Uncertainty Quantification in Neural Network-Based Classification Models

by

Mohammad Hadi Amiri

A thesis proposal submitted to University Of Ottawa in partial fulfillment of the requirements for the degree of

Master of Electrical and Computer Engineering

in

School of Electrical Engineering and Computer Science

University Of Ottawa

Ottawa, Ontario

Abstract

Probabilistic behavior in perceiving the environment and take critical decisions have an inevitable role in human life. A decision is concerned with a choice among the available alternatives and is always subject to unknown elements concerning the future. The lack of complete data, insufficient scientific, behavioral, and industry development and of course defects in measurement methods, affect the reliability of an action's outcome. Thus, having a proper estimation of this reliability or uncertainty could be very advantageous particularly when an individual or generally a subject is faced with a high risk. With the fact that there are always uncertainty elements whose values are unknown and these enter into a processes through multiple sources, it has been a primary challenge to design an efficient representation of confidence objectively. With the aim of addressing this problem, a variety of researches have been conducted to introduce frameworks in metrology of uncertainty quantification that are comprehensive enough and have transferability into different areas. Moreover, it's also a challenging task to define a proper index that reflects more aspects of the problem and measurement process.

With significant advances in Artificial Intelligence in the past decade, one of the key elements, in order to ease human life by giving more control to machines, is to heed the uncertainty estimation for a prediction.

With a focus on measurement aspects, this thesis attends to demonstrate how a different measurement index affects the quality of evaluated predictive uncertainty of neural networks. Finally, we propose a novel index that shows uncertainty values with the same or higher quality than existing methods which emphasizes the benefits of having a proper measurement index in managing the risk of the outcome from a classification model.

Keywords : uncertainty quantification ,neural network, classification , deep learning

Acknowledgements

I would like to express my deepest appreciation to my supervisors, Prof. Hussein Al Osman and Prof. Shervin Shirmohammadi for giving me the unique opportunity to work in the discover lab. It was a huge honor to conduct research and learn under the guidance of such smart, aspirational, and personable supervisors. I appreciate your unwavering patience and unmatched professionalism. I would also like to thank my family specially my sister Maryam and her husband Ashkan whose efforts provided me with this priceless chance.

Contents

Acronyms

ABNN Approximate Bayesian Neural Networks. 25

AI Artificial Intelligence. 2

ANN Artificial Neural Network. 14

BNN Bayesian Neural Network. 24

CDF Cumulative Distribution Function. 56

CEC constant error carousel. 20

CNN Convolutional Neural Networks. vi, 15

DL Deep Learning. 2

ELBO Evidence Lower Bound. 26

ENN Ensemble of Neural Networks. 25

FNA Fine Needle Aspirate. 54

GUM Guide on the Expression of Uncertainty in Measurement. 7

LSTM Long short-term Memory. 20

ML Machine Learning. 2

MPL Multi Layer Perceptron. 14

MSE Mean Square Error. 28

NN Neural Network. 2

VI Variational Inference. 25

VIM International Vocabulary of Basic and General Terms in Metrology. 7

WD Wasserstein Distance. 56

List of Tables

List of Figures

Chapter 1

Introduction

"False confidence bred from an ignorance of the probabilistic nature of the world, from a desire to see black and white where we should rightly see gray" — Immanuel Kant

We live in an uncertain world, these days more than any other. Much of what lies ahead in life is still unpredictable, whether it is related to a worldwide epidemic, the economy, or our finances, health, and relationships. However, we humans yearn for security. We desire to feel secure and in charge of our lives and well-being. Risk and uncertainty are related concepts that are frequently used interchangeably. Risk can refer to both the danger that uncertainty poses and the reaction to that danger[1]. We frequently deploy our awareness of uncertainty implicitly to steer clear of severe damage. To leverage this awareness, in scientific fields, we seek techniques that can quantify and express the uncertainty of an experiment or a measurement. However, we face the challenge that Different quantification strategies are required for various forms of uncertainty [2]. Moreover, "uncertainty" may refer to different phenomena in various contexts. As a result, researchers have proposed operational definitions of uncertainty that consider how truth, uncertainty, and failure are perceived differently.[3]

Nowadays, Artificial Intelligence (AI) based models, particularly Deep Learning (DL) techniques, are increasingly being deployed to solve various problems in diverse fields . Disease detection and autonomous vehicles are two examples of such fields that have drawn the greatest attention. The risk associated with incorrect predictions produced by such models highlights the need for uncertainty quantification [4]that allows us to make informed decisions regarding the reliability of model outputs for safety-sensitive applications.

In the Machine Learning (ML) and Neural Network (NN) disciplines, there are two general types of modeling, namely regression, and classification. Hence, we need to devise different strategies of uncertainty evaluation for each type. Although several methods for uncertainty quantification for regression models have been proposed, has been little progress in uncertainty estimation for classification models[5]. Not only in ML, but even in the broader field of metrology, it has been a challenge to define, evaluate, and express the uncertainty in the examination of categorical properties [6].

1.1 Motivation

AI and DL techniques are being relied on to make decisions. In classification models, these models resolve the class the inputs belong to. Examples of such classes include disease category, object type, human emotion, etc.
In low-risk DL models, such as the ones employed for gaming or text-to-speech models, a wrong prediction does not typically present severe negative consequences.

However, for certain high-risk applications, such as disease detection or autonomous vehicles, incorrect predictions can result in considerable harm. Seven cases of driver disengagements in 2016 for google cars would lead to contact with other objects, or the fatal accident which happened by an uber vehicle that killed a woman in California in 2018. Another example would be AI used by law enforcement to do various tasks such as criminal patterns detection, face detection, etc which again can be affected by failures due to system errors.

Such important challenges are the main motivation for the efforts to resolve issues of AI reliability where uncertainty evaluation plays a major role.

1.2 Problem Statement

This thesis proposes a new index(measure) to quantify predictive uncertainty in NN-based classification models. Since the output of such models contains a confidence score for all present classes (instead of a single value), the problem belongs to the measurement of categorical variables from a metrology perspective and there could be different forms of distributions such as discrete, binomial and Poisson [2] instead of a normal distribution which plays the center role for uncertainty quantification in regression tasks. Hence, we need to investigate related frameworks, concepts, and conditions in metrology that affect the outcome and factors that are involved. So our first goal is to identify the framework that corresponds more to the specifications of our mentioned problem.

Moreover, we need to downscale the coarse elements to the NN classification structure in order to have better adaption and performance. we need to implement one of the existing methods to provide a distribution of the output of a NN classification

model. In fact, we are aiming to enable a model to produce a probability distribution for all classes at inference time. Finally, we seek a set of mathematical equations which satisfy all technical conditions and also produce comparable values.

We finally look for reliable assessment benchmarks that are used to compare indexes. So far there are just a few tasks designed that are able to relatively illustrate the quality of output from different uncertainty quantification methods.

1.3 Thesis Statement and Contributions

We propose a new index for uncertainty evaluation in neural network-based classification models. We do so By applying adjustments to a general framework in metrology and adopting attributes to quantitative factors in the neural network pipeline . In order to have a reliable assessment of our proposed index, we also needed to choose datasets with certain characteristics based on categorizations in metrology (ex. nominal, ordinal and binary) and different accuracy rates. We considered more realistic datasets, especially in healthcare and clinical cases.

We totally deployed 5 datasets with various data formats including video recordings, image, and EED signals, and retrained models applying modifications required for probabilistic inference.

1.4 Thesis Structure

The rest of the thesis is organized as follows. In chapter 2 we discuss the background and concepts that are pertinent to our solution. We present fundamental ideas and definitions of categorical properties in metrology with a focus on uncertainty evaluation. We overview the structure of NN and probabilistic deep learning approaches. Moreover,

We review various uncertainty sources.

In Chapter 3, we present our literature review. Where we review the metrology frameworks proposed in [6] and [7]. We discuss NN probabilistic networks and methods to estimate NN probabilistic output. Furthermore, we review existing indices of uncertainty.

In chapter 4, we introduce our proposed method. We will explain the difference in our interpretation of uncertainty.

In chapter 5, we describe our datasets and present the results of our evaluations.

In chapter 6, we present our conclusions and describe our limitations. We also provide ideas for future work on the topic.

Chapter 2

Background

In this chapter, we intend to review the main principles that must be understood before we can build a solution to the problem of predictive uncertainty evaluation. We start from related topics in categorical data analysis and study principles of uncertainty quantification. By exploring the idea of ML and NNs architecture to gain the related knowledge in the second part, we aim to discuss the workflow of probabilistic modeling and prediction in the third part to merge the information in the first two parts to show how uncertainty in classification models is quantified.

2.1 Uncertainty and Measurement Methods

Evaluation Uncertainty in its generic meaning is doubt and lack of sureness about the result of an experiment. This results from having limited knowledge about the data or measurement procedure making it difficult to control, plan, or predict a future outcome. In measurement, EURACHEM[8] defines uncertainty as "a parameter associated with the result of a measurement, that characterizes the dispersion of the values that could reasonably be attributed to the measurand". This definition is in the

line with the Guide on the Expression of Uncertainty in Measurement (GUM) [9] and the International Vocabulary of Basic and General Terms in Metrology (VIM)[10].

The major advantage of having proper information about a result's uncertainty is to prevent irreparable damages as a measurement cannot be interpreted properly without knowledge of its uncertainty [11].

2.1.1 Uncertainty Components and layers

There are ten elements considered in GUM [9] that can cause uncertainty or affect its value and quality in measurement from which we mention those that are linked to our problem :

a) incomplete definition of the measurand;

b) imperfect realization of the definition of the measurand;

c) non-representative sampling — the sample measured may not represent the defined measurand;

d) inadequate knowledge of the effects of environmental conditions on the measurement or imperfect measurement of environmental conditions;

e) inexact values of measurement standards and reference materials;

f) approximations and assumptions incorporated in the measurement method and procedure;

g) variations in repeated observations of the measurand under apparently identical conditions ;

In order to define any index for uncertainty quantification, we need to take into account not only the factors above but there are layers that may be considered so as to have an acceptable index in terms of mathematics and conceptual specifications. Luca. Mari et al.[6] mention three cases as basics to designing a new framework:

"L1. a generic concept: uncertainty as a quantifiable attribute;

L2. a mathematical concept: uncertainty as a generic quantitative attribute, corresponding to a yet unspecified index of a distribution;

L3. several specific mathematical concepts: uncertainty as a given quantitative attribute, related to a specified index of a distribution."

2.1.2 Types of Uncertainty

Although there are numerous potential causes of uncertainty, it is useful to classify uncertainties as either aleatory or epistemic in the context of modeling. From a practical perspective, categorizing the uncertainty within an experiment is helpful since it makes it evident which uncertainties have the potential to be decreased [12]. Both sorts of uncertainties are present in the majority of engineering problems and it has been argued that the two types can be distinguished clearly (e.g., [13]).

Aleatoric uncertainty is assumed to be the inherent randomness of a process (e.g. Random noise in data). Hence, it is either impossible or very complex and expensive to reduce such uncertainty.

Epistemic or systematic uncertainty is caused by a lack of knowledge or shortage of data. In fact, if alternative measurement techniques might be considered, this measurement uncertainty is classified as epistemic.

Let us consider the measurement of the flexibility of alloy used in an aircraft's body as an example. If a sample obtained from the aircraft can be tested, providing information about the flexibility, then the uncertainty should be classified as epistemic if the desired flexibility is that of the alloy in an existent aircraft, Random measurement errors may, of course, occur throughout the testing. On the other hand, if no attempts

are made to create more precise modeling related to, for example, the control of the alloy manufacturing, the uncertainty in the flexibility of alloy in a future aircraft should be classified as aleatory. No amount of testing will be able to eliminate the inherent variability in the alloy flexibility of future aircraft until the aircraft has been realized. According to [14], the character of the aleatory uncertainty "transforms" into epistemic uncertainty as the manufacturing is realized.

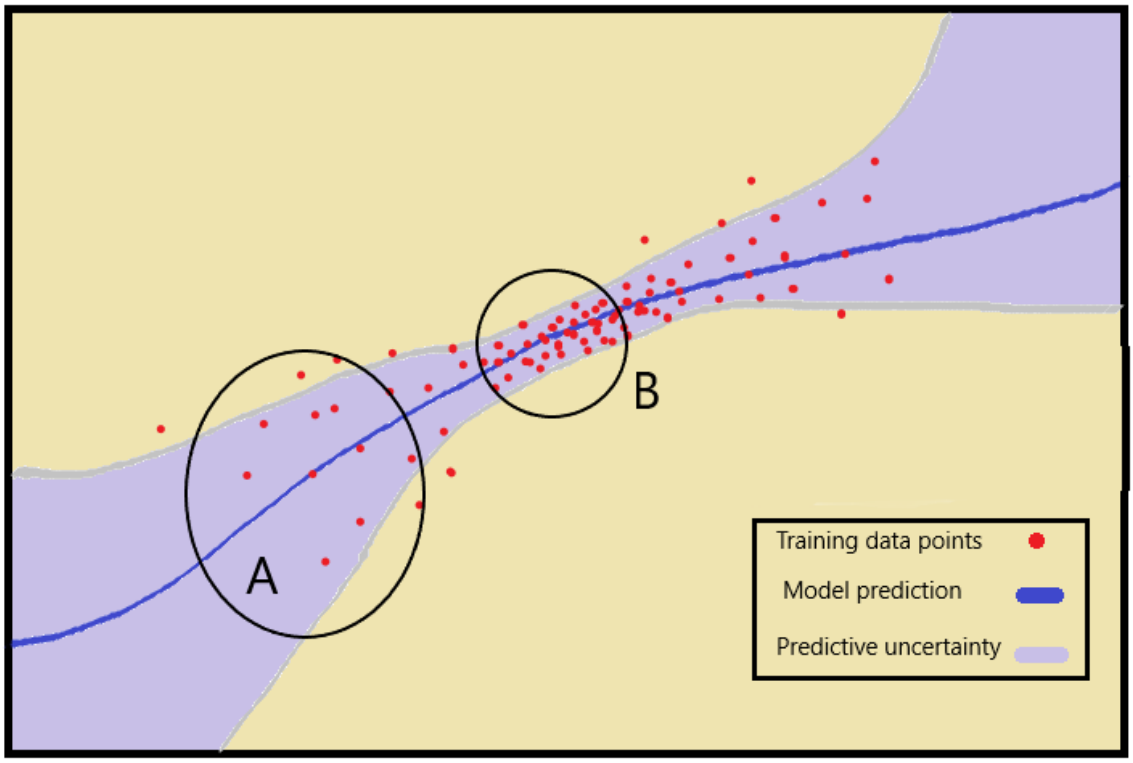


Figure 2.1: A simple regression model that fits the data points(blue line) . The Gray area demonstrates the predictive uncertainty. (A) shows the region with relatively high data sparsity and consequently high aleatoric uncertainty and also lack of enough data points caused high epistemic uncertainty. region (B) shows less sparse data, therefore less aleatoric uncertainty and more data points lead to more confidence in prediction which shows less epistemic uncertainty as well.

