples, almost by definition, are not included in the training set. These samples may range from within-class rare samples to rare classes, and from clinically relevant abnormalities to insignificant deviations.

Modern convolutional neural networks (CNNs) are known to generalize well when the training set and the testing set are sampled from the same data distribution [2, 3]. However, in a real-world clinical setting, there is generally only limited control over the testing data distribution once the system is deployed, and distribution shift is a common occurrence. Recent work has shown that CNNs tend to fail silently for unrecognizable or even unrelated input images by making highly confident predictions. These results are unsurprising because the models were not designed to solve these problems.

In this sense, deep learning models perform local generalization well, but exhibit erratic predictions far outside the space of training examples. It tends to be more akin to template matching or a locality-sensitive hashing function, than it is to a broad generalization function that performs abstraction. Such mistakes, however, can be unsafe; a classifier could give the wrong medical diagnosis with such a high confidence that the data case is not flagged for further scrutiny by a human and/or additional examinations, possibly resulting in an inaccurate patient treatment. Constructing a principled method to detect such unreliable behavior and having statistical guarantees about how often they will occur are critical to building safe and predictable systems.

Concretely, let us consider a machine learning model that is trained on distribution $P_X$, and
deployed on a possibly different "in-the-wild" test distribution, $P_W$. An important assumption is that a large amount of labeled data is available at training time, but little or no labeled data at test time. Our aim is to make sure that the model performs reasonably on $P_W$, in the way that (1) it often performs well on $P_W$, and (2) it indicates when it is performing poorly or has been given an anomalous input. The system can then ensure that risks are avoided by withholding prediction such that the system stays within safe limits regardless of the inputs encountered.

In this thesis, we consider the problem of distinguishing new kinds of inputs, i.e. out-of-distribution samples, from "regular" in-distribution samples (i.e. distribution of training samples) for several computational pathology tasks. Let $Q_X$ denote the out-distribution and $P_X$ again the in-distribution, and assume that a neural network is trained on a dataset drawn from the distribution $P_X$. At test time, we draw new samples from a mixture distribution $P_W \times Z$ with $Z \in \{0, 1\}$ where the conditional probability distributions $P_{W|Z=0} = P_X$ and $P_{W|Z=0} = Q_X$ denote in- and out-distribution respectively. The problem then becomes: Given a sample from the mixture distribution $P_{W \times Z}$, can we distinguish whether it comes from $P_X$ or $Q_X$?

The rest of the thesis is structured as follows. First, we develop the background needed to understand current out-of-distribution detection methods and digital pathology, and we discuss related work. In the third chapter, we construct the multiple heads topology that will be used throughout the thesis as a model base. The fourth chapter motivates the use of the multiple heads topology by demonstrating representational diversity between members of the ensemble. In the fifth chapter, we expand on an input preprocessing based on adversarial samples for our model. The sixth chapter discusses introducing a density estimation element by converting the softmax classifier to a generative classifier. In the seventh chapter, we present the results of the proposed model and the baselines on two real-world digital pathology tasks. The final chapters discuss the limitations of the work, and present our conclusions.

## 1.1 Contributions of this thesis

We close the introduction by summarizing what we see as the major contributions of this research.

- We propose a simple yet effective multiple heads topology in CNNs to train an ensemble to yield highly diverse predictive distributions for out-of-distribution inputs. We demonstrate it is a competitive method for the detection of out-of-distribution samples and compare it to the state-of-the-art ensemble based uncertainty quantification methods, including Deep Ensembles and MC dropout.

- We propose canonical correlation analysis (CCA) as a tool to compare the representational diversity of samples in the aforementioned ensemble-based uncertainty quantification methods, and demonstrate a correlation between representational diversity in ensembles and out-of-distribution detection performance.

- We demonstrate improved out-of-distribution detection performance by inverting the method of adversarial sample generation, i.e. the fast gradient sign method (FGSM), and using it to preprocess input images in the multiple heads approach.

- We propose two methods of hyperparameter validation in the real-world case where no out-of-distribution samples are assumed to be available, including generating adversarial samples and using hard-negatives as a conservative proxy.

- We demonstrate the effectiveness of the proposed method on two hard same-manifold out-of-distribution tasks in the digital pathology domain, specifically on the clinically
significant detection of lymphoma in sentinel lymph node images and the detection of colorectal tissue in prostate gland images.
Chapter 2

Preliminaries

2.1 Computational pathology

In recent years, the digitization of microscopic evaluation of stained tissue sections (whole slide images; WSIs) has been made feasible in histopathology due to advancements in microscopic imaging hardware. This allows for remote diagnostics, better accessible archives, and it facilitates consultations between clinician colleagues. Further, there may be an advantage by using computer-aided diagnostics. Recent studies [4, 5] have shown the potential of deep learning models in this field to reduce the workload for pathologists and increasing objectivity of diagnoses, with comparable performance to board-certified pathologist’s on tumor localization tasks [5].

2.1.1 Lymph nodes metastases

An essential element in breast cancer staging is the microscopic examination of lymph nodes that are adjacent to the breast, sentinel lymph nodes, to inspect whether the cancer has metastasized. This time-consuming procedure is performed by board-certified pathologists and can be prone to error due to small tumor size. The automated detection of lymph node metastases could improve sensitivity, cost and objectivity in breast cancer staging.

A rare abnormality that occurs in sentinel lymph node biopsies is lymphoma. While metastatic adenocarcinoma is sought, the coexistence of lymphoma has been reported as well. A recent finding shows that an incidental discovery of lymphoma while searching for metastases has an incidence of about 1% of patients [6]. The small incidence rate coupled with a high clinical relevance make the detection of lymphoma in sentinel lymph nodes a relevant use case for the task of out-of-distribution detection.

2.1.2 Prostate biopsy

When an examination is required for the presence of prostate cancer, often a prostate biopsy is performed, which is the removal of a number of samples from the prostate gland using a small hollow needle-core. Incidentally sampled colorectal tissue is a sporadic non-prostatic finding in the specimens, specifically colonic mucosa. Although the finding is not clinically relevant for a pathologist per se, it has been reported at the RadboudUMC center that the presence of colonic mucosa tends to cripple classification performance of conventional segmentation networks, such as the U-Net. Therefore detecting such incidental findings may improve the reliability of automated diagnostics, such as the automated segmentation of epithelium in prostate tissue [7].
2.2 Uncertainty in machine learning

Machine learning models can be used for a wide range of applications such as breast cancer diagnosis from mammogram images, autonomous driving and classifying cat breeds. For example, given a number of pictures of cat breeds as training data, when a user feeds a photo of their cat, the model should return a prediction with a high confidence. But what should the model output if a user upload a photo of a dog and wants the model to decide on a dog breed?

The above example is what is called out of distribution data, where the model has been trained on the task of distinguishing between different cat breeds, but has never seen a dog before. So a photo of a dog would lie outside of the distribution the model was trained on. There are more serious real-life settings imaginable, such as in CT scans with abnormalities or in self driving cars with traffic signs that the model has not seen before. In these cases, we may perhaps want the model to return a predict, but also to return additional information that the data case lies outside of the data is has been trained on. That is, we want it to pass on a high level of uncertainty, or a low level of confidence.

There are multiple types of uncertainty identified [8–11] in the literature, and this type of uncertainty is considered by Bayesian approaches [12] to be epistemic uncertainty (or model uncertainty). Epistemic uncertainty measures the uncertainty in estimating the model parameters given the training data. Is is an "unknown-unknown", and measures how well the model is matched to the data in terms of model structure and parameters. Further, it is reducible as the size of the training data increases. The other type of uncertainty is aleatoric uncertainty, which is irreducible uncertainty that arises from the natural complexity of the data, e.g. from class overlap or label noise. It is considered a "known-unknown", where the model understands the data and can state whether a given input is difficult to classify (unknown) with some confidence.

We argue that out-of-distribution data is implicitly modeled through epistemic uncertainty, and conflates the uncertainty about the model parameters and the test data that it is unfamiliar with, although a thorough decomposition of the uncertainty is beyond the focus of this thesis. Together, epistemic and aleatoric uncertainty can be used to induce predictive uncertainty, the confidence we have in a prediction.

2.3 Uncertainty via ensembles

Here we describe recent approaches to predictive uncertainty quantification.

