UNIVERSIDAD DE INVESTIGACIÓN DE TECNOLOGÍA EXPERIMENTAL YACHAY

Escuela de Ciencias Matemáticas y Computacionales

Procesos Gaussianos Profundos y Redes Neuronales Infinitas para el Análisis de Señales EEG en Enfermedades de Alzheimer

Trabajo de integración curricular presentado como requisito para la obtención del título de Matemático

Autor/a:

Andy Mauricio Cumbicus Jiménez

Tutor/a:

Ph.D - Saba Infante

Urcuquí, Octubre 2022

Autoría

Yo, **Andy Mauricio Cumbicus Jiménez**, con cédula de identidad [**1720900354**], declaro que las ideas, juicios, valoraciones, interpretaciones, consultas bibliográficas, definiciones y conceptualizaciones expuestas en el presente trabajo; así cómo, los procedimientos y herramientas utilizadas en la investigación, son de absoluta responsabilidad de el autor del trabajo de integración curricular. Así mismo, me acojo a los reglamentos internos de la Universidad de Investigación de Tecnología Experimental Yachay.

Urcuquí, Marzo del 2020.

Firmado electrónicamente por:
ANDY MAURICIO CUMBICUS JIMENEZ

Andy M. Cumbicus

CI: 1720900354

Autorización de publicación

Yo, **Andy Mauricio Cumbicus Jiménez**, con cédula de identidad **1720900354**, cedo a la Universidad de Tecnología Experimental Yachay, los derechos de publicación de la presente obra, sin que deba haber un reconocimiento económico por este concepto. Declaro además que el texto del presente trabajo de titulación no podrá ser cedido a ninguna empresa editorial para su publicación u otros fines, sin contar previamente con la autorización escrita de la Universidad.

Asimismo, autorizo a la Universidad que realice la digitalización y publicación de este trabajo de integración curricular en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Urcuquí, Marzo del 2020.

Firmado electrónicamente por:
ANDY MAURICIO CUMBICUS JIMENEZ

Andy M. Cumbicus

CI: 1720900354

Dedication

"Dedicate to Clara & Patricio."

You're the light on my way.

Acknowledgment

Me resulta dificil agradecer en unas lineas a todas las personas que me han acompañado durante estos 6 años. En primer lugar quiero agradecer a mi maravillosa familia. Especialmente, a mis padres Silvio y Teresa que me apoyaron y fueron mi soporte durante todo este tiempo. Agradezco también a mis hermanos Cristian, Patricia, Lola y Silvia por su incondicional cariño y sostén.

Quiero agradecer a mi segunda familia: mis increíbles amigos/as que estuvieron conmigo durante mi etapa universitaria. Gracias Adrián por compartir tus gratos conocimientos y experiencias, y por vivir grandes aventuras conmigo. A Kathy por ser una brillante amiga, y compañera de innumerables andanzas. A Josué y Franklin por ser colegas de academia y de vida. A Mishel, por su valiosa amistad y por compartir tiempo de calidad junto a mí. A los amigos-hermanos que hice durante mi último semestre (los Pibes) Antonio, Carlos, Christian, Eder, Francis, Jerry y Jorge, que convirtieron el último semestre en Yachay en el mejor. Gracias también a mi amigote Fernando, que es un amor de persona. Gracias a los amigos/as de la carrera de matemáticas, especialmente a Guido y Naomi por su compañerismo y por el conocimiento compartido. A San Francis por tan bonita amistad dentro y fuera de una cancha de futbol. Gracias a todos/as vosotros/as mi etapa en la universidad fue grandiosa, maravillosa e inolvidable. Me regalasteis grandes momentos que llenaron mi corazón de felicidad, dicha y paz. Os llevaré siempre conmigo. ¡Gracias totales!

Resumen

Los procesos Gaussianos profundos (DGPs) se representan jerárquicamente mediante una composición secuencial de procesos Gaussianos a prior, y son equivalentes a una red neuronal multicapa (NN) de ancho infinito. Los DGPs son modelos estadísticos no paramétricos y se utilizan para caracterizar los patrones de sistema no lineales complejos, por su flexibilidad, mayor capacidad de generalización, y porque proporcionan una forma natural para hacer inferencia sobre los parámetros y estados del sistema. En este artículo se propone una estructura Bayesiana jerárquica para modelar los pesos y sesgos de la red neuronal profunda, se deduce una formula general para calcular las integrales de procesos Gaussianos con densidades de transferencias no lineles, y se obtiene un núcleo para estimar las funciones de covarianzas. Para ilustrar la metodología se realiza un estudio empírico analizando una base de datos de electroencefalogramas (EEG) para el diagnóstico de la enfermedad de Alzheimer. Adicionalmente, se estiman los modelos DGPs, y se comparan con los modelos de NN para 5, 10, 50, 100, 500 y 1000 neuronas en la capa oculta, considerando dos funciones de transferencia: Unidad Lineal Rectificada (ReLU) y tangenge hiperbólica (Tanh). Los resultados demuestran buen desempeño en la clasificación de las señales. Finalmente, utilizó como medida de bondad de ajuste el error cuadrático medio para validar los modelos propuestos, obteniéndose errores de estimación bajos.