2.2 Categorical Properties Analysis

Our specific subject in classification needs a clear understanding of the differences in a coarser level for uncertainty quantification of categorical variables (also known as qualitative or non-quantitative variables) and its properties. A categorical variable, is a statistical variable that, based on some qualitative characteristics, assigns each human or other unit of observation to one of a small, and typically fixed, a number of possible groups or nominal categories.

Stevens [15] proposed four measurement scales from which two of them are the subsets of categorical scale, namely nominal and ordinal:

Nominal variables are labels or names that are given to objects Objects with the same label belong to a common category. Only equality and inequality comparisons between variable values are possible. "Less than" and "greater than" relations, and operations like addition and subtraction, do not exist between them. Nominal variables include, for instance: religion , race, political party etc.

Ordinal variables are the numbers or codes given to objects that correspond to the measured entities rank order, such as first, second, third, etc.; large, medium, small; or alphabetical order, and so forth. Distance between two generic levels on the same scale has no real relevance. In addition to equal/unequal comparisons, greater/less than comparisons are also possible. However, basic operations like addition and subtraction still have no significance. Ordinal variables include, for example, the severity of illness measure[16], quality orders, and so on.

Binary classification outputs variables that can be categorized as ordinal or nominal depending on the characteristics described above.

2.2.1 uncertainty quantification

So far, there is no consensus on a single framework to represent dispersion and uncertainty for categorical variables. It is in fact rooted in the conceptual aspects through different types of measurements which are classified in one way into "quantitative" and "non-quantitative" groups that lead to practical discrepancies and commonalities and therefore it can have a considerable impact on measurement methods . We would mention one common aspect and one difference in form of an example similar to [17]:

Assume measuring the weight of an object and labeling its color; The value v:= 2.54 kg implicitly contains information about the reference set V from which 2.54 is selected, with V being the set containing the multiples of the kilograms that may or may not be integers. As a result, v:= 2.54 kg actually means v:= 2.54 kg in V, where V is typically left as an implicit definition. However, the entity v:= Red does not contain information about the reference set V to which v belongs. For example:

V1:= "red, other" or V2:= "red, blue, green, black, other," where v:= red in V2 is more informative than v:= read in V1. Because of this, the information v:= red is insufficient (in the functional formalization of P, this is evident: v:= red is insufficient to determine the range of the function P, i.e., the set of its potential values, leaving P as an undefinable function). To mention the similarity, The statements "weight X a = 2.54 kg" and "color X a = red (in V)" both claim that the X has a weight that is equal within the bounds of the accepted experimental approximation to the length that is produced by multiplying 2.54 by the weight that is typically defined as the kilogram, and it has a color that is equal within the bounds of the accepted experimental approximation to the color identified as red in the given set of colors V .

A measurement should typically convey information on both the standard uncertainty, U_v, as well as the estimated value, v, according to GUM[9], which is focused

on the treatment of quantities, refers to as the "measured quantity value" [10]. GUM demonstrates that, given a set of circumstances, v may be used to represent the mean value of a probability distribution and U_v can be used to represent the standard deviation of the mean (in case of quantitative variable) [18]. however, The parametric distribution-based GUM paradigm does not apply to categorical property evaluations (for example averaging two colors like red and blue is meaningless).

Although a distribution of the categorical variables is non-parametric and has undefined intervals, it can still be used to derive a pair of values that play the roles of a location and scale parameter. The idea of location is not specified for a categorical property, and the mean value of a distribution like f cannot be calculated. On the other hand, given that $f(red) = 0.8$ is the highest probability in the distribution, the mode of the distribution may be thought of as its most reasonable value, such as $v = red$ in the above example. It must be noted that a multi-modal distribution produced via categorical assessments would not include a single representative location, which is one issue with this decision. Since categorical characteristics are algebraically weaker than quantities, it is practically inevitable that some of the mathematically sophisticated tools of quantitative treatment will be lost.

One of the possible indexes of a scale parameter for the distribution f is Shannon entropy :

$$H(f) = -\sum_{v} f(v) \log v(v) \tag{2.1}$$

More proposed indices are reviewed in chapter 3. Additional information on the treatment of uncertainty in categorical property in Possolo [19].

2.3 Machine Learning and Artificial Neural Networks

2.3.1 Machine Learning

With the use of ML, which is a form of AI, software programs can predict outcomes more accurately without having to be explicitly programmed to. In order to forecast new output values, machine learning algorithms use historical data as input. Technically ML methods consist of two parts based on their task and output format namely regression and classification.

Regression modeling Approximates a mapping function (f) from input variables (X) to a continuous output variable (y). A regression can have continuous-valued or discrete input variables. For instance, the prediction of house price could be in the range of $500,000 - 800,000$.

On the other hand, classification modeling includes approximating a mapping function (f) from input variables (X) to discrete output variables (y) and labels or categories are the output variables. For a given observation, the mapping function predicts the class or category. Input needs to be categorized into one of two (binary classification) or more classes(multi-label classification) in order to solve a problem.

A continuous number is typically predicted by classification models as the likelihood that a given example will belong to each output class. The probabilities can be thought of as a measure of how likely or certain we are that a given example belongs to a particular class. The predicted class is chosen as the label with the highest probability value. Well-known example of a classification of dogs and cats or spam email detection.

2.3.2 Artificial Neural Networks

Artificial Neural Network (ANN) are computing systems that are naturally inspired by biological NNs that are the constructive unit of human's brain. Their forms and complexity have been evolved based on brain plausibility or simply through direct experiments and what tends to work out.

2.3.2.1 Perceptron

The smallest computational unit of an ANN, the perceptron was created to mathematically simulate the neuron, the basic building block of the brain and nervous system. Scientists were inspired to do binary classification using a model that matched the behavior of the neuron, which only sends nerve signals passing a particular threshold to the subsequent cells. More accurately, a perceptron calculates a weighted sum of the inputs, sends the result to a non-linear activation function that serves as a trigger, and uses the result as the output or $o = f\left(\sum_i w_i x_i\right)$. Figure 2.2 illustrates how real neurons acted as an inspiration for the perceptron model. Perceptron was able to imitate basic classifiers and logical operations.

2.3.2.2 Multi Layer Perceptron (MPL)

MLP was introduced as an organized block of perceptrons as a result of improvements in computing power and the advent of more activation functions. The idea of DL became popular due to designs with hidden layers, or any number of intermediary layers between the input and output layer. These models were able to simultaneously extract features and classify data. This feature makes it simpler to apply DL models to novel challenges, but a larger black box may cause questions about the model's interpretability. The backpropagation algorithm, which updates weights proportionally

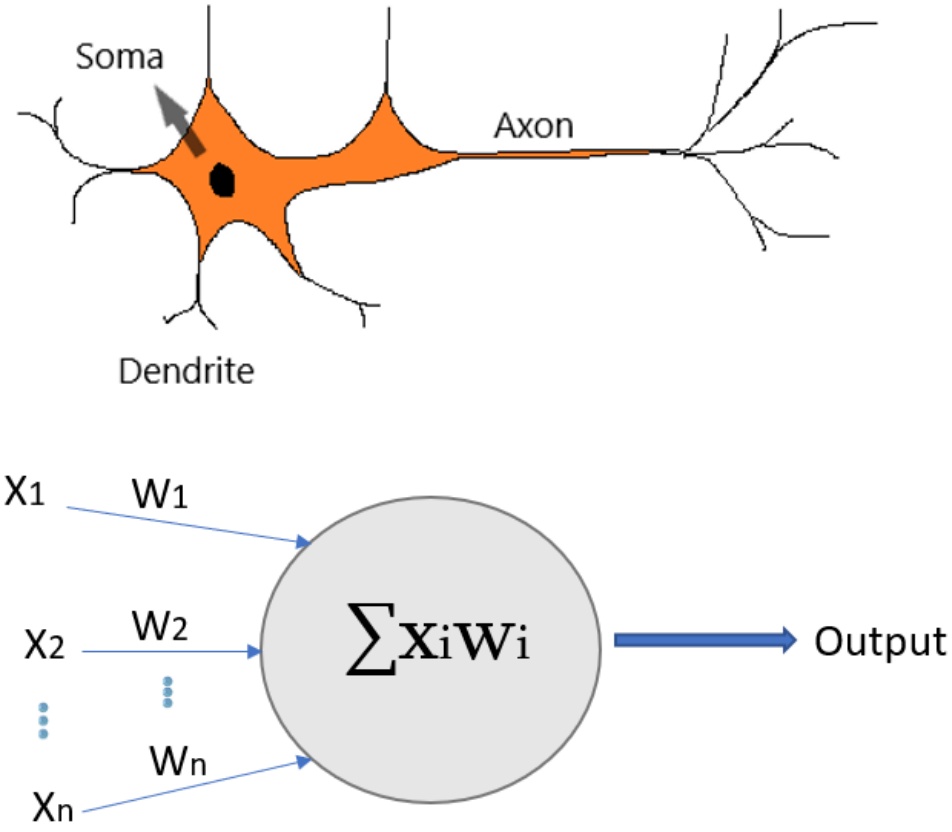


Figure 2.2: a neuron's resemblance to a perceptron

to the derivative of the error with respect to each weight determined using the chain rule, is used in the training of MLPs.

2.3.3 CNN

This category of ANNs was established after showing a promising classification performance on ImageNet [20] dataset. Since AlexNet [21] which was the architecture

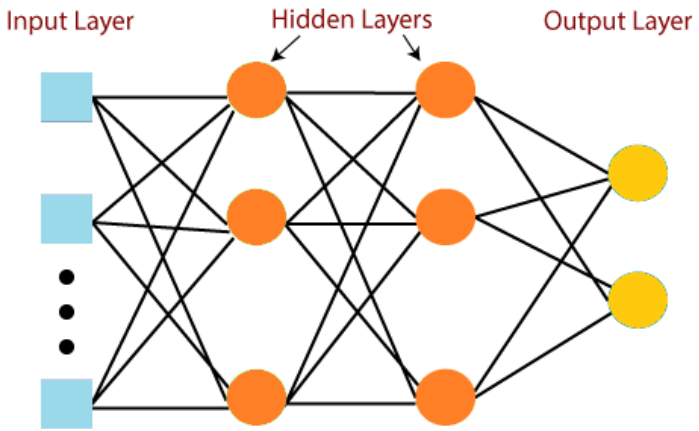


Figure 2.3: Architecture for a simple MLP with two hidden layers.

that outperformed non-Deep Learning methods for the first time in 2012, CNNs have evolved every year. However, the components of each architecture have been almost the same and, we want to introduce those that were the most important to us.

This class of ANNs was created after demonstrating good classification performance on the ImageNet dataset [20]. CNN's have evolved every year since AlexNet [21], which was the architecture that outscored non-Deep Learning techniques for the first time in 2012. However, the elements of each architecture have been nearly identical, and we want to highlight those that were most significant to us.

2.3.3.1 Convolution Layers

MLPs are not capable of tracking spatial dependencies efficiently. On the other hand, convolution layers have an intuitive way of connecting learnable parameters together called parameter sharing that makes them more capable of learning local correlations compared to fully connected layers. The weights of these layers are grouped as multi-dimensional kernels (filters) that move across the input computing a weighted sum of what they see. These kernels can represent image processing crafted filters such as

horizontal or vertical edge detection, more flexibly. In Figure 2.4, you can find how the output of a convolution kernel should be calculated.

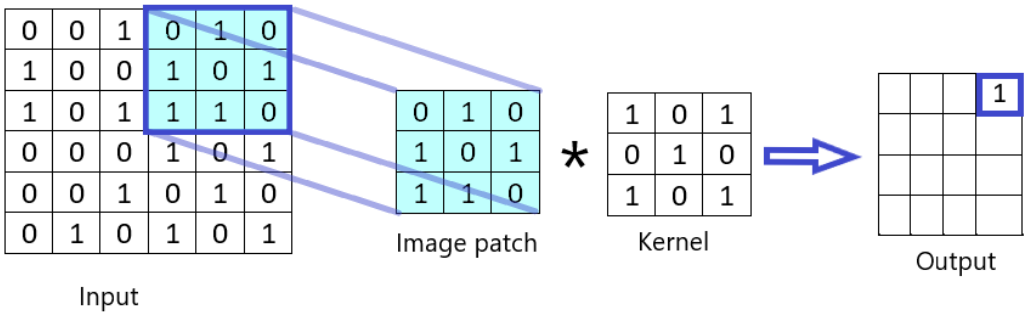


Figure 2.4: The process of calculating a convolution.

Several hyperparameters in a convolution layer should be fixed throughout the design phase, including:

- The shape of the kernel that we show by k.
- The number of pixels we use for padding is shown by p. This parameter helps us avoid the outputs with too small shapes due to multiple convolutions.
- The stride of the convolution shown by s, controlling the step size the kernel takes during the convolution.
- The number of kernels shown by d determines the depth (number of channels) of the output.

Consequently, the shape of the output shown by o follows the equation below for the input with the shape of i.

$$o = \frac{i - k + 2p}{s} + 1 \tag{2.2}$$

2.3.3.2 Activation Functions

An Activation Function determines whether or not a neuron should be activated. Moreover, it provides neural networks with nonlinear expression ability, allowing them to better fit the data and enhance accuracy. The obtained values are mapped between 0 and 1 or -1 and 1, etc (depending upon the function). There are two types of activation functions namely linear and nonlinear functions. nonlinear activation functions are the most used ones since their components add nonlinearity to the CNN architecture. Among existing functions like Sigmoid, hyperbolic tangent, or Tanh and ReLU, we would use Relu variants, which are frequently used in CNNs, enabling us to stack additional layers and create deeper structures capable of resolving more challenging optimization issues. Utilizing ReLU requires less computation compared to other methods, and since its derivative does not tend to zero, it prevents the vanishing problem (The value of the product of the derivative falls as the number of layers in the network increases until, at some point, the partial derivative of the loss function approaches a value close to zero and vanishes). Due to its significance in back-propagation, the derivative of an activation function is very noteworthy. By contrasting the behavior of ReLU's derivative with that of other well-known activation functions like Sigmoid and Tanh, where the derivative tends to zero when we are located on the extremums of each function and affects the submodule's capacity for learning, we can determine the advantage of ReLU. To add more flexibility there are other variations of ReLU such as Elu and LeakyReLU.