Let us consider a distribution $p(x, y)$ over inputs $x$ and targets $y$. For the sake of this work, let $x$ correspond to images, and $y$ class labels. The predictive uncertainty of a classification model $p(y = \omega_c | x^*, D)$ in the Bayesian framework trained on a dataset $D = \{x_j, y_j\}_{j=1}^N \sim p(x, y)$ results from both aleatoric and epistemic uncertainty. The estimate of epistemic uncertainty is described by the posterior distribution over the parameters given the data and aleatoric uncertainty is described by the posterior distribution over the targets given a set of model parameters $\theta$, or

$$p(y = \omega_c | x^*, D) = \int p(\omega_c | x^*, \theta)p(\theta | D)d\theta$$

(2.1)

with the first term in the integral representing aleatoric uncertainty and the second epistemic uncertainty. We can find the expected distribution $p(y = \omega_c | x^*, D)$ by marginalizing out the parameters $\theta$. However, computing the integral in Equation 2.1 is computationally intractable for neural networks. There are many methods of approximating this posterior, and the main categories are sampling-based techniques and variational inference [13]. We will now discuss a few of the current popular sampling-based methods for neural networks.
The approximation by sampling uses an ensemble,

\[
p(y = \omega_c \mid x^*, D) \approx \frac{1}{M} \sum_{i=1}^{M} p(y = \omega_c \mid x^*, \theta^{(i)}), \theta^{(i)} \sim q(\theta)
\]

(2.2)

where members \( p(y = \omega_c \mid x^*, \theta^{(i)}) \) in the ensemble \( \{p(y = \omega_c \mid x^*, \theta^{(i)})\}_{i=1}^{M} \) are sampled from an approximate model posterior \( q(\theta) \). Each sample is a categorical distribution \( y^* = [p(y = \omega_1), ..., p(y = \omega_K)]^T \) with \( K \) the number of object classes.

Given an ensemble from such a distribution, the uncertainty can be indicated by computing the predictive entropy of the expected distribution \( p(y = \omega_c \mid x^*, D) \). However, this does not allow us to distinguish between aleatoric uncertainty or epistemic uncertainty. Alternatively, we can compute the sample variance \( \text{Var}(p(y = \omega_c \mid x^*, D)) \) for which [12] establishes that it obtains epistemic uncertainty, with consistent predictions for in-domain inputs and diverse predictions for out-of-distribution inputs.

### 2.3.1 MC dropout

A recent development that has seen wide adoption in this area is Monte-Carlo dropout (MC dropout). It generates the ensemble of Equation 2.2 using multiple stochastic forward passes. The stochasticity is induced by Monte Carlo sampling of the dropout masks. Specifically, given a new input \( x^* \), the output \( y^* \) is computed with stochastic dropouts at each layer; in other words randomly drop out each channel in the network with a fixed probability \( p \). The stochastic feedforward is repeated to obtain \( \{y^*_1, ..., y^*_M\} \) based on which the sample variance is computed. Note that the method requires \( M \) feedforward passes for each image to obtain an uncertainty estimate.

### 2.3.2 Deep Ensembles

Another approach based on explicitly training an ensemble of neural networks is Deep Ensembles [14](DE). It works conceptually very simple by independently training randomly initialised instances of a model on same (randomly ordered) data, or in the case of bootstrapping on different random subsets of the data. Note that the uncertainty estimates based on \( q(\theta) \) do not have the usual Bayesian interpretation in this case. In addition to this, an adversarial training schema [15] is proposed to smooth the predictive distribution. The method achieves comparative performance with MC dropout for a number of classification tasks.

A relatively large drawback of the method is that, in addition to \( M \) feedforward passes during test time, it requires training \( M \) networks independently, which can be prohibitive in a limited resource setting.

To combat this, we experiment with speeding up the training of multiple models by using a recent technique, Fast Geometric Ensembling [16](FGE). The work is based on a discovery that the local optima for modern deep neural networks are connected by very simple curves, such as a polygonal chain with only one bend, and they show that such mode connectivity holds for a wide range of deep neural networks. It works using a training scheme that finds geometric paths in the loss surface of near-constant accuracy between modes of large deep networks. First a model is trained to convergence for \( N \) epochs using a regular training procedure. The remainder of the training run (typically \( \sim N/4 \)) is performed with a short cylical cosine annealing learning rate schedule. New models are saved and added to the ensemble at the lowest learning rate point of each cycle. The work shows improved classification performance on CIFAR-10 and CIFAR-100.
While segmentation outputs produced by these methods are consistent, they are not necessarily diverse and they are not typically able to learn rare variants since the members are trained independently.

2.3.3 Reconstruction-based and density-based methods

Two different classes of approaches are reconstruction-based methods and density estimation. Reconstruction methods generally use auto-encoders that aim to reconstruct normal data well while producing high reconstruction errors for out-of-distribution data. These are widely used in medical imaging settings since they naturally allow for pixel-wise uncertainty estimation. [17] use a generative adversarial network [18] (GAN) to estimate an uncertainty score. Based on the fact that a trained GAN can only produce samples from its learned data distribution, they design an iterative backpropagation algorithm that finds the closest match to the sample of interest that the GAN can produce. The uncertainty score is then derived from the similarity between the generated and real sample. Many autoencoders have been used for out-of-distribution detection in medical imaging. [19] uses variational auto-encoders [20] (VAEs) and used the reconstruction error to localize MS lesions. [21, 22] shows that a combination of a VAE with an adversarial loss on the latents improves performance in detecting brain anomalies using a pixel-wise reconstruction error. Despite their frequency in related work, reconstruction-based methods have seen no formal treatment regarding the reconstruction error, obscuring the interpretation and the comparability of their scores.

Alternatively, density-based methods give a probability estimate for each data case, which simplifies ordering the cases based on an uncertainty score. VAEs are able to alleviate the curse of dimensionality problem that many previous methods (e.g. OC-SVM [23] and PCA [24]) have struggled with in high dimensional data settings [25]. State-of-the-art classification models however outperform VAEs in classification tasks, such that it may cripple classification performance to use VAEs for both tasks of out-of-distribution detection and classification. In addition, if they were to be used in tandem, an uncertainty estimate for the VAE may not translate to uncertainty for the classifier for the same sample due to differences in optimization. Finally, although a VAE can produce segmentation maps, it does not generate a pixel-wise uncertainty map which is essential in many medical imaging settings to pinpoint locations of interest.
Chapter 3

Multiple hypotheses model

In this chapter we formulate a simple yet effective topology for CNNs inspired by [26, 27] that consists of a shared architecture followed by a bifurcation of $M$ identical isolated subgraphs with different initializations near the end of the DAG, referred to as M-Heads. We will argue that these subgraphs are an ensemble, of which the members can be implicitly specialized on modes in the density of the training data with a modification to the training procedure.

This emerging specialization implies a representational decorrelation and, as we hypothesize, allows for the better capturing of ambiguity in test samples. It does this by jointly producing a set of multi-modal hypotheses, outputs that cover the space of high probability predictions. Further, many modern models tend to exhibit "mode-seeking" behavior in order to reduce loss over a dataset [28]. With diverse solution sets, the coverage of lower density regions of the solution space (or mode coverage) is improved without a drop in performance for the highest density regions.

3.1 Network architecture and training procedure

Given a dataset of input-target pairs $\{(x_i, y_i) | x_i \in \mathcal{X}, y_i \in \mathcal{Y}\}$, we consider the task of training an ensemble of $M$ heads that together produce a set of hypotheses, i.e. a function $g : \mathcal{X} \rightarrow \mathcal{Y}^M$. See Fig. 1 for an illustration. The goal is to train the ensemble such that we obtain minimal loss and implicit decorrelation between members. Since there is only ground-truth available for the single true target, we must design a method that predicts multiple hypotheses $\mathcal{Y}^M$ with each having meaningful information. To this end, we use a meta-loss $\mathcal{M}$ [26] that acts on top

![Figure 1: Illustration of the M-Heads setup. (a): Prediction process. The arrows denote the flow of operation, the blue blocks represent feature maps and the purple boxes represent the softmax outputs. Predictions are pooled in the final step. (b): Training process. Note the soft Kronecker delta that distributes the gradient signal according to Equation 3.2.](image-url)
of a given standard loss $\mathcal{L}$ (e.g. cross entropy loss) for a single datapoint $(x, y)$:

$$
M(g(x), y) = \sum_{i=1}^{M} \delta_i \mathcal{L}(g^i(x), y)
$$

(3.1)

where $g^i(x)$ is the softmax output of the $i$'th head, and

$$
\delta_i = \begin{cases} 
1 - \epsilon & \text{if } i = \arg \min_j \mathcal{L}(g^j(x), y) \\
\epsilon (M-1) & \text{else}
\end{cases}
$$

(3.2)

where $\epsilon$ is the assignment relaxation constant.

In other words, $\delta_i$ acts as a soft Kronecker delta such that a fraction $1 - \epsilon$ of the gradient signal flows through the head with the best hypothesis according to the ground truth. The other heads receive the remaining signal, summing the fractions up to 1. Preliminary experiments have shown that if instead a hard Kronecker delta is used such as in the Winner-Takes-All loss by [27], the training collapses to the prediction of a single mode $g^k$. This is due to the initialization of the other heads that is too far from the targets $y$ such that all data points are closer to the single mode. Consequently, the function $g^k$ is optimized and the remaining functions $\{g^i \mid \forall i \neq k\}$ will never receive a gradient signal.