Palabras Clave: Procesos Gausianos Profundos; Enfermedad de Alzheimer; Electroencefalogramas.

Abstract

Deep Gaussian Processes (DGPs) are hierarchically represented by a sequential composition of a prior Gaussian processes and are equivalent to a multi-layer neural network (NN) of infinite width. DGPs are non-parametric statistical models and are used to characterize patterns of complex nonlinear systems, due to their flexibility, greater generalization capacity, and because they provide a natural way to make inferences about the parameters and states of the system. In this article, a hierarchical Bayesian structure is proposed to model the weights and biases of a deep neural network, a general formula is deduced to calculate the integrals of Gaussian processes with non-linear transfer densities, and a kernel is obtained to estimate the covariance functions. To illustrate the methodology, an empirical study is carried out analyzing a database of electroencephalograms (EEGs) for the diagnosis of Alzheimer's disease. Additionally, the DGPs models are estimated, and compared with the NN models for 5, 10, 50, 100, 500, and 1000 neurons in the hidden layer, considering two transfer functions: Rectified Linear Unit (ReLU) and hyperbolic Tangent (Tanh). The results show good performance in the classification of the signals. Finally, the mean square error was used as a goodness of fit measure to validate the proposed models, obtaining low estimation errors.

Keywords: Deep Gaussian Process; Neuronal Networks, Alzheimer Disease; Electroencephalogram.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Supervised learning has received great attention in recent times due to its multiple applications in science, engineering and industry. Among them, classification tasks using Gaussian Processes (GP) have become popular because of the ease with which calculations can be performed in the Gaussian framework, and because of the flexibility with which they can be used for these tasks. In general, deep machine learning systems are computational algorithms that provide powerful modern tools that allow the use of mathematical models including multiple intermediate layers at different levels combined with transfer functions with non-linear structures [1]. Neural networks are machine learning models that have received a lot of attention in recent years due to their success in many real-world applications: they have been used very frequently in filtering content on social networks, in natural language processing, pattern recognition in Big data, tracking objects in sequence of images or videos, face and voice recognition, human mobility, mass information dissemination in networks, in electronic commerce and in classification of relevant information, among many other applications such as in [2], [3] and [4].

A Gaussian process is a generalization of the Gaussian probability distribution. Mathematically, these models may appear to be complex. It is precisely within the framework of the Gaussian process that it unites a sophisticated and consistent view with computational manageability [5]. The constant search for probabilistic models best fits the data in

terms of accuracy. DGPs result from this search, combining Bayesian inference with random processes. In practical terms, DGPs constitute a model used for specific conditions. Being a deep model, it has structural advantages that can improve learning about complex or functional data linked to abstract information extraction. The data needed to train the DGP is directly related to its depth. It is interesting to analyze how this model behaves in practice when the amount of data used is modified.

1.2 Problem statement

Bayesian inference is an alternative to conventional training for neural networks with advantages that include the automatic determination of the appropriate degree of "regularization", the quantification of the uncertainty in predictions, and the possibility of comparison with other models. Consider a deep fully-connected neural network with i.i.d. random parameters. Each scalar output of the network, an affine transformation of the final hidden layer, will be a sum of i.i.d. terms. As we will discuss in detail below, in the limit of infinite width the Central Limit Theorem implies that the function computed by the neural network (NN) is a function drawn from a (GP). In the case of single hidden-layer networks, the form of the kernel of this GP is well known. This correspondence implies that if we choose the hypothesis space to be the class of infinitely wide neural networks, an i.i.d. prior over weights and biases can be replaced with a corresponding GP prior over functions. As noted by [6] and [7], this substitution enables exact Bayesian inference for regression using neural networks. The Deep Gaussian Process (DGP) models can be used for classification and regressions tasks [8] demonstrated that deep ensembles provide an effective mechanism for estimating the marginal distributions. They also investigated priors on the considered functions by defining vague priors on the neural network weights; in addition, they demonstrate properties, generalize the models from a probabilistic perspective, and obtain results equivalent to those reproduced using a GP [5].

In this work, a combined technique of Gaussian processes with neural network models is used, following the approach outlined by: GPs are used to model functional data because they are flexible, robust to outliers, and provide an estimate of calibrated uncertainty.