2.3.3.3 Fully-connected Layers

fully connected layers are created by arranging several perceptrons in a layering pattern and logically connecting each perceptron in turn. With regard to feature extraction and

supervised learning of categorical data, fully connected layers demonstrated excellent performance. They are not resistant to geometrical modifications like rotation, scaling, and shifting, nor are they sensitive enough to spatial properties. Since we focus more on classification tasks than feature extraction in the final layers of CNN architectures, they are frequently employed.

2.3.4 Recurrent Neural Networks

Recurrence adds memory to the ANN. Recurrent Neural Networks (RNNs) were introduced to learn temporal dependencies and process sequential data more efficiently. For instance, CNN considers each sample independent from the others but such an assumption will not be valid for movie frames, raining forecasts, or stock market analyses. The most trivial group of RNNs is formed by adding a link from the output of a node to its input and making the next output of the model-dependent on every input fed to it according to equation 2.3.

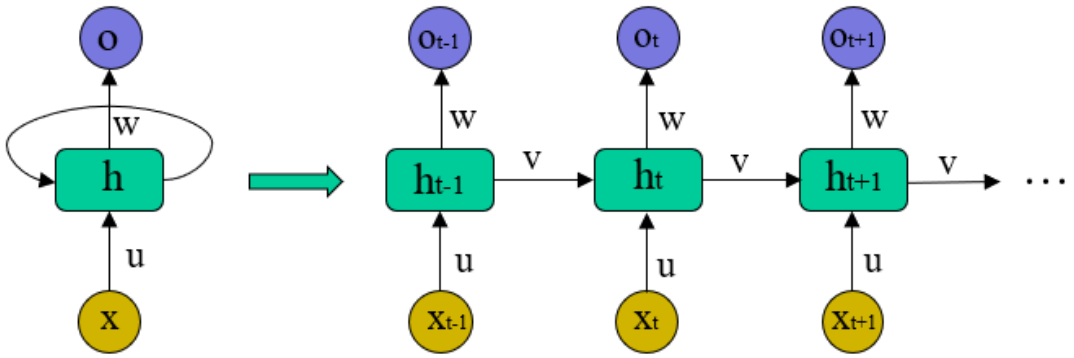


Figure 2.5: RNN node encodes temporal dependencies that are revealed by unfolding the node.

$$s_t = \sigma\left(Ux_t + Ws_{t-1}\right) \tag{2.3a}$$

$$h_t = softmax\left(Vs_t\right) \tag{2.3b}$$

When utilizing the backpropagation approach to train RNNs, we would discover an exponential dependence on the derivative of the activation function because the extended model requires us to multiply it by each time point. As a result, the training would even lag behind due to gradient explosion or gradient vanishing.

2.3.4.1 LSTM

Long short-term Memory (LSTM), [22], is a more sophisticated type of RNN that can learn any kind of long-term dependencies by using a feedback loop called constant error carousel (CEC), which lowers the likelihood of gradient disappearing. The forget gate determines which information is important and which may be disregarded. The sigmoid function is used to process data from the current input $X(t)$ and the hidden state $h(t-1)$. Among all outputs, it picks those with values closer to 1 (in the range of $0-1$). Equations 2.4 mathematically describe the process occurring inside an LSTM cell, where $\odot$ stands for element-wise matrix multiplication. Figure 2.6 illustrates this process.

$$z_t \;=\; tanh\left(W_z\left[h_{t-1},\; x_t\right]\right) \tag{2.4a}$$

$$i_t,\; f_t,\; o_t \;=\; \sigma\left(W_{i,f,o}\left[h_{t-1},\; x_t\right]\right) \tag{2.4b}$$

$$s_t \;=\; z_t \odot i_t \;+\; s_{t-1} \odot f_t \tag{2.4c}$$

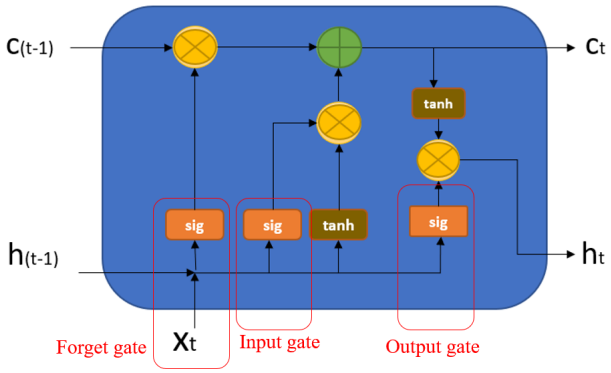


Figure 2.6: Input gate, output gate, forget gate, cell state, and their interconnection in an LSTM cell.

$$h_t = tanh\,(s_t) \odot o_t \tag{2.4d}$$

2.3.4.2 Conv-LSTM

Conv-LSTM, which encodes spatial properties present in movies or other dependent series of images, combines the learning abilities of RNNs with CNNs. The same restriction applies to processing such data using LSTM units when feeding images to fully connected networks. The model won't be able to maintain track of spatial relationships due to the redundant manner the nodes are connected. As a result, we can create Conv-LSTM layers by simply substituting convolutional kernels for dense trainable weight $W_{i,f,o,z}$ [23] . It's important to note that the widely used equation 2.5 contains a simplified version of the cell proposed in conv-lstm.

$$z_t = tanh\,(W_z * [h_{t-1},\, x_t]) \tag{2.5a}$$

$$i_t,\ f_t,\ o_t\ =\ \sigma\left(W_{i,f,o} * [h_{t-1},\ x_t]\right) \tag{2.5b}$$

$$s_t\ =\ z_t \odot i_t\ +\ s_{t-1} \odot f_t \tag{2.5c}$$

$$h_t\ =\ tanh\left(s_t\right) \odot o_t \tag{2.5d}$$

2.3.4.3 Over fitting

The main strength of NNs is actually the high flexibility of having numerous model parameters whose values may be learned from data via gradient-based optimization. Because they are good at estimating functions when there is a large amount of data available. However, this flexibility leads to weakness at inference time when the model can't accurately execute against unobserved data. In fact, the model fits the training data too much 2.7. This issue is called overfitting.

This problem is addressed by research-based of probability theory and its implementation in the training structure of NNs which is discussed in the next section.

2.3.5 Probabilistic Deep Learning

The goal of having a DL network with probabilistic behavior is to produce a distribution either for model parameters or predictions instead of a single class(in classification) or value(in regression) and as a result, an estimation of uncertainty of model weights would help for both solving overfitting issue and also having predictive uncertainty. As

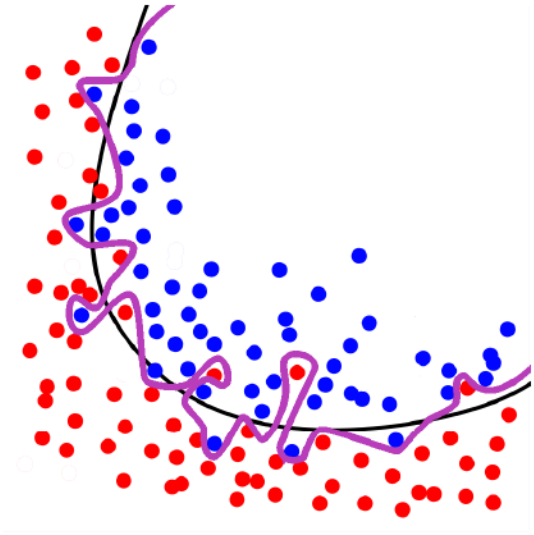


Figure 2.7: Overfitting occurs when a model learns noise and bias in the training data to the point where it severely impairs the model's performance on test data and decreases the accuracy.

mentioned before, we try to enable repeatability either for training or at inference time. A simple real-world example would help for a better understanding of the Practical usage of having model uncertainty .

Assume you aim to go from Carleton University to the University of Ottawa and you have to be there within 15 minutes. You typically use a satellite navigation system like google maps to find the shortest route. The application outputs two routes(single values that represent arrival times) shown in figure 4.1 left. The blue route takes 10 minutes and the gray route takes 12. It seems absolutely reasonable to choose the route within 10 minutes. Now Consider a more sophisticated GPS system that employs a probabilistic model. It not only provides you the best guess for the arrival time, but it also accounts for the uncertainty of that time. As shown in figure 4.1 right Two Gaussian bell curves represent the expected travel-time distributions. The question is how to take benefit from these distributions. Given the described circumstance the

mean value would not be enough to insure you arrive on time. We need to estimate a level of confidence for each prediction by calculating the probability of times less than 15 minutes for both distributions. The percentage of the area under the curve to the left of the dashed line in Figure denotes a crucial value of 15 minutes, corresponding to mentioned probability. You know that your chance of arriving in less than 15 minutes is 93% if you choose the lower route and just 69% if you take the upper route which explains why this is the better choice.

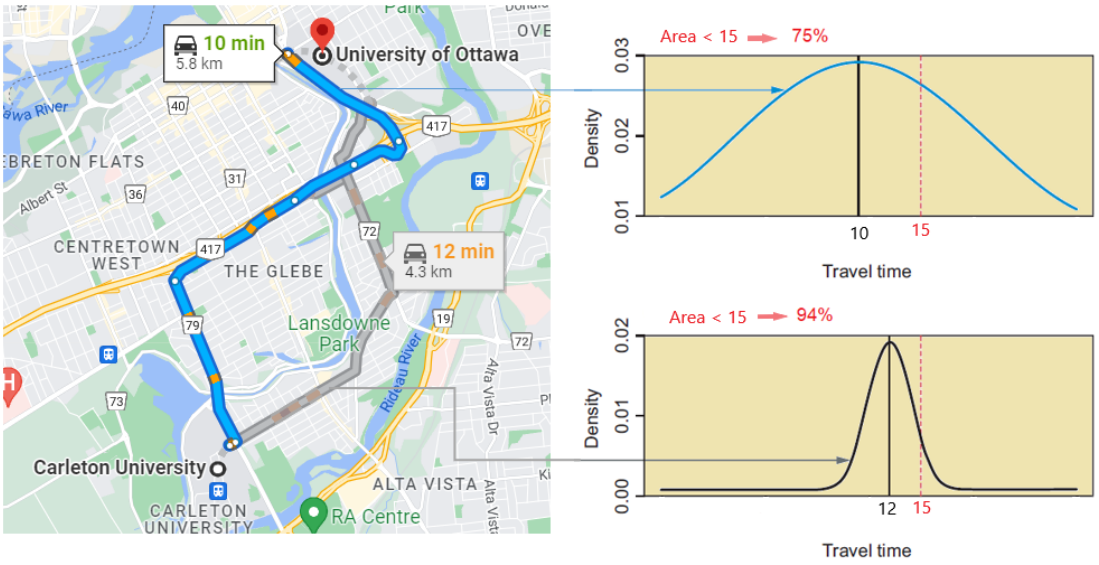


Figure 2.8: The google maps prediction of travel time. A deterministic version of the map, with just one reported number, can be seen on the left. The probability distributions for the travel times of the two routes are displayed on the right side.

In deep neural networks, having a probabilistic network has been a challenge and just a small number of efforts proposed a solution. Depending on the idea they deployed to add randomness to the networks pipelines there are three types of probabilistic neural networks:

- Bayesian Neural Network (BNN)

- Approximate Bayesian Neural Networks (ABNN)
- Ensemble of Neural Networks (ENN)

2.3.5.1 Bayesian Neural Networks

In Bayesian inference, instead of learning the parameter values, we attempt to compute $p(w|D)$ which is the conditional distribution of the weights given the training data. It is also called the posterior distribution, or often the posterior for short. Bayes rule 2.3.5.1 makes it possible to calculate such distribution.

$$p(D|\omega) = \frac{p(\omega)p(D|\omega)}{p(D)} \tag{2.6}$$

While the posterior distribution is simple to express, computing it is often difficult due to the troublesome integral over all potential values of ω. This integral cannot be calculated analytically for the majority of neural networks. We could alternatively estimate it numerically, but this will only be viable for tiny neural networks because ω represents all of the weights and biases, making deep networks incredibly highly dimensional.To address this complexity there are two general algorithms namely sampling-based or variational[24].

2.3.5.2 Variational Inference (VI)

Variational methods, model the posterior using a parameterized distribution $q_\theta(w)$, the structure of which may be easily evaluated. We want the approximation distribution

to be as similar to the original model's posterior distribution as possible. Thus, the Kullback-Leibler (KL) divergence[25] is minimized with respect to, which is intuitively a measure of similarity between two distributions:

$$KL(q_\theta(w)||p(w|X,Y)) = \int q_\theta(w) \log \frac{q_\theta(w)}{p(w|X,Y)} dw \tag{2.7}$$

It should be noted that this integral is only determined when q_θ is absolutely continuous with respect to $p(w|X,Y)$. We denote the minimum of this optimisation target by $q^*_\theta(w)$.

We may estimate the predictive distribution by minimizing the KL divergence:

$$p(y^*|x^*,X,Y) \approx \int p(y^*|X^*,w) q^*_\theta(w) dw = q^*_\theta(y^*|X^*) \tag{2.8}$$

Minimizing KL divergence is also comparable to maximising the Evidence Lower Bound (ELBO) with respect to the variational parameters determining $q_\theta(w)$:

$$\mathcal{L}_{VI}(\theta) = \int q_\theta(w)(Y|X,w) dw - KL(q_\theta(w)||p(w)) \leq (Y|X) \tag{2.9}$$

It establishes the goal of Maximizing the first term in the last equation while minimizing the second term (known as the prior KL) and promotes $q_\theta(w)$ to be in the least possible distance to the prior. VI [26] process is a basic technique in Bayesian modeling which replaces Bayesian modeling marginalization with optimization.The integral computation is replaced with the derivative calculation. However, unlike the optimization methodologies commonly employed in deep learning, we optimize over distributions rather than point estimates in this context.

The main advantage of VI is that Derivatives are considerably easier to calculate than integrals, thus making approximations tractable.

2.3.5.3 Dropout

Srivastava et al. [27] designed the dropout method which addresses the problem of overfitting having much less complex than previous approaches. In this technique, instead of a single network, multiple subnetworks are trained. Each one is built by randomly dropping nodes from the NN's training layers following a probability from a Bernoulli distribution (figure2.9). In the end, All network nodes are used for prediction after the network has been trained.