Note that this loss also deviates from [26] that computes $\delta_i$ over a batch where we compute it per sample instead, improving performance in preliminary experiments. In their case, it limits the upper bound of the batch size severely, since the sample distribution in large batches approaches the full data distribution, crippling specialization.

Adapted from [29], we add randomness by randomly dropping out full predictions with a low probability to prevent weaker predictions from vanishing. Specifically, we drop out the prediction of the $i$'th head with $i = \arg \min_j \mathcal{L}(g^j(x), y)$ with a probability $r$.

Importantly, we do not induce any explicit repulsive potentials in this loss to e.g. reduce the mutual information between the hypotheses. The training procedure implicitly pushes the heads to seek out lower density regions of the solution space. Specifically, we can interpret the procedure as an instance of the Expectation-Maximization (EM) algorithm. In the E-step the soft (albeit discontinuous) assignment is computed between the true target $y_i$ and the prediction $g^j(x_i)$ and during the M-step the predictor $g^j$ is updated to better predict the target $y_i$ and hence move $g^j$ in feature space to the closest mode that contains $x_i$ in the image space.

### 3.1.1 Depth of heads

The related work of [26, 28] either trains $M$ full models with no shared weights and effectively partitions the sample space, or shares all weights except for copying the last layer $M$ times. We find that these strategies are not the best ones per se, and experiment with multiple depths of the heads. Specifically, we train a Wide ResNet network [30] with a depth of 16 and a widen factor of 8, henceforth referred to as WRN-16-8, on the CIFAR-10 dataset [31] and evaluate the model’s performance on depths $d \in \{1, 2, 4, 8, 16\}$ with $M = 8$. The hyperparameters $\epsilon$ and $r$ are validated for each depth $d$ by performing a grid-search on $\epsilon \in \{0.01, 0.05, 0.1, 0.2, 0.3, 0.4\}$ and $r \in \{0.001, 0.005, 0.01, 0.02, 0.05\}$.

Table 1 reports the results. The best classification performance is found for a depth $d = 4$, supporting the notion that a depth larger than 1 and smaller than the full network depth

<table>
<thead>
<tr>
<th>Case</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d = 1$</td>
<td>90.1</td>
</tr>
<tr>
<td>$d = 2$</td>
<td>90.6</td>
</tr>
<tr>
<td>$d = 4$</td>
<td>90.9</td>
</tr>
<tr>
<td>$d = 8$</td>
<td>90.0</td>
</tr>
<tr>
<td>$d = 16$</td>
<td>89.2</td>
</tr>
</tbody>
</table>

Table 1: Classification performance on CIFAR-10 in terms of Top-1 accuracy (%)
improves performance. We further observe that the optimal assignment relaxation constant $\epsilon$ is strongly dependent on the depth of the head, where a larger depth requires a significantly higher epsilon. For example, for $d = 1$ a value of 0.05 works best, and for $d = 8$ it is $\epsilon = 0.3$. We find that for a sufficiently small $\epsilon$ value (e.g. 0.05 for $d = 8$), the updates collapse to a single head. We refer to this phenomenon in subsequent chapters as heads collapse, which we define as when a single head accounts for more than 90% of the best predictions. The issue comes from the fact that the additional layer initializations make it more likely that all data points are closer to a single head in label space. This effect is also present when we fine-tune a pretrained network with randomly initialized heads, where due to the converged shared weights there is little joint learning between the initialized weights and the shared weight, and the distribution of best predictions quickly approaches collapse to a single head as well. As such, it is necessary to train "from scratch" to prevent heads collapse.
Chapter 4

Representational diversity in ensembles

With ensemble-based uncertainty estimation methods, we hypothesize that it is important to have a high representational diversity between members of the ensemble. The key intuition is that given a highly uncertain or an out-of-distribution sample we wish to capture the ambiguity which can be interpreted as the disagreement between members. For members to respond in an identical manner, it is evident that the disagreement will be minimal, and vice versa for a highly diverse ensemble, we expect the disagreement to be high in cases of ambiguity and to be relatively low in the case of unequivocal samples.

If we consider Bayesian Model Averaging, where ensembles are a finite sample approximation to integration over the model space [32], and Model Combination, where ensembles should enrich the hypothesis space considered by the base model and are representationally richer [33], the importance of decorrelation between representations of ensemble members follows naturally. For a decorrelated ensemble, we not only expect task performance to improve for averaged predictions, but also to better capture ambiguity.

The difficulty in comparing representations between members mainly stems from the observation that channels in neural networks are not directly aligned, i.e. there is no neuron-to-neuron alignment for two separately trained networks even though they may have an identical topology. In this chapter, we introduce canonical correlation analysis (CCA) as a method for studying the similarities of representations learned by multiple networks. Since CCA is invariant to affine transforms, it enables us to find common structure across representations which may seem dissimilar on the surface.

4.1 Canonical correlation analysis on representations in neural networks

Canonical correlation analysis identifies the linear relationship between two sets of multidimensional variates with a maximized correlation. These variates are observations arising from neural network predictions; they are channel activation vectors [34] over a dataset $X$, that is they denote the outputs that a single channel $z$ has on $X$. For a dataset $X = \{x_1, \ldots, x_m\}$, the channel $z$ outputs a vector with scalars $z(x_1), \ldots, z(x_m)$. Where a single channel is one multidimensional variate, a neural network layer denotes a set of multidimensional variates. We apply CCA to two layers, $L_1$ and $L_2$, which come from two different members of the ensemble and are identical in topology, to determine the similarity between the two layers and consequently the ensemble members. Further, to help remove channels that are noisy or have a low variance, we preprocess the channels by applying singular value decomposition (SVD) to determine the number of channels needed to explain 99% of the variance in the observations.
4.1.1 Mathematical interpretation of CCA

To consider the formal model of CCA, let \( L_1 \) and \( L_2 \) both be \( m \times n \) dimensional matrices representing \( m \) multidimensional variates. The goal is to find vectors \( w, s \) (both in \( \mathbb{R}^m \)) such that the following dot product is maximized:

\[
\rho = \frac{\langle w^T L_1, s^T L_2 \rangle}{\|w^T L_1\| \cdot \|s^T L_2\|}
\]

(4.1)

We can rewrite Equation 4.1 by assuming that \( L_1, L_2 \) are centered, and letting \( \Sigma_{L_1,L_1}, \Sigma_{L_2,L_2} \) denote their respective \( m \times m \) covariance matrices, and \( \Sigma_{L_1,L_2} \) denote the cross covariance matrix, such that

\[
\frac{\langle w^T L_1, s^T L_2 \rangle}{\|w^T L_1\| \cdot \|s^T L_2\|} = \frac{w^T \Sigma_{L_1,L_2} s}{\sqrt{w^T \Sigma_{L_1,L_1} w} \sqrt{s^T \Sigma_{L_2,L_2} s}}.
\]

(4.2)

By changing bases with \( w = \Sigma_{L_1,L_1}^{-1/2} u \) and \( s = \Sigma_{L_2,L_2}^{-1/2} v \), we get

\[
\frac{w^T \Sigma_{L_1,L_2} s}{\sqrt{w^T \Sigma_{L_1,L_1} w} \sqrt{s^T \Sigma_{L_2,L_2} s}} = \frac{u^T \Sigma_{L_1,L_1}^{-1/2} \Sigma_{L_1,L_2} \Sigma_{L_2,L_2}^{-1/2} v}{\sqrt{u^T u} \sqrt{v^T v}}
\]

(4.3)

and we find a solvable SVD equation:

\[
\Sigma_{L_1,L_1}^{-1/2} \Sigma_{L_1,L_2} \Sigma_{L_2,L_2}^{-1/2} = U \Lambda V
\]

(4.4)

with \( u,v \) the first vectors in \( U,V \) and the canonical correlation coefficient \( \rho \in [0,1] \) is the top singular value of \( \Lambda \) and informs us to what degree \( w^T L_1 \) and \( s^T L_2 \) are correlated.

The CCA output is a collection of pairwise orthogonal singular vectors \( u_i, v_i \in \mathbb{R}^m \) corresponding to a correlation coefficient \( \rho_i \in [0,1] \) such that there are \( m \) correlation coefficients.