Deep Gaussian Processes are a generalization of a multi layer neural network viewed as a GP in the limit width. Also, a theoretical proof of the equivalence between GP and NN in width limit. The pertinent mathematical details of the theorem that states this equivalence will be provided. Later, a public dataset will be used to experimentally validate this result. Models of both types will be created and their behavior will be analyzed. In the same way that Roman, et al. (2022) [9] proposed a probabilistic model to classify EGG signals, in this work we also add the theory support of the DGP model giving the mathematical fundamentals.

1.3 Objectives

To design and to provide a supervised learning model based on Bayesian training using Gaussian Process theory. Give the mathematical details about the proposed bayesian hierarchical model step by step. Also, training this model using a dataset to classify Alzheimer's test signals, and measure its performance. In these tasks, two kinds of classification models are used: Gaussian Process and Artificial Neuronal Networks.

1.3.1 General Objective

Demonstrate the exact equivalence between infinitely wide neural networks and Gaussian Process over functions and some assumptions. In the computational context, we will prove the proposed model accuracy with EGG signal classification tasks. In this way, experiments were made with different kinds of wide neural networks to test the hypothesis.

1.3.2 Specific Objectives

- Review the most important theorems which build the mathematical background behind Bayesian training in Deep Gaussian Process.
- To compute the integrals of Gaussian processes for sigmoid functions with non-linear structures, and obtain a kernel to update the covariance functions.
- Testing all models accuracy to prove the equivalence between DGP and NN using different metrics for the goodness of fit.

- To measure the classification of EGG signals to determine the models performance for this kind of tasks.

1.4 Justification

The study of probabilistic models for classification tasks is highly interesting to the scientific community. Although the theoretical development was decades ago, modern computational power allows the training of these models [10]. There are various types of Gaussian process models and architectures. These vary in the number of random variables, the type of adjustment of the hyperparameters, and dimension. Each model can be adjusted to a type of data according to the problem to be solved. One of the great advantages of GPs over traditional machine learning models is Bayesian inference [11]. Therefore, we can include prior information as bayesian theory says. The rest of the models are fitted to the data by identifying patterns in the information held. Some scenarios are consider for model parameters, such as changing the type of weight distribution, activation function, and dataset size. This work proposes a further comparison between several machine learning models and a type of Gaussian Process.

The main contribution of this study is to analyze the DGP performance against the neural networks. The comparison is measured by the accuracy of the models when performing the EEG signal classification task. We include the new parameters for bias and weights of the DGP in order to measure the performance and accuracy. Also, a new formula for estimation of the kernel DGP, under the proposed assumptions, is given in detail and a numeric method is suggested. Later, we review a complete mathematical aspects to explain the exact bayesian equivalence between NN and DGP. In this way, we build an alternative model that can be used in regression and classification tasks, such as in the detection of Alzheimer's diseases.

The results of this research explicitly show the good behavior of the DGP model for this type of problem. In addition, the comparison of the models for each change made is explained in detail. Furthermore, an equivalence between neural networks and Gaussian

processes is detailed in the theoretical part. Mathematical development is given that allows us to see the equivalence of these models under certain conditions. In addition, the theorem that evidences this relationship is proved. In this way, the practical results should be similar, which is what is shown later in this work.

Chapter 2

Theoretical Framework

2.1 Artificial Neural Networks

Mathematically, an artificial neural network can be defined as a directed graph with vertices representing neurons and edges representing connections. The input of each neuron is a function of a weighted sum of the output of all neurons that are connected by their edges to the input. There are many variants of neural networks that differ by their architecture. The simplest of these forms is the forward neural network, which is also known as a is the percetron neural network [12], [13] and [14].

Deep neural networks compose computations performed by many layers. Denoting the output of hidden layers by $h_i^{(l)}$, where $l \in \{1, \ldots, L\}$, $i \in \{1, \ldots, N_l\}$ denote the indices of the neuron within the layer that receives the information from the neurons of the previous layer $h_j^{(l-1)}$, $j \in \{1, \ldots, N_{l-1}\}$. The output j represented by the $j-$th neuron in the output layer, is connected to the input vector x via a biased weighted sum and an non-linear activation function ϕ. The $j-$th component of the network output, $h_j^{(l)}$, is computed as:

$$h_j^{(l)}(x) = \phi\left(z_i^{(l-1)}(x)\right) \tag{2.1}$$

where:

$$z_i^{(l)}(x) = b_i^{(l)} + \sum_{j=1}^{N_{l-1}} W_{ij}^{(l)} h_j^{(l-1)}(x) \tag{2.2}$$

For convenience, the parameters of the neural network are combined into a vector of

parameters $\theta = \left(b^{(1)}, W^{(1)}, \ldots, b^{(L)}, W^{(L)}\right)$ and input data $h^{(0)} = x = (1, x_1, \ldots, x_{N_l})$. Deep neural networks compose computations performed by many layers. The computation for a network with L hidden layers is:

$$\hat{y} = f\left[h^{(L)}\left(z^{(L)}\left(h^{(L-1)}\left(\ldots z^{(2)}\left(h^{(2)}\left(z^{(1)}\left(h^{(1)}\left(z^{(0)}\right)\right)\right)\right)\right)\right)\right] \tag{2.3}$$

When L is large it is called a deep neural network, and each pre-activation function $z^{(L)}(x)$ is typically a linear operation with matrix $W^{(L)}$ and bias $b^{(L)}$, which can be combined with the parameters θ [15], [7] and [16]. An architecture of a feed-forward neural network with L hidden layers is shown in Figure 1.