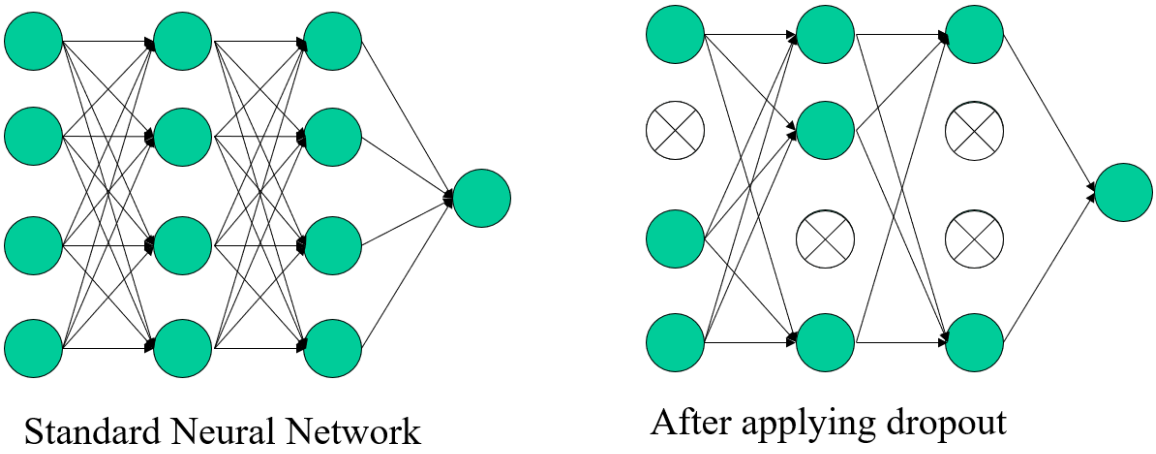


Figure 2.9: left: Shows a standard network. Right : shows how some units are switched off in an iteration of the training process

Approximate Bayesian Neural Networks

2.3.5.4 Monte Carlo Dropout

Inspired by the dropout idea gal and Ghahramani [28] proposed MCDO as a bayesian approximation to estimate predictive uncertainty. As a result, the prediction method now includes some randomness because some nodes are arbitrarily dropped, and the network may provide various outputs for the same input. The input can be applied to the model multiple times when dropout is used during runtime. As the dropout

cancels some nodes with a certain probability, the input is processed by a slightly different network each time. We can calculate an output distribution that provides information about uncertainty using the collection of predictions. They define the dropout loss function as :

$$L_{dropout} = \frac{1}{N}\sum_{i=1}^{N} E(y_i, \hat{y}_i) + \lambda \sum_{i=1}^{L} \left(\|W_i\|_2^2 + \|b_i\|_2^2\right) \tag{2.10}$$

Where having a NN with L layers $\hat{Y}$ denotes NN output and W_i is weight matrices with dimensions $K_i \times K_{i-1}$ and b_i as bias vector for the layer $i = 1, ..., L$

2.3.6 Deep Ensembles

The term "Deep Ensembles" [29] is another major and advance method to enable the estimate of both epistemic and aleatoric uncertainty.

In Deep Ensembles, the ANN is trained to produce an output that includes an approximation of the epistemic uncertainty, and additionally, an ensemble of related models that make predictions based on the same input is used to assess aleatoric uncertainty. The authors suggest a novel method that includes uncertainty in the ANN's output. Given an input x, the ANN produces a predictive distribution rather than predicting an output. Finding the predictive distribution parameters that accurately capture the ANN's prediction and the accompanying uncertainty is the aim of the training procedure. In order to train the network parameters for regression, a Mean Square Error (MSE) loss function is minimized. Nevertheless, this enables

the ANN to forecast the result without taking into account the underlying probability distribution. If the ANN must output the parameters of a predictive distribution, the loss must take these parameters into account so that they might influence the gradient-descent weight optimization process. Due to the assumption that the observation is drawn from a Gaussian distribution, authors' proposal is a negative log-likelihood loss stated in terms of a normal distribution:

$$Loss = -\frac{\log(\ Variance(x))}{2} - \frac{(\ y - Mean(x))\ ^2}{2 \times Variance(x)} + c \tag{2.11}$$

where "c" stands for all constant terms that don't have an impact on loss minimization, $Mean(x)$ is the average, and The variance linked to the input prediction's Gaussian distribution is $variance(x)$.

The maximum likelihood of the combined probability of independent occurrences may be calculated using the log-likelihood function. Assuming that all data are independent, the maximum-likelihood statistic depicts the most probable probability distribution from which the observations were drawn.

If there is a significant discrepancy between the estimated mean and the observation, the second term in the loss equation generates high loss values, which puts pressure on the network to change the weights to lessen the disparity. When this difference, however, cannot be further decreased, the variance in the denominator rises to make up for the loss and minimize it.

Ensemble learning is also suggested by Lakshminarayanan et al. [29] as a way to further enhance uncertainty estimates.It includes training multiple models on the training dataset. These model outputs are combined at runtime using a number of

approaches. Every ensemble member offers a prediction to assess an input. Given that each member expects a Gaussian distribution, all predictions are averaged to get a Gaussian mixed distribution. As a result, the ensemble members' estimates of all the variance values are averaged to get the aleatoric uncertainty and the variance in the means of each ensemble member calculated corresponds to the epistemic uncertainty. To generate a comprehensive assessment of the model's uncertainty, both elements can be merged [30].

Chapter 3

Literature review

In this chapter, we review the works that addressed relevant problems in uncertainty estimation.

3.1 Categorical Properties and Uncertainty Evaluation

Watanabe [31] discusses two views regarding the informativeness of measurement results by nominal properties. The author mentions that according to stevens' [15] traditional theory of measurement since the measurements using nominal values correspond to matching or identifying the identity of measured items with respect to some characteristics, this scale has been regarded as the least informative scale in measurement theory.Nevertheless, with efforts to create a new framework as the basis for testing, diagnosis, identification, and pattern detection(as well-known areas using nominal scale measurement) some researchers argue the first point of view. Watanabe [31] justifies both views as :

Generally, in a measurement procedure, we analyze the measurand in relation to the characteristics being considered before allocating a number or symbol to it. The scale of measurement is then used to get information by referring to the symbol or number, and it serves as a "frame of reference". The more organized such a frame is, the more data would be gathered.Hence, on this basis nominal scale has the simplest form. However, when someone claims that a nominal measurement scale is informative, they frequently refer to information that was obtained from another, pertinent system of empirical knowledge and use that system as their point of reference(even though there can be a type of analytical knowledge evolved in the system).But in the end, there is still a gap in the framework to estimate the uncertainty of the outcome.

Various frameworks and indices have been proposed in metrology and statistical measurements for uncertainty estimation of categorical (nominal or ordinal) variables

.

Capecchi and Lannario [32] argue for the usefulness of computing the Gini heterogeneity index as a non-parametric indicator for detecting uncertainty in a model-based approach for ordinal data. They discussed that the heterogeneity of a discrete random variable R, which assumes m categories (c_j) with probability $p_j, j = 1, ..., m$, differs between a degenerate (minimum uncertainty) and equal probabilities(maximum uncertainty) scenario in a probabilistic framework. Any heterogeneity index should have values of 0 and 1, respectively, in these two extreme cases.

They demonstrate that the uncertainty in predictions is proportional to heterogeneity. As the probability approaches equality for a given m classes, heterogeneity grows. In such a circumstance, the likelihood of correctly predicting a specific instance $(R = c_j)$ is the lowest, and it increases as the distribution begins to collapse on c_j. A

measure of heterogeneity is also offered, utilizing the delta function eq 4.2 of mean difference and actually as a measure of variability proposed by Gini[33] to illustrate the distance between c_i and c_j. For categorical data, this distance has the following form:

$$\dot{(}C_i, C_j) = \begin{cases} 0, & i = j \\ 1, & i \neq j \end{cases} \tag{3.1}$$

Luca Mari et al. [6] discuss the gap for categorical properties from a different perspective. They define the problem on a coarser scale and divide uncertainty evaluation into two main groups namely measurement uncertainty and examination uncertainty where they believe the latter includes nominal properties. They propose a foundational framework to apply to categorical variables. They consider conceptual and operational differences mentioned in [31],[18],[19] between measurement uncertainty and examination(where they put nominal variables in this category) uncertainty.They conclude that measurement uncertainty is not applicable to the examination of nominal properties and needs proper adoption. They tackled this challenge by defining a framework as a generalized version of measurement uncertainty that matches the principles of evaluation uncertainty in metrology [17].

The three layers of evaluation uncertainty (L1, L2, L3) presented in chapter 2 are the basis of the author's work. In fact, Layers L1 and L2 create a foundation upon which layer L3 defines measurement-specific and examination-specific indexes. They presented a case study including an alphabet recognition task that outputs a set of values containing the desired probability to all letters in the reference set. They finally obtain formalized indices of variation as follows:

$$f_1(R) = (R-1)/(C-1) \tag{3.2}$$

where C is the reference set that includes all existing classes and R is the predictions subset from a specified number of repeated examinations(R is the cardinality of the prediction subset) is also normalized between 0 in the case of no uncertainty ($R = 1$) and 1 in case of maximum uncertainty ($R = c$).

Passolo [19] emphasizes the importance of having a proper estimation of examination uncertainty for nominal properties Regardless of whether or not the measurement should be used to describe the attribution of values to nominal attributes.

Iannario [34] proposed a probability distribution produced by a combination (mixture) of discrete random variables to model uncertainty in ordinal data.It is based on a generalization of CUB[35] which is a framework known to analyze responder behavior by simulating the probability distribution of the answer using a (mixed) Combination of Discrete Uniform and Shifted Binomial Random Variables. CUB models consider the response to be a linear synthesis of factors evolved in the final decision. The author highlights that although CUB models presuppose that the ordinal answer variables underlying distribution are multinomial(like standard models); their key requirement is the use of a structured probability distribution. In particular, CUB models need clearly specified probability mass functions, while standard model techniques do not presume knowledge of random variables for the generation process.

Another idea that is applicable to categorical data is imprecise probabilities (Walley 1991[36]). It is based on the concept that information should be specified in terms of a set of probability distributions rather than a single distribution. The idea of credal sets and related conceptions serves as a foundation for an expansion of Bayesian

inference.The main procedure is to exchange a single prior distribution on the model space with a group of candidate priors, as in Bayesian inference. Having a set of data, Bayesian inference may be performed on each possible prior to getting a set of posteriors. Any value or quantity obtained from a posterior is correspondingly replaced by a group of similar values or numbers. A notable example of such a method is the imprecise Dirichlet model for categorical data, which is an extension of inference with the Dirichlet distribution as a conjugate prior for the multinomial distribution (Bernardo 2005[37]).

Yager[38] has established a fundamental distinction between two categories of uncertainty present in a credal set, known as conflict (randomness, discord) and non-specificity, respectively which is studied in [39] as well. Being correspondence with aleatory and epistemic uncertainty respectively is a particular feature of this approach. Shannon entropy [4] and standard uncertainty measure are respectively shown as following equations:

$$H(A) = \log(|A|), \tag{3.3}$$

where uncertain information is represented as subsets $A \subseteq Y$ of potential alternative distributions .

Regarding the epistemic part Abellan and Moral[40] suggested a generalized version of 3.3 :

$$H_G(Q) = \sum_A m_Q(A) \log(|A|) \tag{3.4}$$

where; $m_Q : 2^Y \to [0, 1]$ is the Möbius inverse of the capacity function [39]

Hullermeier and Weageman [41] define total(aggregate) uncertainty as sum of both types :

$$U(Q) = H_G(Q) + H(Q) \tag{3.5}$$

where H(Q) is shannon entropy .

3.2 Uncertainty in Machine learning and Neural Networks

There have been a number of research studies with the purpose of estimating different types of uncertainty in ML and NNs data-driven models. These models have been in a variety of fields to improve prediction quality .

With the aim of proposing a reliable estimate of uncertainty Senge et al.[42] make a point of distinguishing between aleatoric and epistemic uncertainty. They offer a quantification of these uncertainties and demonstrate the applicability of their method in the context of medical decision-making. Kull and Flach[43] apply a very similar approach in the context of their work on reliability maps.

Varshney and Alemzadeh [44] provides an example of a recent autonomous cars

accident that resulted in death (one in 130 million miles of testing) They attribute the car's failure to exceedingly unusual conditions, stressing the necessity of epistemic uncertainty estimation .

Given that x is a model input and y is a prediction distribution, Kendall and Gal [28] show aleatoric uncertainty can be heteroscedastic. They suggest a method for learning aleatoric uncertainty through loss attenuation. The objective is to train the neural network to predict not just the conditional mean of y given x, but also the residual error. The associated loss function to be minimized is designed (or derived from the probabilistic interpretation of the model) in such a manner that prediction mistakes for points with a large residual variance are penalized less, but a penalty is also incurred for forecasting a high variance.

Depeweg et al.[45] make an explicit effort to measure and discriminate aleatoric and epistemic uncertainty for regression models. Their idea is to quantify the total and aleatoric uncertainty, and then calculate the difference to obtain the epistemic uncertainty. They suggest, more particularly, expressing the total uncertainty in terms of the entropy of the predicted posterior distribution:

$$H[\ p(y|x)]\ = -\sum_{y} p(y|x) \log_2 p(y|x) \tag{3.6}$$

Epistemic uncertainty regarding the network weights w is also included in this uncertainty. Thus epistemic uncertainty is eliminated by fixing a set of weights, or by taking into account a distribution $p(y|w, x)$. Consequently, the expectation across these distribution's entropies is an index of aleatoric uncertainty and the difference is used to calculate epistemic uncertainty:

$$_{p(w|D)}H[\ p(y|w,x)]\ = -\int p(w|D)\left(\sum_{y} p(y|w,x)\log_2 p(y|w,x)\right)dw, \tag{3.7}$$

$$U_{epistemic} := H[p(y|x)] -_{p(w|D)} H[p(y|w,x)] \tag{3.8}$$

The epistemic uncertainty intuitively represents the amount of information about the model parameters w that would be obtained by knowing the true y. Mobiny et al.[46] recently took a similar approach for 3.8 approximate computation.