4.1.2 CCA distance

Next, to construct a CCA distance measure, we can combine the correlation coefficients by averaging:

\[
d_{CCA}(L_1, L_2) = 1 - \frac{1}{m} \sum_{i=1}^{m} \rho_i
\]

(4.5)

However, this measure assumes equal importance to the representation of \( L_1 \) for all \( m \) CCA vectors, although [35, 36] has shown that CNNs do not rely on all channels of a layer to represent high performance solutions, or at least not evenly. Therefore a weighted mean is proposed in which canonical correlations that carry more importance to the latent representation have a higher weight. [34] propose to determine these weights by a method called projection weighting, constructed on the intuition that the CCA vectors that correspond to a larger proportion of the original outputs also are more important to the latent representation. Specifically, let \( L_1 \) again have channel activation vectors \( [z_1, \ldots, z_m] \) and CCA vectors \( [h_1, \ldots, h_m] \), then we compute the proportion of how much each \( h_i \) accounts for the original output as

\[
\alpha_i = \sum_j |\langle h_i, z_j \rangle|
\]

(4.6)

After normalizing the weights \( \alpha_i \) to sum up to 1, the projection weighted CCA distance becomes

\[
d_{CCA}(L_1, L_2) = 1 - \sum_{i=1}^{m} \alpha_i \rho_i
\]

(4.7)
Note that this distance is not symmetric, so technically it is a pseudo-distance. For our experiments, this is resolved by averaging over all pairwise distances.

An interesting result from [34] using CCA distances is that for regular ensembles (such as in Deep Ensembles) the higher the classification performance of the members is, the more similar their solutions become. That means that if we do not explicitly specialize ensembles such as in M-Heads, high classification performance and representational diversity may not be naturally jointly optimizable. In other words, high performance members may be pushed towards the same solution, and if we do not account for specialization, there is a trade-off between sample quality and sample diversity.

4.2 Representational diversity in ensembles on CIFAR-10

To study the representational diversity of the M-Heads setup, we compare its diversity as measured by the CCA distance measure from Equation 4.7 with the aforementioned alternative ensemble based methods. As in Section 3.1.1, we use the WRN-16-8 network and train on the CIFAR-10 dataset. To obtain a better intuition of the interaction between ensemble size and representational diversity, we examine the models performance in all cases when drawing a different number of samples \( n \in \{2, 4, 8, 16\} \) from each of them.

The training routine for the Deep Ensembles (DE) and the Fast Geometric Ensembling (FGE) is adjusted according to the prescribed formulas [14, 16], and a sample is defined as a model in the ensemble. For the FGE method, the first phase takes up 75% of the training budget. For the Deep Ensembles method, the models are randomly initialized and the training data is shuffled such that each model is trained in a slightly different manner. Further, for the DE method, we do not employ adversarial training since it is both computationally expensive and it does not improve training in preliminary experiments. For MC dropout, the dropout layers are placed between the two convolution sub-blocks of each ResNet block and the dropout rate is set to 0.3 as in [30]. The dropout masks are randomized independently per sample for each batch and a sample is defined as a dropout mask initialization. The batch size for the baselines is set to 64 for all cases, except for M-Heads for which it is reduced to 32 since this was found to improve training by preventing head collapse. For the M-Heads approach, we follow the same hyperparameter search for \( r \) and \( \epsilon \) as in Section 3.1.1.

In all cases, we compute the \((M - 1)!\) pairwise CCA distances of the ensemble on the 50k images in the training set of CIFAR-10. Fig. 2 reports the distances and Table 2 reports the classification performance. The results in Fig. 2 indicate that the M-Heads approach consistently exhibits greater representational diversity between samples than all comparison methods in terms of the pairwise CCA distance. We observe that in the case of M-Heads the CCA distance increases for later layers, which hints at the need for depth in specialization which perhaps can be increased by further adding depth to the heads. At the softmax layers, all cases converge to nearly identical solutions, which corresponds to the near-zero training loss of all samples.

The FGE ensemble displays significantly less representational diversity than the DE approach, although the classification performance of FGE is greater than that of DE, shown in Table 2, indicating that representational diversity and classification performance do not necessarily complement each other. This, of course, is evident when we consider the trivial case of random weights for each ensemble member, resulting in trivial classification performance and high pairwise CCA distances. It is further notable that FGE obtains both a higher ensemble accuracy and a higher CCA than MC dropout with the same training and inference budget.

Finally, comparing M-Heads to the strongest baseline Deep Ensembles results, we see that
Table 2: Classification performance on CIFAR-10 in terms of Top-1 accuracy (%) for M=8 samples. The ensemble accuracy is determined by average pooling the predictions of the samples. Confidence bounds of the sample accuracy represent mean ± standard deviation of samples.

<table>
<thead>
<tr>
<th>Case</th>
<th>Ensemble</th>
<th>Single member</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deep ensembles</td>
<td>92.0</td>
<td>90.1±0.4</td>
</tr>
<tr>
<td>Fast geometric ensembles</td>
<td>93.1</td>
<td>92.3±0.3</td>
</tr>
<tr>
<td>MC dropout</td>
<td>91.6</td>
<td>90.2±0.1</td>
</tr>
<tr>
<td>M-Heads</td>
<td>92.8</td>
<td>90.9±0.3</td>
</tr>
</tbody>
</table>

the larger diversity is predominantly due to the increased specialization in samples due to the M-Heads training subroutine in Section 3.1, which allows the model to capture an increased number of modes.

4.2.1 Diminishing diversity of M-Heads

To study the limits of the M-Heads approach, we increase the number of heads to M=16 and find that the diversity diminishes. See Figure 2 for a comparison between multiple numbers of heads. These diminishing diversity returns can be interpreted by the concept of mode coverage: each additional head can learn an additional mode in the data, but after a number of heads the marginal specialization benefit decreases since even the low density regions can already be largely explained by the existing heads. The specialization in the additional head will therefore overlap increasingly more with the other heads, and the average representational diversity decreases.
Chapter 5

Input pre-processing using FGSM

5.1 Adversarial examples

In this chapter, we explain how the concept of adversarial attacks can be harnessed to improve out-of-distribution detection. A few years ago, it was found [37] that several state-of-the-art neural networks are vulnerable to adversarial examples, i.e. examples that are misclassified although they are only slightly different from correctly classified examples of the data distribution. Following work showed [15] that adversarial examples can be explained as a property of high-dimensional dot products and contrary to earlier work, it proposes that they are a result of models being too linear, rather than too nonlinear.

5.1.1 Model linearity

The linear explanation of adversarial examples goes as follows [15]. If we consider that the precision of an individual input feature is limited (say 8 bits per pixel), it follows that a classifier $F$ will not behave in a different manner to an input $x$ than to an adversarial input $\tilde{x} = x + \eta$, i.e. $F(x) = F(\tilde{x})$, as long as each element of $\eta$ (the perturbation) is smaller than the precision of the features. Specifically, the classifier assigns the same class to $x$ and the perturbed $\tilde{x}$ if and only if $||\eta||_1 < \epsilon$, where $\epsilon$ is chosen such that it corresponds to the precision of the input feature (e.g. $\epsilon = 0.007$ or $2*1/255$ in the case of the 8 bits pixel encoding example).

If we then consider the dot product of the weight vector $w$ with an adversarial input $\tilde{x}$: $w^T \tilde{x} = w^T x + w^T \eta$, we can see that the activation grows by $w^T \eta$ when adversarially perturbed. This increase can be maximized constrained by the max norm on $\eta$ such that $\eta = \text{sign}(w)$. For an $n$-dimensional weight vector $w$ with an average magnitude of $m$, the perturbation increases the activation by $\epsilon m n$.

The crucial corollary is that the max norm of $\eta$ is independent of the dimensionality of $w$, such that the activation increase actually grows linearly with $n$. So for high dimensionality models, many small changes can be made to the input that accumulate as a large change in the output (in our case: of the classifier $F$).

5.1.2 A simple attack: Fast Gradient Sign Method

Now that we have established that linear models are susceptible for linear adversarial perturbations, it is further hypothesized that neural networks are too linear to be robust for these perturbations as well. Activation functions such as ReLU [38] are designed for optimization purposes to behave mostly linearly, and the same goes for nonlinear activation functions such as the sigmoid, which optimizes best in its non-saturated (linear) region. If neural networks
exhibit mostly linear behavior, it is reasonable to suggest that these simple perturbations also work in an adversarial sense for these models.

Consider a neural network with parameters $\theta$, an input $x$ with ground truth $y$ and a loss function $J(\theta, x, y)$ to train the network. We can construct a perturbation $\eta$ that satisfies the optimal max norm constraint from Section 5.1.1

$$\eta = \epsilon \text{sign}(\nabla_x J(\theta, x, y))$$  \hspace{1cm} (5.1)

This method of adversarial attack is referred to as the Fast Gradient Sign Method (FGSM) and is computed by backpropagating the gradient of the loss function for the true class with respect to the input. It is very effective in fooling CNNs by generating misclassified samples, and as such is evidence for the linearity explanation for adversarial examples. For example, [15] shows that it is able to fool a standard convolutional neural network trained on CIFAR-10 such that an average probability of 96.6% is assigned to incorrect labels.

5.2 FGSM for detection of OOD samples

We next consider how the fast gradient sign method can be used for detecting OOD samples instead of generating adversarial samples. Essentially, the goal of generating adversarial samples is decreasing the softmax output for the true class and, for targeted attacks, increasing the output for the target class. By changing the direction of the perturbation $\eta$ to its opposite, we can increase the softmax output for the true class, or in fact any class. See Algorithm 1 for the routine.

<table>
<thead>
<tr>
<th>Algorithm 1: Input pre-processing using FGSM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> Test image $x$, trained classifier $F$ and perturbation factor $\epsilon$.</td>
</tr>
<tr>
<td><strong>Output:</strong> Perturbed image $\tilde{x}$.</td>
</tr>
<tr>
<td>1 $S \leftarrow F(x)$</td>
</tr>
<tr>
<td>2 $\tilde{y}_x \leftarrow \text{argmax}(S)$</td>
</tr>
<tr>
<td>3 $\eta \leftarrow -\epsilon \text{sign}(\nabla_x J(F, x, \tilde{y}_x))$</td>
</tr>
<tr>
<td>4 $\tilde{x} \leftarrow x + \eta$</td>
</tr>
<tr>
<td>5 return $\tilde{x}$</td>
</tr>
</tbody>
</table>

It is hypothesized that "positive" perturbation (i.e. $\eta < 0$) has a larger effect on in-distribution images than on OOD images. If this is indeed the case, FGSM can be effectively used to separate their respective softmax output further by increasing the score for the highest predicted class, or formally: $\max(S(\tilde{x})) \geq \max(S(x))$, where $S_x \in \mathbb{R}^N$ is the softmax output for the classifier with $N$ classes of input image $x$ and $\{s_i \in S \mid 0 < s_i < 1\}$. To understand why this may be the case, consider the first order Taylor expansion of the log softmax function for $\tilde{x}$:

$$\log S_{\tilde{y}}(\tilde{x}) = \log S_{\tilde{y}}(x) + \epsilon \| \nabla_x \log S_{\tilde{y}}(x) \|_1 + o(\epsilon)$$  \hspace{1cm} (5.2)

It is shown empirically that the distribution of the L1 norm of the gradient of the log-softmax function [39] with respect to the input $x$, or $\| \nabla_x \log S_{\tilde{y}}(x) \|_1$, typically has larger values for in-distribution images than most of the out-of-distribution images. The effect is illustrated in Figure 3. Consider an in-distribution image $x_1$ (red) and an out-of-distribution image $x_2$ (blue) with $S(x_1) \approx S(x_2)$. By performing the routine in Algorithm 1, the softmax score of the perturbed in-distribution image $S(\tilde{x}_1)$ tends to be significantly higher than that of the out-of-distribution image $S(\tilde{x}_2)$. 19
After input processing, the in-distribution image can have a much larger value on the norm of softmax gradient than the out-of-distribution image. For example, setting the perturbation magnitude is too large.

The effect of softmax scores on detecting out-of-distribution examples in clinical scenario. Sabokrou et al. (2016) train a convolutional network on deep models have been proposed. Schlegl et al. (2017) train a generative adversarial networks to detect out-of-distribution examples in low-dimensional space. Our method does not require retraining the neural network and significantly improves the understanding of these phenomenon. We provide in Figures 5 (d) the conditional expectation results in a much larger value on the norm of softmax gradient than the out-of-distribution image.

Our approach leverages the following two interesting observations to help better distinguish between in- and out-of-distribution images. All these works require enlarging or modifying the neural networks. In contrast, the proposed method can be applied to an existing well-trained model.

The clustering method is based on the statistical distance, and declares an example to be out-of-distribution if it locates in the low-density areas. The density estimation approach uses probabilistic models to estimate the in-distribution density and declares a test example to be out-of-distribution if it locates in the low-density areas. In various contexts (see the survey by Pimentel et al. (2014)), the density estimation, nearest neighbor and clustering analysis are widely used in detecting low-dimensional out-of-distribution examples (Chow, 1970; Vincent & Bengio, 2003; Ghoting et al., 2008; Devroye et al., 2006).

In recent years, out-of-distribution detectors based on deep models have been proposed. Theis et al. (2015) train a generative adversarial networks to detect out-of-distribution images. Adapted from: [39].

Dezfooli et al., 2017, which show that the softmax scores tend to change significantly if small perturbations does not change the predictions of the neural network, the softmax scores between in- and out-of-distribution images can be separable from each other after input preprocessing. Our observation is consistent with the fact that CIFAR-10 images (in-distribution) tend to have larger values on the norm of gradient than the out-of-distribution images.

5.2.1 FGSM for M-Heads

Applying the FGSM routine for the multiple heads approach is possible through either (1) repeating the routine for each head or (2) adapting the routine to accommodate perturbation for all heads at once. Since option (1) scales linearly with $M$ and we have limited computational resources, we opt for the more suitable second option. The routine is adjusted for (2) as can be seen in Algorithm 2. That is, the resultant input image $\tilde{x}$ is perturbed such that the softmax scores are affected by the perturbation factor $\eta$.

Note that since we perturb the image $M$ times, the perturbation magnitude $\epsilon$ should also scale with $1/M$ to satisfy the max norm constraint.

**Algorithm 2:** Input pre-processing for M-Heads

**Input:** Test image $x$, trained M-Heads classifier $F$, perturbation factor $\epsilon$ and the number of heads $M$.

**Output:** Perturbed image $\tilde{x}$.

1. $S \leftarrow F(x)$
2. $\eta \leftarrow 0$
3. for $i \leftarrow 0$ to $M$
4.    $\tilde{y}_{x,i} \leftarrow \text{argmax}(S_i)$
5.     $\eta \leftarrow \eta - \epsilon \text{ sign}(\nabla_{x,i} J(F, x, \tilde{y}_{x,i}))$
6. end
7. $\tilde{x} \leftarrow x + \eta$
8. return $\tilde{x}$

5.3 Validation of $\epsilon$

Obtaining optimal hyperparameters for the task of detection of OOD samples is fundamentally challenging, since the task does not assume the availability of any OOD samples a priori. Although related work surprisingly does assume that a limited set of uniformaly sampled OOD
samples from the test set is accessible, this simplifies the task incorrectly. Since the distribution of OOD samples cannot be sampled uniformly in a real-world setting, this simplification does not guarantee good performance for OOD samples that do not resemble the given set.

Alternatively, we can use two methods of generating proxy OOD validation samples without real samples. The first method employs the extraction of hard-negative samples of the validation set during training. We select the in-distribution images that were consistently misclassified or where the predictions have a high variance during the last $n$ epochs, and select the value for $\epsilon$ that has the best OOD proxy detection performance. The secondary method consists of generating adversarial samples based on the validation set, by performing an FGSM step with $\eta > 0$, i.e. a “negative” FGSM step as in [15].
Chapter 6

Distance-based OOD detection

The backbone in many of the related studies [39, 40] has been the standard softmax classifier. Following the discussion in Section 2.3.3, a different avenue of research is to use density estimation. In this chapter, we explore a combination of the two directions by measuring the probability density of test images on the feature space of a neural network by converting the softmax classifier to a distance-based classifier.

The assumption is that features of the training set can be successfully fitted by a class-conditional Gaussian distribution. For a visual interpretation, Fig. 4 shows embeddings of the final features of a ResNet from CIFAR-10 test samples by t-SNE [41], where the colors correspond to the different classes. The visualization supports the assumption, since the classes tend to be clearly separated in the feature space.

![Figure 4: Visualization by t-SNE of final features from ResNet trained on CIFAR-10, adapted from [42].](image)

Once the class-conditional Gaussians are fitted, the confidence score for a test sample can be measured by the minimal Mahalanobis distance to any class distribution. To understand why a distance measure may outperform the softmax function in terms of detecting OOD samples, examine the illustration in Fig. 5 for a two-dimensional classification problem. For the softmax classifier, we have a single decision boundary where samples close to the boundary can be classified as OOD. However samples far from the decision boundary will not be considered outliers, although they may deviate strongly from the data distribution. As seen in the bottom figure, fitting a class-conditional Gaussian and using a distance-based measure enables the classifier to capture these outliers as well.

![Figure 5: Illustration of the difference in OOD detection with (top) the softmax classifier and (bottom) the distance-based classifier.](image)
6.1 Mathematical formulation of distance-based detection

6.1.1 Linear discriminant analysis

The softmax classifier defines the posterior distribution \( p(y|x) \) as follows:

\[
p(y = c|x) = \frac{\exp(w_c^T f(x) + b_c)}{\sum_{c'} \exp(w_{c'}^T f(x) + b_{c'})}
\] (6.1)

where \( w_c \) and \( b_c \) are the weights and biases for class \( c \), respectively, and \( f(\cdot) \) is the output of the penultimate layer of the network. To convert the discriminative classifier to a generative classifier, the class conditional distribution \( p(x|y) \) and class prior \( p(y) \) are instead defined to indirectly define posterior distribution by specifying the join distribution \( p(x,y) = p(x|y)p(y) \).