3.2.0.1 Uncertainty Indices

Except for predictive entropy, there are three other uncertainty indices that are commonly used in the researches . Mutual Information(MI) [47] is defined as the information gap between the model parameters and the data and leads to an estimation of epistemic uncertainty. More practically, it means that given the dataset D, the amount of knowledge we would learn about the model's parameters if we received a label y for a new point x:

$$I(w,y|D,x) = H[p(y|x,D)] - E_{P(w|D)}H[p(y|x,w)] \tag{3.9}$$

The next indices are based on the variance of predictive distributions. SoftMax variance and its expanded version as SoftMax variance average. SoftMax variance takes into account the variance of the distribution associated with the class with the

highest probability average .

$$\mu_c = \frac{1}{T}\sum_{t=1}^{T} p(y_c|x, X, Y) \tag{3.10}$$

Where C is the number of classes and y_c is the probability score of a class in the Tth prediction iteration. And SoftMax variance average [48] calculate variance across all classes probability distributions:

$$\sigma^2 = \frac{1}{C}\sum_{c=1}^{C}\sqrt{\frac{1}{T}\sum_{t=1}^{T}(p_c^t - \hat{\mu_c})^2} \tag{3.11}$$

Ghoshal et al. [48] believe that the strategy described above decreases the number of required hyperparameters and accelerates computation. It takes into account the model uncertainty associated with each class prediction.

3.2.1 Uncertainty in Medical diagnosis

We would name some efforts that deployed one of the mentioned indices to have an estimation of an NNs prediction reliability.

We emphasize examples from crucial fields like healthcare, since most AI work in the such area involve a classifications model (which our proposed method focuses on too)

Ghoshal et al. [48] proposed an uncertainty method similar to SoftMax variance for a CNN in the case of Segmenting Nuclei Image Data. Ghoshal and Tucker[47] implemented a CNN for COvid-19 virus detection from chest X-ray images. They

leveraged predictive entropy and Bayesian methods and compared the output uncertainty values on different benchmarks such as dropout rate and prediction accuracy. They also filtered out test samples and set a prediction uncertainty threshold of 0.5 which shows predictive accuracy improvement for the remaining test data .

Angermueller et al. [49] fitted a NN to embryonic stem cell data to predict the methylation rate in DNA (methylation affects the transcription of genes). They demonstrated that model uncertainty grows in difficult-to-predict genomic regions.

Yang et al. [50] conducted a task deploying a CNN to stitch images from brain scans. They illustrated how to get model uncertainty in their context and experimentally tested their model. They showed, for instance, samples with changes in blood flow lead to higher predictive uncertainty.

For diabetic retinopathy detection, Leibig et al.[51] examined the uncertainty information from a deep NNs model. The authors compared dropout-based Bayesian uncertainty estimates to discriminative approaches. They claimed that the Bayesian approximate treatment performed better than the softmax output and that uncertainty-aware decision referral can enhance diagnosis.

Chapter 4

Proposed method

To introduce the proposed index, in this chapter, we first discuss how we use the principles and requirements of uncertainty evaluation in metrology (which we surveyed in chapters 1 and 2) to create a plausible link to deep learning classification. Following that, we outline the workflow and establish the mathematical formulation.

4.1 Idea and Adoption

We listed seven components of uncertainty quantification in chapter 1. These components form the basis of our uncertainty index formulation. The first three emphasize having a proper interpretation of the measurement as the main subject for which we would like to define an index. Since any index is defined over the outcome of the measurement or examination result, it is a critical part to obtain a correct component as the measurand that represents exactly what the task is expected to do. We discuss that through the prediction process of a NN classification model there could be different components to be considered as the measurand. The second and third edition of VIM (ISO [18]) defines the measurand as "the particular quantity subject

to measurement" and "the quantity intended to be measured". Inspired by these definitions, we intend to propose a new perspective to determine the measurement. As seen in chapters 2 and 3, the outcome of the NN classifier is the one-hot encoded of the SoftMax activation function output which is a uniform probability distribution. We discussed four existing indices namely: SoftMax variance, average Variance of all classes, mutual information (3.2.0.1), and predictive entropy (3.2).

The main commonality for all of these approaches is that after deployment of a probabilistic network, they consider the outcome probability distribution as the measured and they defined their uncertainty index over probabilities of either the class with the highest mean value or they evolve all classes in their formulation (more on this in next section). However, we don't consider that distribution as the final outcome the of NN classification model and as we know there will be only a single nominal value that represents a class as the model prediction. Therefore, based on this fact, the problem is switched to the field of categorical variables since we will have a set of discrete values instead of probabilities. Hence, we intend to obtain a formula for predictive uncertainty which corresponds to the attributes and properties of the discrete set as the final outcome.

4.2 Formulation

The SoftMax activation is shown in equation 1 where $z^{\rightarrow}$ corresponds to the output vector of the last layer's neurons, M is the number of classes where $m \in [1, ..., M]$. We refer to the activation of the NN classifier's output layer as the model probability distribution(equation4.1).

$$p_m(z^{\rightarrow}) = \frac{e^{z_m}}{\sum_{i=1}^{m} e^{z_i}} \tag{4.1}$$

To estimate uncertainty, the NN-based classifier must produce a probabilistic output. This means that instead of outputting a probability for each class, it must produce a probabilistic distribution for each class (i.e, class probability distribution). Note that the model probability distribution refers to the set of values calculated by the SoftMax layer for all classes while the class probability distribution refers to the probabilistic output associated with each class. Non-probabilistic NN-based classifiers produce a model probability distribution but do not output class probability distributions. However, probabilistic NN-based models produce a set of M class probability distributions, where M is the number of classes.

The existing uncertainty estimation methods we reviewed in chapter 3, leverage class probability distributions to calculate the predictive uncertainty. To produce a probabilistic output, we can either use a Bayesian NN [24] or approximate Bayesian inference. we leverage monte carlo dropout as a Bayesian inference approximation approach that produces class probability distributions through the calculation of the output over T passes 4.2 or using multiple members of an ensemble [29]. These methods allow us to calculate an outcome frequency distribution. For the scheme proposed in 4.2, we calculate the outcome frequency distribution for a classifier with M classes as follows:

1. During inference, perform T passes on an input x. For each pass, drop certain weights according to a predefined probability. We deploy a dropout layer after the first dense layer (FC6) of the fully connected layer in the trained network.;
2. For each pass, denote the highest output probability $\hat{p}_t$ were $t \in [1, ..., T]$
3. Calculate the outcome frequency distribution F_m Fm which specifies the number

of times during the T passes when class C_m (where $m \in [1, ..., M]$) had the highest output probability $\hat{p}_t$.

We can apply a similar mechanism to calculate the outcome frequency distribution for the method proposed in [29], however, in this case, we do not require T passes, but a single pass for all the members of the ensemble. We denote the class C_F as the class predicted by the model for an input x. Existing methods define C_F as the class with the highest mean within the set of class probability distributions. Hence, they calculate C_F according to equation (4.2).

$$C_F = max\bar{p_m} : m \in [1, ..., M] \tag{4.2}$$

They obtain $\bar{p_m}$ using 4.3 (see Figure 1).

$$\bar{P_m} = \sum_{t=1}^{T} \frac{p_m^t}{M} \tag{4.3}$$

Where p_m^t corresponds to the output of the SoftMax function for class m at pass t and is calculated using 4.2 . For the proposed method, we define CF as the most frequent class in the outcome frequency distribution. Hence, we calculate CF using 4.4.

$$C_F = value(maxF_m : m \in [1, ..., M]) \tag{4.4}$$

Where $value(f)$ is a function that returns the value associated with frequency f in a frequency distribution. F_m is the outcome frequency distribution which we calculate using 4.5 where $\delta(x, y)$ is the Kronecker delta function that is defined in 4.2.

$$F_m = \sum_{t=1}^{T} \delta(\hat{p}_t, p_m^t) \tag{4.5}$$

$$\delta(\hat{p}_t, p_m^t) = \begin{cases} 0, & \hat{p}_t \neq p_m^t \\ 1, & \hat{p}_t = p_m^t \end{cases}$$

We calculate $\hat{p}_t$ using 4.6(Figure 1) .

$$\hat{p}_t = max p_m^t : m \in [1, ..., M] \tag{4.6}$$

We calculate the proposed index for uncertainty evaluation as follows:

1. We denote n_{min} as the minimum frequency required in the outcome frequency distribution to designate a class as C_F. We calculate nMin using 4.7.

$$n_{min} = \lfloor \frac{T}{M} + 1 \rfloor \tag{4.7}$$

When T is much larger than M, we can estimate U_{max} using 4.8.

$$U_{max} \approx 1 - \frac{1}{M} \tag{4.8}$$

U_{max} denotes the maximum uncertainty value that can be calculated for input, before normalization which depends on the number of classes. For example, if there are 3 classes, U_{max} will be 66.6.

3. We denote n_{Miss} as the difference between the sum of all frequencies and the frequency of C_F in the outcome frequency distribution and calculate it using (In fact

n_{Miss} represents the times that the top class(top probability in softmax output) is not the final class (C_F)) 4.9

$$n_{Miss} = \sum_{m=1}^{M} F_m - maxF_m : m \in [1, ..., M] \tag{4.9}$$

4. We denote U as the index of uncertainty and calculate it using 4.10:

$$U = \frac{n_{Miss}}{T \times U_{max}} \tag{4.10}$$

For a better understanding of the procedure by which we obtain the final predicted class (out of all iterations) in order to determine uncertainty value and how different other indices work, figure 4.1 shows while existing indices extract mean probability of each class to reach top-class $\bar{P_m}$ (column-wise), we aim to first get the final class from each test iteration $\hat{p}_t$ which finally forms a discrete set of classes through which we obtain the top class and calculate uncertainty (row-wise).

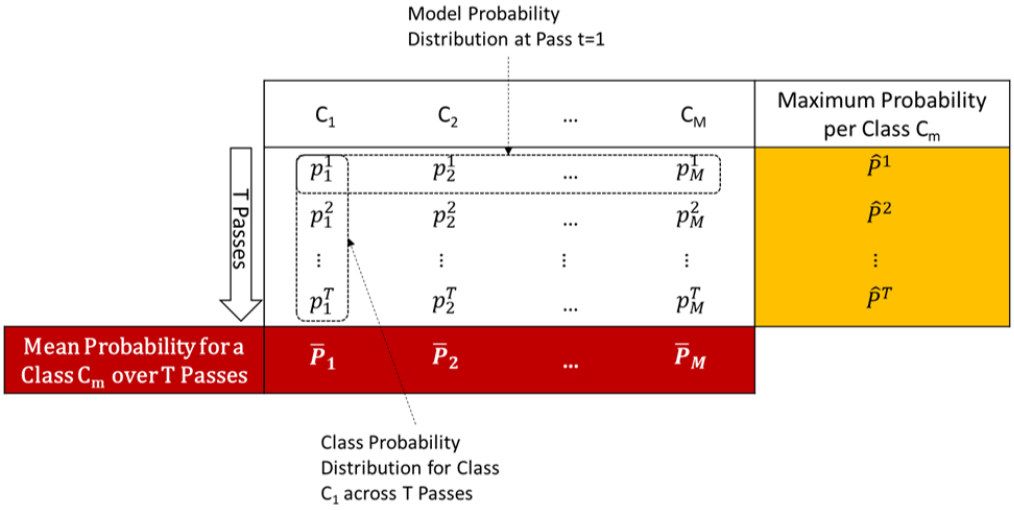


Figure 4.1: difference between the model probability distribution and the class probability distribution

4.3 Extreme Cases

There are two possible extreme situations that could occur during uncertainty calculation regardless of which measurement method is used.

In the first case, a class is selected for all iterations which produce zero uncertainty or total certainty. This scenario is expected for the majority of test samples(we have seen this through our experiments). However, in the second case the opposite situation i.e., when more than one class share exactly the same score in prediction probabilities. It leads to the maximum possible uncertainty or total uncertainty with a value of one. Although this situation has been discussed by [32] at the statistical level, to the best of our knowledge, there are no related studies in ML and NNs discipline that discuss this issue.

For previous indexes, the total uncertainty case occurs when the uniform distribution is split between classes (ex. all out of 3 classes have 0.33 score), while for the proposed index all classes have the same count across all iterations .

Chapter 5

Implementation and Results

For experimental results total of five classification datasets were recruited to assess outcomes from the proposed method and other mentioned indexes. We first explore datasets and models we used to train and test. Afterward, we explain two different benchmarks in order to compare results on this basis. Finally, we discuss training and test accuracy and more importantly, uncertainty values for all datasets in the comparison section.

It should be noted that in order to prevent overfitting, for all datasets we used the dropout tool in training with the rate of 0.4. The same dropout rate is deployed for MCMC sampling at the inference level, however, the usage of the dropout technique is a different application and they don't have any relation, dependency, or effect on each other's process. Moreover, the effect of applying different the different dropout rate is discussed later in this chapter.

Before we go through datasets and practical results, we would mention two possible and specific SoftMax outputs to compare the final uncertainty using existing and proposed indexes over 4 iterations. Table 5.1 gathers all indexes formulations and

table 5.4 shows the results using class distributions in table 5.2 and 5.3 respectively.

Table 5.1: Summary of existing indexes

Method	Final class selection	Uncertainty formula	Description
SoftMax variance - over final class	Highest SoftMax average	$\sigma^2 = \frac{\sum(p_t - \bar{p})}{T}$	Variance of SoftMax distribution of final class
SoftMax Variance – average of all classes	Highest SoftMax average,	$\sigma^2 = \frac{1}{C}\left(\sum_{j=1}^{C}\left(\frac{1}{T}\sum_{i=1}^{T} p_{ij}^2\right) - \hat{p}_j^2\right)$	Mean-variance of SoftMax probabilities across all classes
Predictive entropy(H)	Highest SoftMax average,	$H(y\|x,X,Y) = -\sum_c p(y\|x,X,Y)\log p(y = c\|x,X,Y)$	Entropy of the probability distributions
Mutual information score	Highest SoftMax average,	$\bar{I} = H(\bar{p} - \frac{1}{T}\sum_i H(p_i))$	information gain between the model parameters and the data
Proposed method	most times as highest SoftMax prediction	$U = \frac{N_{miss}}{T \times U_{max}}$ Eq.4.10, Eq.4.9	Number of mismatches (changes in model predicted label)

Table 5.2: SoftMax output of a common scenario with low predictive uncertainty

Forward pass	C1	C2	C3
1	0.97	0.02	0.01
2	0.99	0.01	0
3	0.96	0.01	0.03
4	0.98	0	0.02

Table 5.3: SoftMax output of an unlikely but possible case

Forward pass	C1	C2	C3
1	0.45	0.46	0.09
2	0.49	0.51	0.02
3	0.47	0.48	0.03
4	0.51	0.45	0.04

Table 5.4: Predictive uncertainty output from all five indexes using network outputs in table 5.2,5.3

Index	Scenario 1	Scenario 2
SM variance	0.00012	0.00051
SM variance average	0.017	0.14
Mutual information	0.015	0.13
Predictive entropy	0.041	0.300
Proposed method	0.0	0.371

In tables 5.2 and 5.3, we chose numbers to have low and high uncertainty conditions respectively to discuss how different indexes would change through these scenarios. As shown in table 5.4 for the first scenario final class is C_1 for all methods and all of them give small numbers close to zero while the proposed method gives zero as there were not switches in predictions. Maybe it's taken as a weakness of this method that however, in real-world scenarios it would be pointless to either have a very tiny number close to zero or exact zero, since they all show high confidence in prediction and no need for a second process .