Next, Gaussian Discriminant analysis (GDA) is a simple method that can be used to define the generative classifier by assuming that the class prior follows a Bernoulli distribution – satisfied in our case – and the class conditional distribution follows a multivariate Gaussian distribution, i.e.

\[
p(x|y = c) = \mathcal{N}(x|\mu_c, \Sigma_c)
\] (6.2)

\[
p(y = c) = \frac{\beta_c}{\sum_c \beta_c},
\] (6.3)

where \( \mu_c \) is the mean of \( c \) and \( \Sigma_c \) is the covariance of the Gaussian, and \( \beta_c \) is the (unnormalized) prior for class \( c \).

If we further assume a tied covariance matrix \( \Sigma \) for all classes, the posterior distribution under Linear Discriminant Analysis (LDA) is described by

\[
p(y = c|x) = \frac{p(y = c)p(x|y = c)}{\sum_{c'} p(y = c')p(x|y = c')}
\]

\[
= \frac{\exp(\mu_c^T \Sigma^{-1} - 1/2 \mu_c^T \Sigma^{-1} \mu_c + \log \beta_c)}{\sum_{c'} \exp(\mu_{c'}^T \Sigma^{-1} - 1/2 \mu_{c'}^T \Sigma^{-1} \mu_{c'} + \log \beta_{c'})}
\] (6.4)

(6.5)

If we consider \( \mu_c^T \Sigma^{-1} \) as the weight and \( -1/2 \mu_c^T \Sigma^{-1} \mu_c \) as the bias of it, that is equivalent to the posterior of the softmax classifier of Equation 6.1.

Finally, we can compute the necessary parameters, i.e. the empirical class mean and covariance for the training set \( \{(x_1, y_1), \ldots, (x_n, y_n)\} \), as follows:

\[
\hat{\mu}_c = \frac{1}{n_c} \sum_{i:y_i=c} f(x_i)
\] (6.6)

\[
\hat{\Sigma} = \frac{1}{n} \sum_c \sum_{i:y_i=c} (f(x_i) - \hat{\mu}_c)(f(x_i) - \hat{\mu}_c)^T
\] (6.7)

where \( n_c \) is the number of training samples with class \( c \).

6.1.2 Mahalanobis distance-based classification

Now that we have \( c \) class-conditional Gaussians, we can use the Mahalanobis distance between a sample \( x \) and the closest class-conditional Gaussian to obtain a minimum class distance score:

\[
d_{Mah.}(x) = \min_c (f(x) - \hat{\mu}_c)^T \hat{\Sigma}^{-1}(f(x) - \hat{\mu}_c)\] (6.8)
The Mahalanobis distance is unitless and scale-invariant, so features do not need to be normalized. Specifically, it weighs the Euclidean distance by the standard deviation of the data distribution. Note that a limitation of this method is the strong assumption of Gaussianity of the class-conditional distributions and the lack of outliers in the training data.

### 6.2 LDA for M-Heads

To unify the M-Heads approach with linear discriminant analysis as described in the previous sections, there are two options. The first approach is to fit $M \times c$ class-conditional and head-conditional Gaussian distributions, i.e.

$$p(x|y = c; h = i) = \mathcal{N}(x|\mu_{c,i}, \Sigma_i)$$

(6.9)

The minimum class distance score of Equation 6.8 can then be adjusted to sum over the distances for each head (or return the closest mode with a min(·) operation):

$$d_{Mah.}(x) = \sum_i^M \min_{c} (f_i(x) - \hat{\mu}_{c,i})^T \hat{\Sigma}_i^{-1} (f_i(x) - \hat{\mu}_{c,i})$$

(6.10)

with $f_i(·)$, $\mu_{c,i}$ and $\Sigma_i^{-1}$ the output of the penultimate features, the class-conditional mean and the covariance for head $i$.

Alternatively, we can incorporate the inter-head covariance by combining the penultimate features of each head such that $f(·)$ gives an $Md$-vector, where $d$ is the number of channels in the penultimate layer, which would result in $c$ class-conditional Gaussians with $\mu_c \in \mathbb{R}^{Md}$ and $\Sigma \in \mathbb{R}^{Md \times Md}$. The additional covariance between features of different heads could improve detection of OOD samples, and distance averaging over the heads is unnecessary.
Chapter 7

Experiments

In this chapter, we evaluate the proposed model on two clinically relevant use cases in computational pathology. The multiple heads method is compared with a number of strong baselines, and we perform an ablation study of the input pre-processing technique as well as the Mahalanobis distance classifier. We further perform a diversity analysis of all methods on a histopathology task and compare two different methods for generating out-of-distribution proxy samples to be used for validation.

7.1 Datasets

We consider the two use cases described in Section 2.1 to evaluate the proposed model. For the sentinel lymph node case, we use the Camelyon16 [4] dataset. The Camelyon16 dataset contains 400 H&E stained whole slide images of human sentinel lymph node sections split into 270 slides with pixel-level annotations for the training set and the other 130 slides for testing. The slides were acquired at two different centers using a 40× objective (pixel resolution @ 0.243 µm). Since the images come from different centers, there are cross-lab variations present such as in staining intensities, the WSI scanner make with accompanying noise and blur patterns, and various physical and digital protocols. To ensure this does not obscure the experimental analysis, only images from the center with the most images are used, which come from the RadboudUMC center. Images from the Utrecht center are discarded for all data splits. The slides contain normal healthy tissue as well tumorous tissue. Note that tumor slides can contain between 20 to 150,000 tumor patches corresponding to tumor percentages ranging from 0.01% to 70% [5]. For the out-of-distribution detection task in lymph nodes, we use a set of 26 slides containing diffuse large B-cell lymphoma that were acquired and digitized in the RadboudUMC center; the same as the training set, such that deviations in the slide’s appearance stem solely from the differences in tissue. These slides were selected by a board-certified pathologist.

For the prostate case, we use a dataset from a cohort of 102 patients who underwent a radical prostatectomy at the RadboudUMC, for the task of epithelium segmentation. For each patient, a single slide is digitized based on the Gleason grades reported in the original pathologist’s report, resulting in a balanced group of prostate cancer stages. The epithelium annotations were acquired by unstaining the H&E slides, restaining with IHC, performing color deconvolution, and registration with the original slide, see [7] for details. For this use case, the out-of-distribution detection task consists of detecting foreign tissue, colon mucosa. We use a set of 27 slides of prostate tissue containing colon mucosa of various levels of presence. These slides were selected by the post-processing of board-certified pathologist’s reports, and visual verification by a junior pathology resident. See Table 3 for an overview of the data splits.
Table 3: Number of slides in each dataset/split for the Camelyon16 dataset (left) and the prostate dataset (right) and the corresponding out-of-distribution datasets.

<table>
<thead>
<tr>
<th>Dataset / split</th>
<th>Normal</th>
<th>Tumor</th>
<th>Total</th>
<th>Dataset / split</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camelyon16 / Train</td>
<td>101</td>
<td>65</td>
<td>166</td>
<td>Prostate / Train</td>
<td>50</td>
</tr>
<tr>
<td>Camelyon16 / Validation</td>
<td>26</td>
<td>19</td>
<td>45</td>
<td>Prostate / Validation</td>
<td>12</td>
</tr>
<tr>
<td>Camelyon16 / Test</td>
<td>53</td>
<td>26</td>
<td>79</td>
<td>Prostate / Test</td>
<td>40</td>
</tr>
<tr>
<td>B-cell lymphoma / OOD</td>
<td>-</td>
<td>-</td>
<td>26</td>
<td>Colon mucosa / OOD</td>
<td>27</td>
</tr>
</tbody>
</table>

7.2 Experimental setup

To evaluate the methods, we adapt two standard approaches to semantic segmentation for digital pathology.

7.2.1 Detection of metastases in sentinel lymph node

A conventional approach to semantic segmentation in computational pathology is patch-based classification, where a model is trained on patches of the whole slide image and the aggregate of these patch-based predictions serves as a slide-level representation. We use this approach for the segmentation of metastases in the lymph node sections, as it was the standard method in the Camelyon16 challenge. Specifically, a simple 16-layer CNN with leaky ReLU activations (slope = 1e−2), batch normalization (ε = 1e−5, momentum= 0.1) and no padding for the convolutions is trained with a softmax layer with two classes (tumor and normal). Due to the strongly imbalanced classes (see Section 7.1), a weighted cross entropy loss is used instead of balanced sampling; such oversampling of the underpresent class in preliminary experiments results in overprediction of the tumor class. For the first half of the training budget, the weights are set such that the classes have a balanced effect on the loss signal. In the second half, the weights are reset to equal values as to finetune predictions on the true training set distribution.