However, the second scenario, it is a completely different case where we can see the effects of an index in both final class selection and uncertainty value. For final class selection, all four previous indexes pick the class with a higher mean value which would be C_1 with 0.48 mean value. However, the proposed method would pick C_2 which has the upper value in the first three prediction sets. Moreover, regarding predictive uncertainty where we expect higher numbers compared to the first case , except SoftMax variance (which shows no significant changes due to almost the same deviation in top-class probabilities and as a result it couldn't be a reliable index) other indexes output considerably increased values. Finally, we should note the maximum number belongs to the proposed method.

We explained in the previous chapter the normalization process for the proposed method. For existing indices, we normalize the results considering the most extreme scenarios. For top-class variance and variance average, the most extreme case is when the top-class probability distribution is flat between 0 to 1. which means when all class probabilities are completely uniform. The uncertainty value, in this case, depends on the number of classes and equal probabilities would be:

$$p_m = \frac{100}{m}$$

where m denotes the number of classes. Based on this, table 5.5 shows the maximum uncertainty value for existing indices for a model with 3 classes. For predictive entropy and mutual information, since these are entropy-based indexes, the most extreme scenario happens when there are uniform distributions which in this example is 0.33.

Table 5.5: Equal probability for 3 classes is 0.33 and uncertainty results are normalized using the values

Index	Maximum uncertainty
Top class variance	0.15
SM variance average	0.15
Mutual information	0.79
Predictive entropy	15.8

5.1 Datasets

We selected datasets in a way to cover various types of classifications including binary, multiclass, ordinal, and nominal with a focus on medical cases. Moreover, datasets are in a different range of training accuracy and number of training and test data samples to examine how it could affect uncertainty outcome.

5.1.1 Bipolar Severity Detection

Ciftçi et al. [52] built the Turkish Audio-Visual BD dataset, which was utilized in AVEC 2018. The dataset is available for academic use. It shows video samples of organized interactions with 47 people, 35 of who are men and 16 of whom are women. The respondents were instructed to do seven tasks, including speaking about joyful and painful recollections and counting to thirty while the video was being recorded. Mania (very heightened arousal), Hypomania (euphoria, but less severe than Mania), and Remission are the three clinically different moods that BD patients usually experience. These three categories are used as three class labels to fit a NN classification model. Ciftçi et al. [52] stated the following YMRS score range for each degree of bipolar disorder:

1. Remission: $YMRS \leq 7$
2. Hypo-Mania: $7 < YMRS < 20$
3. Mania: $YMRS \geq 20$

The dataset comprises a total of 218 video recordings, which are divided into 104, 60, and 54 videos for the training, validation, and test phases. Among existing models

[53],[52], [54], [55] we chose the model presented by Abaei et al[55]. which deployed a hybrid CNN-LSTM method by which Each video recording's frames are retrieved at a frame rate of 30 Hz and in total, around 2 million frames are captured. A more detailed structure is shown in 5.1

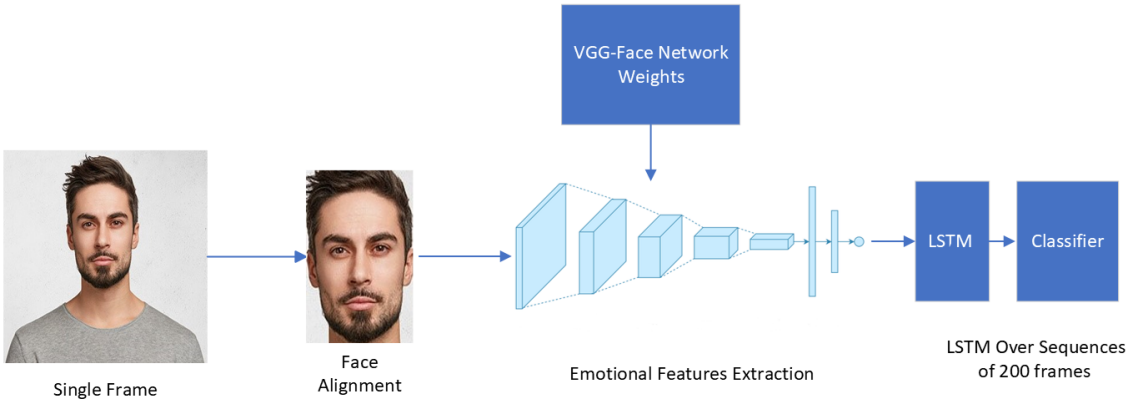


Figure 5.1: after retrieving all frames, the detected face is fed into a CNN to extract facial emotional features and after preparing required sequences as LSTM inputs(which we choose 200), a classifier at the top of the model network predict one of 3 labels.(Remission, Hyper mania, and Mania)

5.1.2 MNIST and Fashion-MNIST

The MNIST database is a handwritten digit database that is often used to train various image processing algorithms. it contains 60,000 training and 10,000 testing 28×28 grayscale images representing 10 classes (0-9). Similarly, fashion-mnist dataset contains the same number of both training and testing samples from Zalando's article (cloths) and the classes are respectively: Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot .

for both datasets we utilized a CNN with a 3×3 filter followed by a max pooling layer and the output layer has 10 nodes corresponding to 10 classes. We apply the ReLU

activation function to all layers, and the stochastic gradient descent optimizer is set with a learning rate of 0.01 and a momentum of 0.9. The categorical cross-entropy loss function, which is ideal for multi-class classification, will be optimized.

5.1.3 Breast Cancer Detection

According to global statistics, breast cancer is one of the most frequent diseases among women globally, accounting for the majority of new cancer cases and cancer-related deaths, making it a serious public health burden.Early diagnosis of BC can dramatically improve patients' prognoses and chances of survival by promoting timely therapeutic therapy.

The University of Wisconsin Hospital provided a dataset containing 699 samples (357 benign and 212 malignant) in total, each with nine input characteristics and one target attribute which is a binary classification between "M = malignant" and "B = benign". It includes digital photos of a breast mass Fine Needle Aspirate (FNA). They describe the features of the cell nuclei seen in the image.

A NN with 3 hidden layers and 512 units is deployed for training and using adam optimizer and categorical cross-entropy loss.

5.1.4 DEAP Dataset

DEAP dataset presented by Koelstra et al. [56], is used to analyze emotions using EEG, physiological, and visual inputs. A multimodal dataset for analyzing human emotional states is presented. The electroencephalograms (EEGs) and peripheral physiological data of 32 individuals were captured while they watched 40 one-minute music video snippets. Each film was scored by participants based on its arousal, va-

lence, like/dislike, dominance, and familiarity. Frontal face footage was also recorded for 22 of the 32 individuals.

For feature extraction, we employed a pre-trained CNN, GoogLeNet [57], and for classification, it employs a block-based residual LSTM network. We increase the number of channels to three to input the raw EEG data to the pre-trained CNN, as shown in [32]. To meet this condition, we employ a 2D convolutional layer with a kernel size of $3x3$ and a stride of 1 over the input. On the output of the convolutional layer, we use a ReLU activation function and batch normalization. As we use CNN as a feature extractor, we remove the top fully connected layer. The hyperparameters used in this work for optimization include learning rate, number of LSTM layers (which determines the number of blocks), number of stacked LSTMs on each LSTM layer, size of the first LSTM layer, and step size to decrease the output size of each LSTM layer accordingly. We deployed the grid search method which finds the best possible combination of hyper-parameters on training and test accuracy. In our model, it yielded the following values: 0.0001 learning rate, 2 stacked LSTMs, 100 hidden sizes for the first LSTM, 3 LSTM layers, and 15 step sizes.

5.2 Benchmarks

There are mainly two tasks to have an assessment of uncertainty quality using different indexes. We utilized both of them to compare the outcomes of the proposed index with the other four indexes.

5.2.1 True/False Wasserstein Distance

Uncertainty comparison for all test samples grouped by correct and wrong predictions can give a reasonable estimation of how informative is the outcome uncertainty. In fact, it leads to two distributions (correct and wrong) and the larger is the distance between distributions it is more probable to have the wrong prediction with a higher uncertainty value. It is used in [58] for covid-19 detection based on chest medical images however they didn't use any distance to represent a value. Two well-known statistical distances are Kullback–Leibler (KL) divergence (eq .5.1) (where $PandQ$ are distributions with the same probability space) and Wasserstein Distance (WD) (eq .5.4)(where F_A^{-1} and F_B^{-1} are the corresponding quantile functions, and F_A and F_B are the associated Cumulative Distribution Function (CDF)) from which we choose to use WD. The main difference between them is that KL divergence is not a metric since it's not symmetric (eq.5.2) and is incompatible with the triangle inequality (eq. 5.3) while WD is. Moreover, WD is a distance function between probability distributions on a certain metric space M that has been specified .

$$D_{KL}(p||Q) = \sum_{x \in X} p(x) \log \frac{p(x)}{Q(x)}. \tag{5.1}$$

$$D_{KL}(P||Q) \neq D_{KL}(Q||P) \tag{5.2}$$

$$D_{KL}(R||P) \leq D_{KL}(Q||P) + D_{KL}(R||Q) \tag{5.3}$$

$$W := W(F_A, F_B) = \left(\int_0^1 |F_A^{-1}(u) - F_B^{-1}(u)|^2 du \right)^{\frac{1}{2}}, \tag{5.4}$$

5.2.2 Accuracy vs Uncertainty

Lakshminarayanan et al [29]. take into consideration a task where the uncertainty efficiency is only assessed on situations when the model's confidence is over a user-specified threshold in order to assess the utility of predictive uncertainty for decision-making. One may trust the model's results if the confidence estimates are calibrated. When the reported level of confidence is high (low uncertainty), predictions are made; when it is low (high uncertainty), a different approach may be taken.

Following this task we consider uncertainty values to obtain how much predictive accuracy is correlated with uncertainty trends comparing different indexes and datasets.

5.3 Results and Comparison

5.3.1 Dropout rate

Authors in [58] discussed the effect of different dropout rates at inference time on the outcome predictive uncertainty.They tested three dropout rates (0.1,0.3,0.5) and there is no obvious correlation between uncertainty values and dropout rate increase. However, theoretically, when the dropout rate is higher it means that more weights are probably dropped and as a result, the predictions are made based on more diverse networks. Hence, we would expect higher sparsity either in softmax probabilities(class probabilities) or the final predicted class(top class) in the series of predictions.

In this regard, we also examined the outcomes of 10 test data from the MNIST dataset

using three different dropout rates (0.1, 0.4, 0.7) and calculated average uncertainty using all indexes. Results are shown in table 5.6. There is a tiny increase in uncertainty values however it is not consistent and it cannot be interpreted as a correlation.

Table 5.6: Uncertainty results for 3 different dropout rates on MNIST test data

Dropout rate	proposed index	predictive entropy	top class variance	SM variance average	mutual information
0.1	0.5299	0.4867	0.2677	0.2712	0.1431
0.4	0.5354	0.4889	0.2842	0.2578	0.1690
0.7	0.5219	0.4801	0.2961	0.2913	0.1545

5.3.2 Softmax Temperature

SoftMax temperature is a hyperparameter that is applied to logits to influence the softmax's final probabilities.

- A low temperature (below 1) makes the probabilities flatter (values closer to eachother).
- A high temperature (above 1) makes the probabilities more biased to one of the classes .

In this regard , facing the first situation, it raises the porbability that we have a higher n_{miss} 4.9 in proposed formulation and as a result it will have a tiny effect to increase the outcome uncertainty value compared to the second scenario in which the model is more confident with one of the classes .

Table 5.7 shows training and test accuracy for all datasets mentioned above . Figures 5.2 to 5.4 visualize uncertainty distributions grouped into True/False pre-

dictions for all methods over three existing datasets (We chose these three datasets which showed more distinguishable distances visually). Ideally, we expect more false predictions to show higher uncertainty values. Therefore, we consider uncertainty outputs that are more correspond to this feature, more informative, and reliable. In some figures, the corresponding distance is more obvious between the two distributions. In figure 5.2 (b) there are 2 picks in both false and true predictions. the first picks are in 0 uncertainty value which is justified with our method and it explains predictions with a single top class over all 100 iterations. However, the second peaks correspond to higher uncertainty among which the uncertainty value for false predictions (Orange pick) is two times more that in true cases (blue pick). For more clarification, table 5.8 contains the final values of WD. We can say the top two choices for all datasets would be the proposed method and predictive entropy. In four out of five cases, the proposed method shows better results than predictive entropy although they are fairly close . Figure 5.6 illustrates the accuracy trend when uncertainty raises. We built batches of test samples with obtain uncertainty values and tried to consider having at least 10 samples in each uncertainty point to have a more reliable comparison between different indecies. For example, As shown in figure 5.6 (a) for softmax variance and predictive entropy there is a negative correlation between uncertainty and prediction accuracy in $U = 0.25 - 50, U = 0.50 - 0.75$ respectively. The remaining indexes show Descending trends in the entire interval.

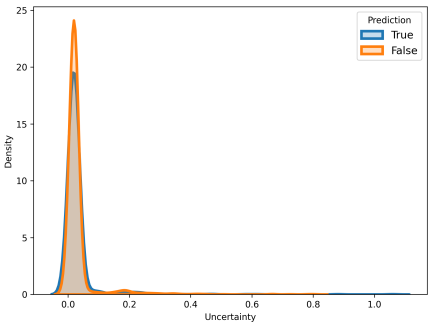
Moreover, we must note different indexes could lead to different accuracy rates in two extreme cases (e.g. zero and one uncertainty). Although maximum uncertainty can be due to a lack of sample points which makes the accuracy not very reliable but for zero uncertainty which normally includes enough data points we can see up to 6% accuracy difference.

Table 5.7: Training and test accuracy

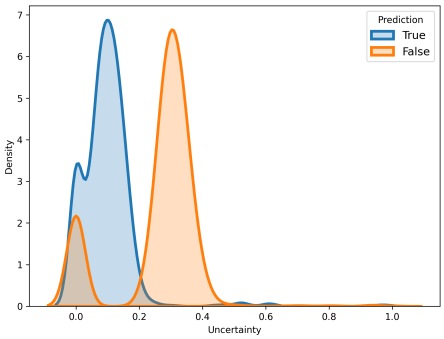
Dataset	Training accuracy	Test accuracy
Bipolar Severity Diagnosis	0.6035	0.5270
Breast Cancer detection	0.8638	0.7896
DEAP dataset	0.7121	0.6194
MNIST	0.9632	0.9091
Fashion-MNIST	0.9451	0.8914

Table 5.8: Wasserstein distance between True/False predictions distributions based on uncertainty values over all test samples

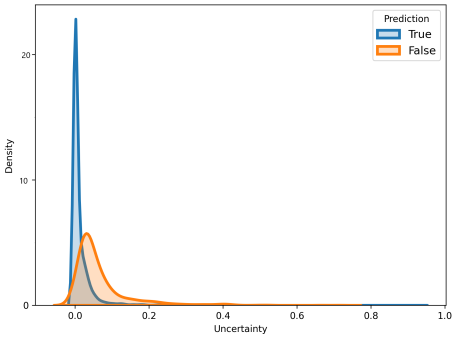
Method \ Dataset	Bipolar corpus	Breast Cancer	MNIST	Fashion - MNIST	DEAP
Proposed method	0.0568	0.3405	0.3875	0.1103	0.0841
SoftMax variance - final class	0.0023	0.2618	0.2715	0.0589	0.0466
SoftMax variance average	0.0027	0.2618	0.2632	0.0617	0.0510
Predictive entropy (H)	0.0057	0.3117	0.4164	0.0398	0.0678
Mutual information	0.0032	0.1317	0.3335	0.0566	0.0481



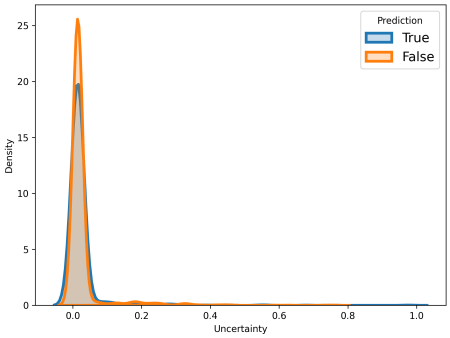
(a) SoftMax variance - top class



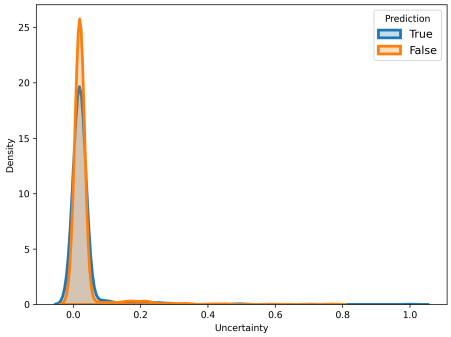
(b) Proposed method



(c) Predictive entropy

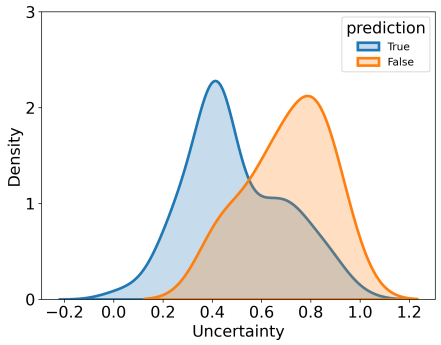


(d) Mutual information

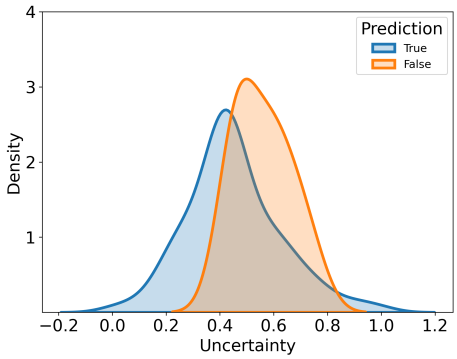


(e) SoftMax variance average

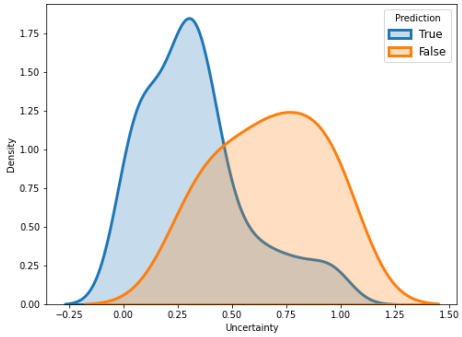
Figure 5.2: Uncertainty distributions for test data grouped by True/wrong predictions - Bipolar corpus dataset



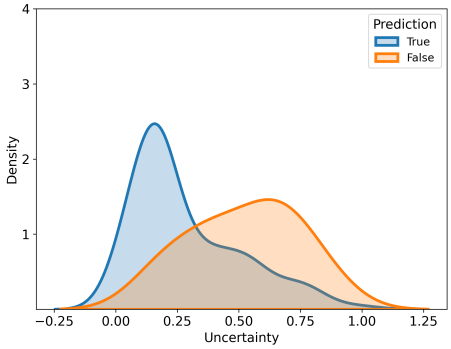
(a) SoftMax variance - top class



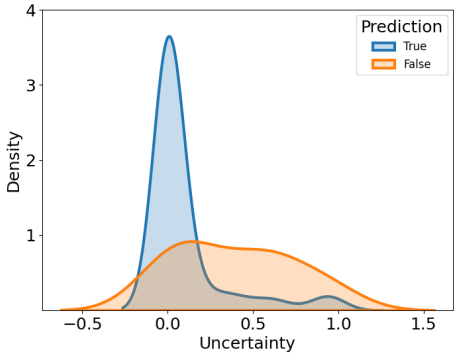
(b) SoftMax variance average



(c) Predictive entropy

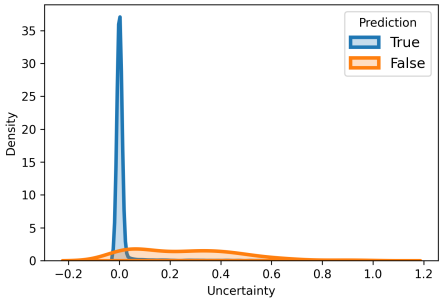


(d) Mutual information

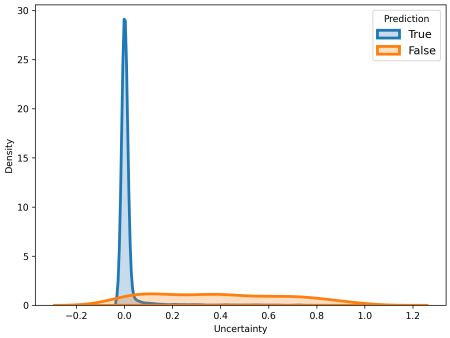


(e) Proposed method

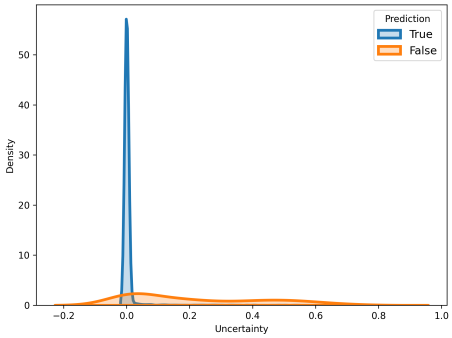
Figure 5.3: Breast cancer dataset



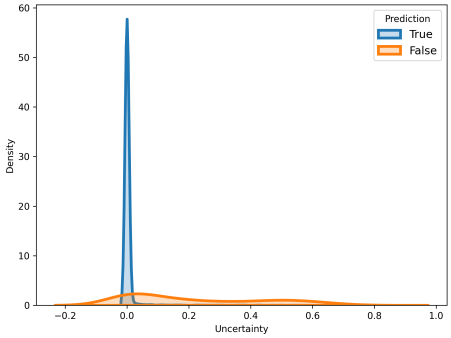
(a) SoftMax variance - top class



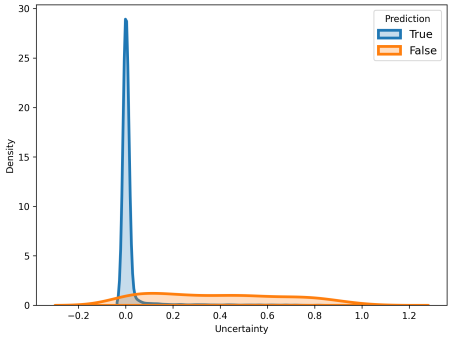
(b) Proposed method



(c) SoftMax variance average

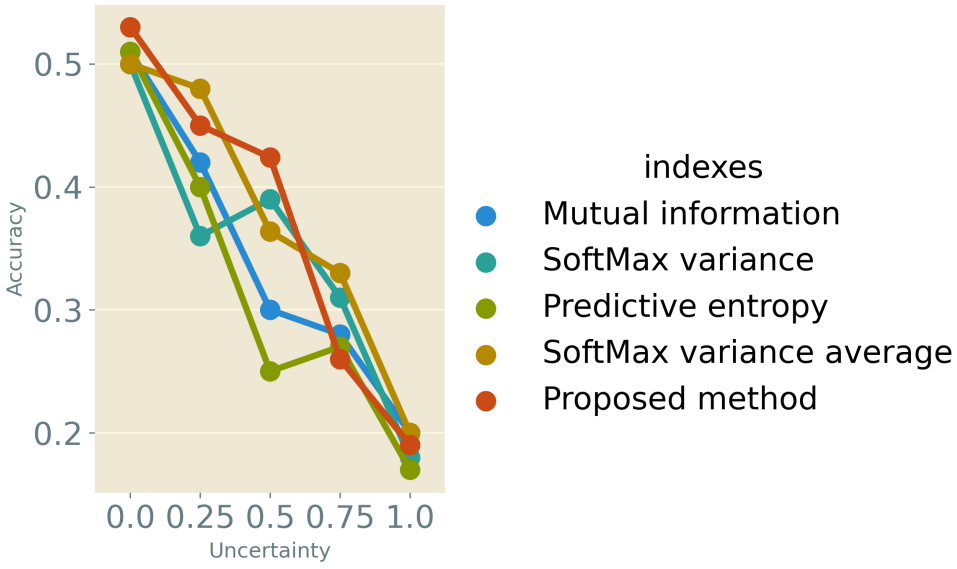


(d) Mutual information

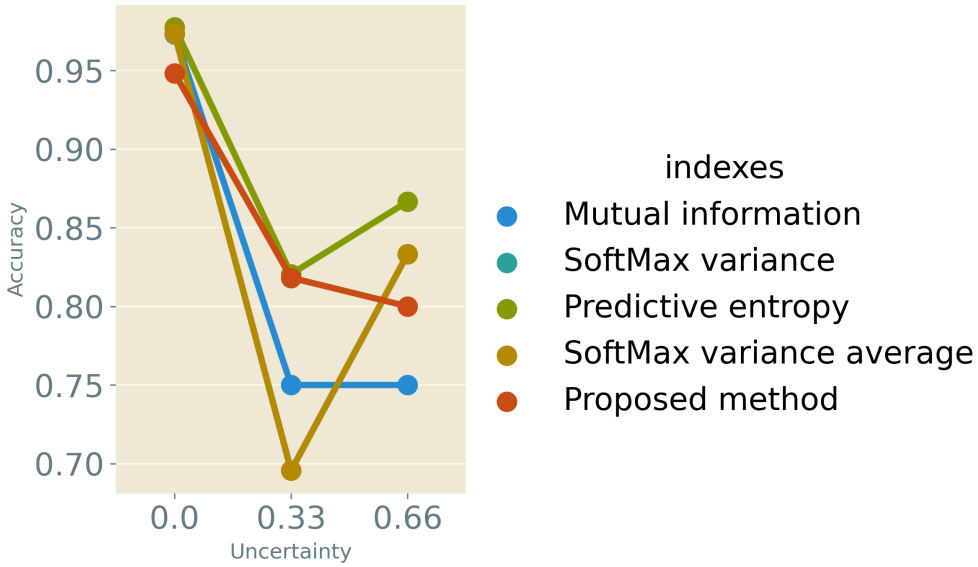


(e) Predictive entropy s

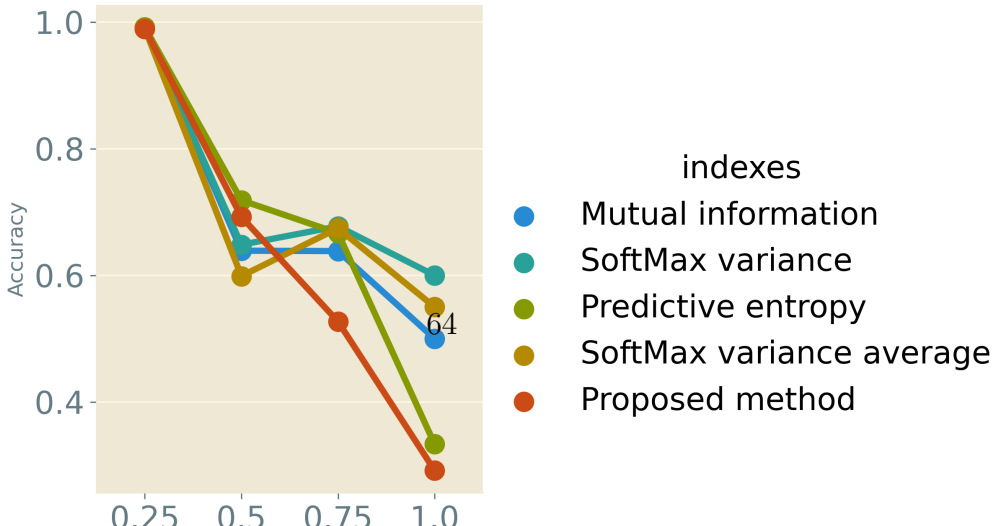
Figure 5.4: MNIST datset

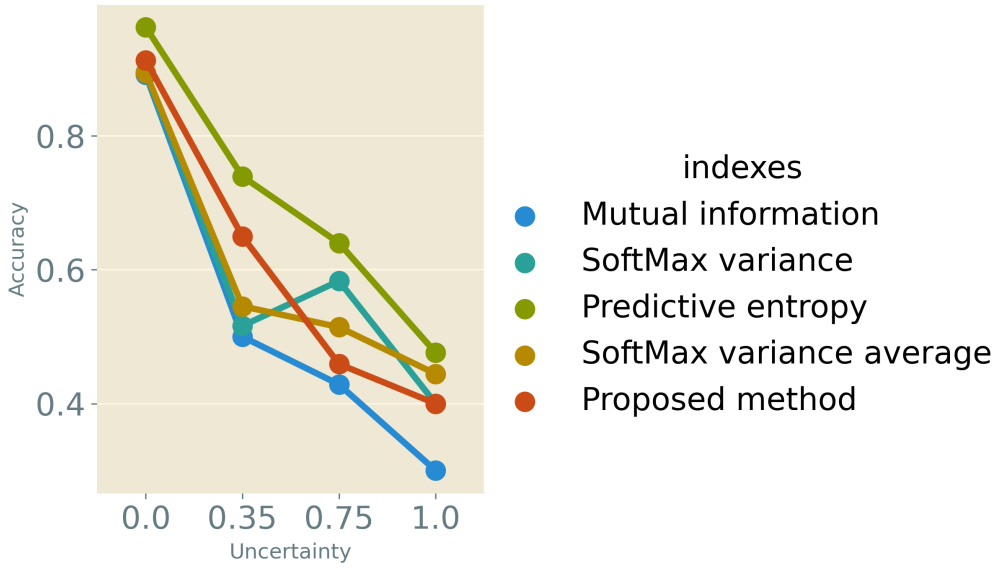


(a) Firts subfigure.