The input of the network is a downsampled 299 × 299 × 3 pixels image, and the output is an estimated probability over the 2 classes. Models are optimized using Adam (β1 = 0.9, β2 = 0.99, ε = 1e−8) with a batch size of 50, an initial learning rate of 3e−4 (halved after 10%, 20% and 50% of the epochs) and no weight decay. The models are trained for 50 epochs on 1.2 million extracted patches at a magnification of 10×, with center-pixel labeling. The patches are extracted at a ratio of 1 normal to 20 tumor patches from tumor slides, which is approximately the true data distribution. The last three convolution layers and the softmax layer are included in the heads for the M-Heads approach, after performing validation for {1, 2, 3, 5} layers on classification performance, which is supported by the insight from Section 3.1.1 that multiple layers provide more diversity, but too many layers may cripple training. For the MC dropout approach, we add a dropout layer (p = 0.5 as in [12]), after each convolution. Standard data augmentation techniques are used, i.e. random horizontal/vertical flips and color jittering in the range [max(0, 1 − δ), 1 + δ] in brightness (δ = 64/255), saturation (δ = 64/255), hue (δ = 0.04) and contrast (δ = 190/255), the same values as in [5]. As a baseline, we compute the entropy of the predictive distribution from a single model of the DE ensemble.

7.2.2 Detection of epithelium in prostate

The alternative approach to semantic segmentation is using the ubiquitous U-Net architecture [43] as the segmentation network for epithelium. We trained a six-level-deep U-Net on patches
with a size of $1024 \times 1024 \times 3$ (pixel resolution $0.48 \mu m$ and batch size of 1. We adapt the U-Net like structure using ResNet [44] blocks and skip-connections between the encoder and decoder. The transposed convolutions and max pooling layers are substituted by nearest neighbour upsampling layer to improve gradient flow. The extra residual connections combined with the nearest neighbour upsampling layer allow for a greater depth due to this improved gradient flow. The number of channels at the start of the network is set to 32, and is increased by a factor of 2 after each max-pooling layer. Further, we use leaky ReLU (same settings as before) and instance normalization. The outputs are again the predicted probabilities for the epithelium and the stroma classes. The same learning rate schedule and optimizer as in the previous section are employed for this task. To implement the M-Heads approach, we replace the last residual block and the softmax layer with $m$ identical blocks. Further, the softmax cross entropy loss is used and since it was found in [7] to be beneficial to account for small class regions, we use balanced class sample weighting: the loss of each pixel in the sample is weighted inversely to the surface of the class in that sample. Finally, weight decay or dropout is used in neither of the experiments.

### 7.2.3 Slide-level out-of-distribution measure

By computing the variance over the predictive distribution maps, we obtain a pixel-level predictive uncertainty map. Since we want to compare uncertainty measures from slide to slide, first an approximate foreground tissue mask is extracted with Otsu [45] thresholding such that non-tissue artefacts do not corrupt the measure. Then a spatial average pool is performed over the tissue-masked variance map, to acquire a slide-level out-of-distribution measure. This routine is performed in all cases for the in-distribution test slides and the out-of-distribution slides. Note, there are certainly more sophisticated methods that possibly perform better at pixel-level to slide-level conversion, such as a rule-based decisions relating to the histograms of the predictive entropy map or post-processing with conditional random fields, but that is beyond the scope of this work.

To evaluate the effectiveness of the methods to distinguish between in- and out-of-distribution images, the slide-level measures for in-distribution and OOD images are compared with the following metrics [39].

1. **TNR at 95% TPR** is the probability that a "negative" (OOD) example is correctly classified as negative when the true positive rate (TPR) is as high as 95%. True positive rate is computed as $TPR = TP/(TP + FN)$, where TP and FN denote respectively the true positives and false negatives. True negative rate (TNR) is computed as $TNR = TN/(TN + FP)$, where TN and FP denote respectively the true negatives and false positives. To obtain this metric, we compute the 95’th percentile of the predictive entropy of in-distribution slides and determine the ratio of OOD slides that are within this percentile threshold to the total number of OOD slides. Note: The baseline score for this metric is 5%, which is the case if in-distribution slides are compared with another set of in-distribution slides, and the best possible score is 100%. To be clear, for this metric threshold, the rejection rate of all test samples is 5%.

2. **AUROC** is the area under the receiver operating characteristic curve, which is a threshold-independent metric. The ROC curve represents the relationship between TPR and FPR, and the AUROC metric can be understood as the probability that a "positive" (in-distribution) sample is assigned a higher detection (=lower predictive entropy) score than a randomly chosen OOD sample. The best possible score for the AUROC metric is 100%.

3. **Detection accuracy** measures the OOD classification probability when TPR is 95%.
The definition is given by \(0.5(1 - TPR) + 0.5FPR\), where it is assumed that negative slides have an equal probability of appearing in the test set as the positive slides.

7.3 Out-of-distribution detection performance

7.3.1 Lymphoma in sentinel lymph node

We assess the performance of the multiple heads model on the sentinel lymph node task. Table 6 reports the performance for \(M = 8\). For the M-Heads method, the experiment is repeated with multiple settings for input pre-processing using FGSM as described in Chapter 5 to evaluate the additional performance benefits.

The results indicate that the vanilla M-Heads method performs consistently better than all comparison methods in terms of the TNR at TPR 95% metric. The baseline result shows that, although a single model has some discriminative power, only a small majority of the slides containing lymphoma were assigned a larger predictive entropy than the regular test slides, which supports the reported notion [15, 46, 47] that CNNs tend to make highly confident predictions while being woefully incorrect. In addition, regarding the ensemble-based comparison methods, the randomly initialized and data-shuffled Deep Ensembles method performs significantly better than MC dropout. Interestingly, the FGE method also outperforms MC dropout at no additional training cost.

Comparing to the vanilla M-Heads method, we see that adding small perturbations through FGSM as an additional preprocessing step during inference improves OOD detection performance by a large margin on this task. The perturbation indeed has a stronger effect on the in-distribution slides, which supports the hypothesis that the L1 norm of the gradient of the log softmax with respect to the input image tends to be greater for in-distribution images than OOD images.

Table 4: OOD detection performance on Lymphoma slides and Camelyon16 for M=8.

<table>
<thead>
<tr>
<th>Method</th>
<th>Input pre-processing</th>
<th>TNR at TPR 95%</th>
<th>AUROC</th>
<th>Detection accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>-</td>
<td>22.6</td>
<td>81.7</td>
<td>74.6</td>
</tr>
<tr>
<td>Deep ensembles</td>
<td>-</td>
<td>51.0</td>
<td>91.9</td>
<td>85.1</td>
</tr>
<tr>
<td>FGE</td>
<td>-</td>
<td>38.7</td>
<td>89.9</td>
<td>83.3</td>
</tr>
<tr>
<td>MC dropout</td>
<td>-</td>
<td>35.7</td>
<td>87.8</td>
<td>84.8</td>
</tr>
<tr>
<td>M-Heads</td>
<td>✓</td>
<td>69.2</td>
<td>93.7</td>
<td>90.9</td>
</tr>
</tbody>
</table>

Further, we provide the classification performance and CCA distances of the penultimate layer for each case on the test set of Camelyon16 in Table 5. The classification performance does not seem to correlate strongly with OOD detection performance, as also seen in Section 4.2 on CIFAR-10. For example, the FGE method outperforms all other methods in terms of AUC, but is worse than DE and M-Heads in terms of TNR at TPR 95% by a large margin.

For a visual inspection, we plot the results for three in-distribution slides in Figure 6 using the M-Heads method with input pre-processing. Red indicates a high variance of predictions, and green indicates a low variance. The maps show generally a very low variance for most of the tissue area, with the exception of a few isolated cells. In Figure 7, we show the uncertainty maps on three of the out-of-distribution slides that contain lymphoma. The maps
Table 5: C16 classification performance and corresponding CCA distances of the penultimate layer.

<table>
<thead>
<tr>
<th>Method</th>
<th>In-distribution ensemble AUC</th>
<th>Mean pairwise CCA distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>97.1</td>
<td>N/A</td>
</tr>
<tr>
<td>Deep ensembles</td>
<td>97.8</td>
<td>0.48±0.02</td>
</tr>
<tr>
<td>FGE</td>
<td>99.1</td>
<td>0.33±0.02</td>
</tr>
<tr>
<td>MC dropout</td>
<td>97.4</td>
<td>0.31±0.01</td>
</tr>
<tr>
<td>M-Heads</td>
<td>98.0</td>
<td>0.54±0.02</td>
</tr>
</tbody>
</table>

Figure 6: Predictive uncertainty maps on in-distribution slides of the sentinel lymph node task. The large images on the left show the complete slide, and the smaller square images show a close-up (indicated by the squares in the complete slide)

on the top-left and the bottom display high uncertainty estimates for a large portion of the tissue. A failure case is shown in the map on the top-right, where the captured uncertainty through predictive uncertainty is minimal although there is clearly a presence of lymphoma. It may be the case that a histological mimic is present in the Camelyon16 training dataset, which is known to contain numerous anomalies.

7.3.2 Colon mucosa in prostate gland

We evaluate our proposed model on the prostate task, where we use the best performing setting from the sentinel lymph node task. Table 6 reports the results. The out-of-distribution
detection performance for this task is significantly lower than for the sentinel lymph node task. A visual analysis is presented in Figure 8. Note that the close-up regions contain out-of-distribution tissue, and the remainder is in-distribution prostate gland tissue.

When we inspect the variance maps in detail, we observe that the colon mucosa has been detected in almost all cases. The reduced performance may be explained by the fact that the class distribution in this dataset is more balanced, which results in more present class boundaries between epithelium and stroma. It follows that there is much more aleatoric uncertainty naturally present, which complicates capturing the epistemic uncertainty of the distribution shift. We can see this clearly in the variance maps of Figure 8 with a generally increased variance throughout the tissue, when compared to sentinel lymph node predictions in Figure 8 where the classes are much more imbalanced. The higher uncertainty estimates for normal prostatic tissue hints at the importance of uncertainty decomposition as described in Section 2.2. Further, we note that due to the relatively minor area that they span, and the higher uncertainty for prostatic tissue, the uncertainty estimates for colon mucosa are diminished in the slide-level metric due to the basic method of spatial average pooling. Additional sophistication in the conversion of a pixel-level to a slide-level uncertainty metric can certainly be beneficial to the reported performance.

### 7.4 Effectiveness of distance-based classifier

To assess the effectiveness of detecting OOD samples with a distance-based classifier compared to using the predictive variance, we perform an experiment for the M-Heads method on the
Figure 8: Predictive uncertainty maps on slides of the prostate task. The zoom-in regions show a presence of colon mucosa. The other areas contain solely prostatic tissue.

Table 6: OOD performance on the prostate use cases for the M-Heads model trained on epithelium and stroma in prostate

<table>
<thead>
<tr>
<th>OOD</th>
<th>TNR at TPR 95%</th>
<th>AUC</th>
<th>Detection accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Colon mucosa</td>
<td>66.7</td>
<td>92.5</td>
<td>90.1</td>
</tr>
</tbody>
</table>

sentinel lymph node task by fitting the tumor and normal tissue classes by class-conditional Gaussians for each head, resulting in $M \times c$ distributions. The distributions are computed under Gaussian discriminant analysis as described in Section 6.2. The Mahalanobis score is used to obtain a distance of the penultimate layer features to the nearest class-conditional Gaussian. These distances are computed per head, and to finally obtain a single slide-level metric, the distances are average pooled following Algorithm 2. Minimum pooling was also attempted but did not yield better results on a validation set.

We find a TNR at TPR 95% score of 69.9 on the task of Section 7.3.1, compared to a score of 76.9 for the variance based method, a drop of 7 percentage points. This considerable drop in performance may be explained by the fact that the Gaussian prior is not expressive enough for this task. The problem of not capturing outliers due to a sample being far from the decision boundary seems to already be largely solved by using many heads and thus many decision boundaries. The probability that an OOD sample will end up on the same side of every decision boundary may simply be too small to affect performance in the case of predictive entropy, such that the Gaussian prior limitation cripples inference more than it is improved by the distance-based classification.
Upon further inspection we observe many anomalies in the Camelyon16 dataset, so it can be considered a dataset with a high level of label noise. The LDA method expects perfect training samples, so label noise may complicate the density estimation. Considering that the maximum likelihood estimate of the empirical covariance is known to be very sensitive to the presence of outliers, the corresponding Mahalanobis distance is as well.

Table 7: Performance using different validation techniques

<table>
<thead>
<tr>
<th>In-distribution</th>
<th>OOD</th>
<th>Validation on adversarial samples</th>
<th>Validation on hard negative samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sentinel lymph node</td>
<td>Lymphomas</td>
<td>TNR at TPR 95% 94.5 92.3</td>
<td>Detection acc. 66.4 89.8 85.0</td>
</tr>
<tr>
<td>Prostate</td>
<td>Colon mucosa</td>
<td>66.7 92.5 90.1</td>
<td>47.2 84.1 75.8</td>
</tr>
</tbody>
</table>

7.5 Comparison of validation methods for FGSM

We compare the methods of validation for tuning the perturbation magnitude $\epsilon$ for input preprocessing using FGSM. As was shown in [39], the robustness of FGSM to this parameter has severe limitations. Additionally, tuning a hyperparameter to detect OOD samples without actually assuming the availability of OOD samples is not trivial; it is therefore essential to consider the possible methods of validating this parameter extensively. The methods, explained in detail in Section 5.3, are validation on adversarial samples (i.e. samples that have been preprocessed with a "negative" FGSM step), and validation on hard negative samples (i.e. samples that were often misclassified during training). Both methods are evaluated for each of the three previous experimental setup and an overview of the results can be seen in Table 7. For both cases, the validation using adversarial samples is superior to using hard negative samples. This may be explained by the the fact that hard negatives are related to aleatoric uncertainty, since hard negatives can contain class boundaries or class overlap. The drop in performance is more severe for the colon mucosa task, which concurs with the observation that more aleatoric uncertainty is naturally present in the label space of the prostate gland task due to a higher number of class boundaries.
Chapter 8

Limitations & Future Work

The results in the previous section are comparable with those of state-of-the-art in this field in their standardized experiments [46]. Note that the tasks we considered deal with out-of-distribution data on the same manifold which is much harder to detect when comparing to the standardized experiments with data from different manifolds, e.g. CIFAR-10 versus the Street View House Numbers dataset [48].

There are a few limitations to the current method. A downside to using the M-Heads model is that, although only a single feedforward pass is needed to obtain uncertainty estimate compared to the comparison methods, the number of samples is decided a priori during training. The current method of conversion from a pixel-level uncertainty map to a slide-level uncertainty scalar is arguably too primitive to represent important peaks in predictive uncertainty, as we have seen in the prostate gland task. There are many avenues of possible research for adding sophistication there, from hand-engineered rule-based methods to training an additional layer on top (using adversarially generated validation samples as an OOD proxy). Further visual interpretation of the results by a pathologist can be beneficial there.

Importantly, we have seen that the variance of the predictive distribution contains aleatoric uncertainty related to class boundaries, which is undesired for the task. A more principled decomposition of the types of uncertainty with the goal of obtaining higher quality epistemic uncertainty estimates is a worthwhile avenue of future research.

The considered LDA method is highly sensitive to label noise, which is often present in histopathology data due to the inherent inhomogeneity of microscopic tissue. It may be worthwhile to consider alternative density estimates that are more robust, such as the Minimum Covariant Determinant estimator [49].

Further, we observed that neural networks have larger gradient norms of log softmax scores when applied on in-distribution images as compared to OOD images. We hypothesize that it may be related to the interpretation of neural networks as performing local generalization, with in-distribution samples having a smaller distance to the training samples and a push towards the target class may push the sample to an existing peak in label space related to a dense cluster of training samples. A better understanding of this phenomenon can lead to further insights in this problem.

Finally, introducing explicit ways of specialization in the M-Heads training procedure could result in more diverse samples, and thus as we have seen a better detection of OOD samples. An explicit diversification term in the loss function is plausible. Recently, a method of differentiable CCA [50] was proposed, which can potentially be used during training to increase the CCA distances of the heads. Additionally, the expressiveness of non-linear CCA approaches [51] can assist in providing insight in the representational diversity of ensemble-based uncertainty estimation methods.
Chapter 9

Conclusions

During this thesis, we casted the problem of anomaly detection in computational pathology as an epistemic uncertainty estimation problem. Due to the difficulty of obtaining a multi-modal predictive distribution, we studied a multiple heads topology in convolutional neural networks to train an ensemble to yield highly diverse posterior distributions. We then related representational diversity to canonical correlation analysis. As our experiments showed, we can use a derived distance measure to empirically prove that the presented model exhibits a greater representational diversity than currently popular uncertainty estimation methods, such as MC dropout and Deep Ensembles. Subsequently, we investigated using the fast gradient sign method to separate the softmax scores of in-distribution and out-of-distribution samples, by exploiting the phenomenon that the distribution of the L1 norm of the gradient of the log-softmax with respect to an input image is larger for in-distribution inputs. We performed an extensive experimental validation of the multiple heads method on two clinically relevant use cases in histopathology, and showed that the model is competitive with the comparison methods while being more efficient during training as well as during inference.

Our results do not yield a method that could be directly deployed for histopathology tasks in a clinical setting. However, we hope that the presented work might still provide insights towards better understanding of the challenges of out-of-distribution detection in the field of computational pathology and the importance of sample diversity and maybe lead to useful epistemic predictive uncertainty estimation methods in the future.