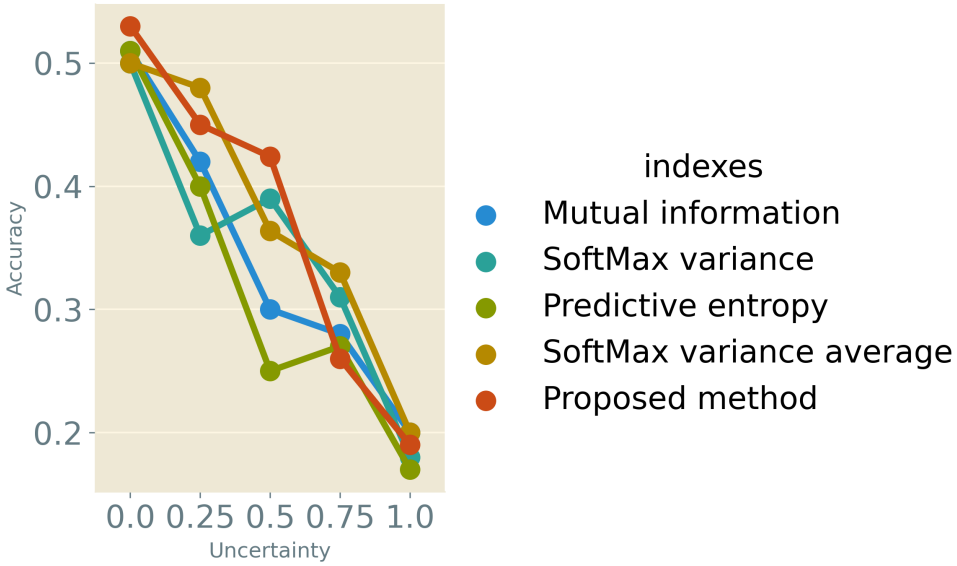


(b) Second subfigure.





(a) Third subfigure.



(b) Third subfigure.

Figure 5.6: Uncertainty vs accuracy as a benchmark for uncertainty quality assessment

Chapter 6

Conclusion and future work

6.1 Conclusion

Through this thesis, we studied the impacts of choosing a new measurement method on uncertainty quantification in predictions of a NNs-based classification model.
Since this significance directly depends on how we approach choosing the final class, we applied a new format of the process. Inspired by the components and concepts in metrology of uncertainty quantification for categorical variables, we consider the measurand in a different way compared to existing methods and it is seen not only the final predicted class would be changed but even there would be a different uncertainty value.
Furthermore, it is shown a new index can lead to a higher quality of uncertainty which makes it more reliable and worthy to pay attention to predictions with a high uncertainty value. It is notable that These techniques generally are exploited to capture rare cases to prevent irreparable damages.

6.2 Limitations and Future works

The main limitation of the proposed index is the possible scenario of having exactly the same prediction counts for all classes. For example, having 50-50 counts for two classes in a binary classification over 100 predictions. This means the model couldn't pick a class and further process is needed to figure out the final class issue.

6.2.1 Uncertainty Fusion

In multimodal data models, we are dealing with different data formats (such as audio and video) at the same time. Accordingly, we would have different sources of uncertainty. Hence, predictive uncertainty evaluation is more challenging and as a future idea it might be helpful to find a way to calculate either aleatoric or epistemic uncertainty separately and somehow fuse them into a single value to conduct a more comprehensive estimation

6.2.2 Turning Predictions

As seen in previous chapters, there is still a lack of proper assessment methods to obtain outcome quality of uncertainty evaluation. Turning prediction can be one of the possible options. It means switching the predicted class to the second choice for all samples with uncertainty over a preobtained threshold (like 0.7) and recalculating the test accuracy.

Bibliography

[1] Andy Alaszewski and Kirstie Coxon. Uncertainty in everyday life: Risk, worry and trust, 2009.

[2] Alan Agresti. *Categorical data analysis*. John Wiley & Sons, 2003.

[3] Edelgard Hund, D Luc Massart, and Johanna Smeyers-Verbeke. Operational definitions of uncertainty. *Trac trends in analytical chemistry*, 20(8):394–406, 2001.

[4] Sina Shafaei, Stefan Kugele, Mohd Hafeez Osman, and Alois Knoll. Uncertainty in machine learning: A safety perspective on autonomous driving. In *International Conference on Computer Safety, Reliability, and Security*, pages 458–464. Springer, 2018.

[5] Daily Milanés-Hermosilla, Rafael Trujillo Codorniú, René López-Baracaldo, Roberto Sagaró-Zamora, Denis Delisle-Rodriguez, John Jairo Villarejo-Mayor, and José Ricardo Núñez-Álvarez. Monte carlo dropout for uncertainty estimation and motor imagery classification. *Sensors*, 21(21):7241, 2021.

[6] Luca Mari, Claudio Narduzzi, Gunnar Nordin, and Stefanie Trapmann. Foundations of uncertainty in evaluation of nominal properties. *Measurement*, 152:107397, 2020.

[7] AR Wilcox. Indices of qualitative variation (ornl-tm-1919). *Oak Ridge: Oak Ridge National Lab*, 1967.

[8] Stephen LR Ellison and Alex Williams. Quantifying uncertainty in analytical measurement. 2012.

[9] JCGM Jcgm et al. Evaluation of measurement data—guide to the expression of uncertainty in measurement. *Int. Organ. Stand. Geneva ISBN*, 50:134, 2008.

[10] JCGM Jcgm et al. Basic and general concepts and associated terms (vim). *Int. Organ. Stand. Geneva ISBN*, 50:200, 2008 Edition with Minor Corrections.

[11] Edelgard Hund, D Luc Massart, and Johanna Smeyers-Verbeke. Operational definitions of uncertainty. *Trac trends in analytical chemistry*, 20(8):394–406, 2001.

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

[13] M Elisabeth Paté-Cornell. Uncertainties in risk analysis: Six levels of treatment. *Reliability Engineering & System Safety*, 54(2-3):95–111, 1996.

[14] Michael Havbro Faber. On the treatment of uncertainties and probabilities in engineering decision analysis. 2005.

[15] Stanley Smith Stevens. On the theory of scales of measurement. *Science*, 103(2684):677–680, 1946.

[16] Douglas P Wagner, William A Knaus, and Elizabeth A Draper. Statistical validation of a severity of illness measure. *American journal of public health*, 73(8):878–884, 1983.

[17] Luca Mari. Toward a harmonized treatment of nominal properties in metrology. *Metrologia*, 54(5):784, 2017.

[18] I Iso and BIPM OIML. Guide to the expression of uncertainty in measurement. *Geneva, Switzerland*, 122:16–17, 1995.

[19] Antonio Possolo. Statistical models and computation to evaluate measurement uncertainty. *Metrologia*, 51(4):S228, 2014.

[20] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[22] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[23] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.

[24] Jouko Lampinen and Aki Vehtari. Bayesian approach for neural networks—review and case studies. *Neural networks*, 14(3):257–274, 2001.

[25] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[26] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

[27] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[28] 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. PMLR, 2016.

[29] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

[30] Hussein Al Osman and Shervin Shirmohammadi. Machine learning in measurement part 2: uncertainty quantification. *IEEE Instrumentation & Measurement Magazine*, 24(3):23–27, 2021.

[31] Hiroshi Watanabe. Coarse-grained information in formal theory of measurement. *Measurement*, 38(4):295–302, 2005.

[32] Stefania Capecchi and Maria Iannario. Gini heterogeneity index for detecting uncertainty in ordinal data surveys. *Metron*, 74(2):223–232, 2016.

[33] Corrado Gini. *Variabilità e mutabilità: contributo allo studio delle distribuzioni e delle relazioni statistiche.[Fasc. I.]*. Tipogr. di P. Cuppini, 1912.

[34] Maria Iannario. Modelling uncertainty and overdispersion in ordinal data. *Communications in Statistics-Theory and Methods*, 43(4):771–786, 2014.

[35] Maria Iannario and Domenico Piccolo. Cub models: Statistical methods and empirical evidence. *Modern Analysis of Customer Surveys: with applications using R*, pages 231–258, 2011.

[36] Peter Walley. Statistical reasoning with imprecise probabilities. 1991.

[37] José M Bernardo and Adrian FM Smith. *Bayesian theory*, volume 405. John Wiley & Sons, 2009.

[38] Ronald R Yager. Entropy and specificity in a mathematical theory of evidence. In *Classic works of the Dempster-Shafer theory of belief functions*, pages 291–310. Springer, 2008.

[39] David Kendall. Obituary: Alfréd rényi. *Journal of Applied Probability*, 7(2):508–522, 1970.

[40] Joaquín Abellán, George J Klir, and Serafín Moral. Disaggregated total uncertainty measure for credal sets. *International Journal of General Systems*, 35(1):29–44, 2006.

[41] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110(3):457–506, 2021.

[42] Robin Senge, Stefan Bösner, Krzysztof Dembczyński, Jörg Haasenritter, Oliver Hirsch, Norbert Donner-Banzhoff, and Eyke Hüllermeier. Reliable classifica-

tion: Learning classifiers that distinguish aleatoric and epistemic uncertainty. *Information Sciences*, 255:16–29, 2014.

[43] Meelis Kull and Peter A Flach. Reliability maps: a tool to enhance probability estimates and improve classification accuracy. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 18–33. Springer, 2014.

[44] Kush R Varshney and Homa Alemzadeh. On the safety of machine learning: Cyber-physical systems, decision sciences, and data products. *Big data*, 5(3):246–255, 2017.

[45] Stefan Depeweg, Jose-Miguel Hernandez-Lobato, Finale Doshi-Velez, and Steffen Udluft. Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. In *International Conference on Machine Learning*, pages 1184–1193. PMLR, 2018.

[46] Aryan Mobiny, Pengyu Yuan, Supratik K Moulik, Naveen Garg, Carol C Wu, and Hien Van Nguyen. Dropconnect is effective in modeling uncertainty of bayesian deep networks. *Scientific reports*, 11(1):1–14, 2021.

[47] Thomas M Cover, Joy A Thomas, et al. Entropy, relative entropy and mutual information. *Elements of information theory*, 2(1):12–13, 1991.

[48] Biraja Ghoshal, Allan Tucker, Bal Sanghera, and Wai Lup Wong. Estimating uncertainty in deep learning for reporting confidence to clinicians when segmenting nuclei image data. In *2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS)*, pages 318–324, 2019.

[49] Lee-H.J. Reik W et al Angermueller, C. Deepcpg: accurate prediction of single-cell dna methylation states using deep learning. *Genome Biol*, (18), 2017.

[50] Xiao Yang, Roland Kwitt, and Marc Niethammer. Fast predictive image registration. *CoRR*, 2016.

[51] Allken-V. Ayhan M.S. et al Leibig, C. Leveraging uncertainty information from deep neural networks for disease detection.

[52] Elvan Çiftçi, Heysem Kaya, Hüseyin Güleç, and Albert Ali Salah. The turkish audio-visual bipolar disorder corpus. In *2018 First Asian Conference on Affective Computing and Intelligent Interaction (ACII Asia)*, pages 1–6. IEEE, 2018.

[53] Le Yang, Yan Li, Haifeng Chen, Dongmei Jiang, Meshia Cédric Oveneke, and Hichem Sahli. Bipolar disorder recognition with histogram features of arousal and body gestures. In *Proceedings of the 2018 on Audio/Visual Emotion Challenge and Workshop*, pages 15–21, 2018.

[54] Fabien Ringeval, Björn Schuller, Michel Valstar, Roddy Cowie, Heysem Kaya, Maximilian Schmitt, Shahin Amiriparian, Nicholas Cummins, Denis Lalanne, Adrien Michaud, et al. Avec 2018 workshop and challenge: Bipolar disorder and cross-cultural affect recognition. In *Proceedings of the 2018 on audio/visual emotion challenge and workshop*, pages 3–13, 2018.

[55] Niloufar Abaei and Hussein Al Osman. A hybrid model for bipolar disorder classification from visual information. In *ICASSP*, volume 2020, pages 4107–4111, 2020.

[56] Sander Koelstra, Christian Muhl, Mohammad Soleymani, Jong-Seok Lee, Ashkan Yazdani, Touradj Ebrahimi, Thierry Pun, Anton Nijholt, and Ioannis Patras.

Deap: A database for emotion analysis; using physiological signals. *IEEE transactions on affective computing*, 3(1):18–31, 2011.

[57] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[58] Biraja Ghoshal and Allan Tucker. Estimating uncertainty and interpretability in deep learning for coronavirus (covid-19) detection. *arXiv preprint arXiv:2003.10769*, 2020.