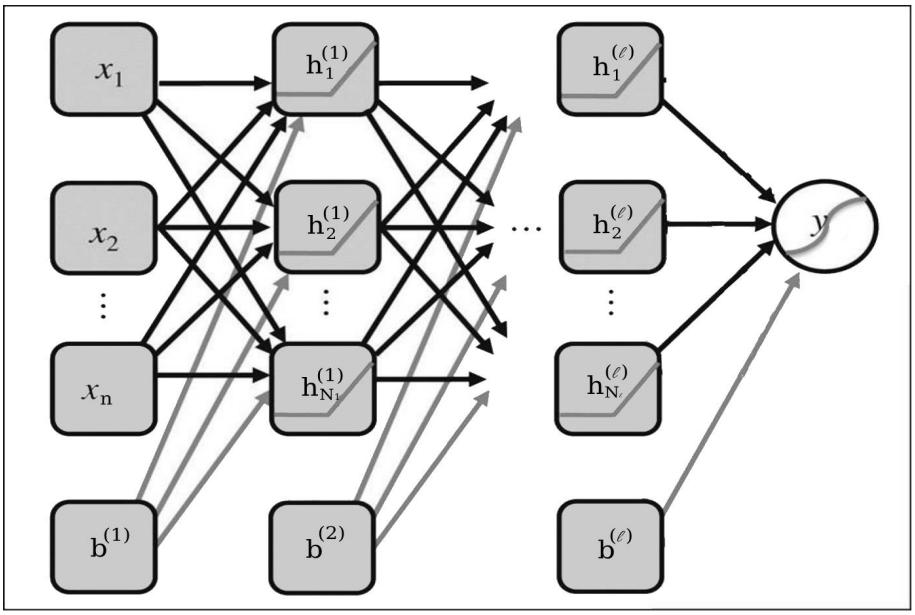

Figure 2.1: An architecture of a feed-forward neural network with l hidden layers.

2.2 Gaussian processes

Assume we have access to a training dataset of n input-output observations

$$\mathbb{D} = \{(x_i, y_i) : i = 1, \ldots, n\} \tag{2.4}$$

y_i is assumed to be a noisy realisation of an underlying latent function $f = f(x)$, i.e.

$$y_i = f(x_i) + \epsilon_i, \quad \epsilon_i \sim N(\mu, \sigma_\epsilon^2) \tag{2.5}$$

where $x_i \in \mathbb{R}^q$ and $y_i \in \mathbb{R}$. The mean of the likelihood is assumed to be input dependent and given a GP prior $\mu = f_i = f(x_i)$.

The interest consists in estimating the function f, which in general is nonlinear. GPs provide a natural way to make inferences about these functions as Rasmussen and Williams explained at [17]. The GP defines a distribution over functions and can be seen as an extension of the multivariate Gaussian distribution. By definition, a stochastic process is a set of random variables $\{f(x) : x \in \mathcal{X}\}$, indexed by a set, $\mathcal{X}$. A GP is a stochastic process such that for any finite set of function evaluations, $f(x) = (f(x_1), \ldots, f(x_n))^T$, where f is multivariate Gaussian distributed. [18] wrote that for any finite set of elements drawn from $\mathcal{X}$, f is a GP described by a mean, $m(.)$, and covariance function, $K(.,.)$, which we write as:

$$f(x) \sim GP\left(m(x), K\left(x, y\right)\right) \tag{2.6}$$

where

$$m(x) = \mathbb{E}\left[f(x)\right], \quad K\left(x_i, x_j\right) = \mathbb{E}\left\{\left[f(x_i) - m(x_i)\right]\left[f(x_j) - m(x_j)\right]\right\} \tag{2.7}$$

$$m : \mathcal{X} \to \mathbb{R}, \quad \textit{and} \quad K(.,.) : \mathcal{X} \times \mathcal{X} \to \mathbb{R} \tag{2.8}$$

Let X_{new} be a matrix with on each row a new input point x_i^{new}, $i = 1, \ldots, n$. To sample a function, we first compute the covariances between all inputs in $X_{new} = (x_1^{new}, \ldots, x_n^{new})$

and collect these in an $n \times n$ matrix:

$$K\left(X_{new}, X_{new}\right) = \begin{pmatrix} k\left(x_1^{new}, x_1^{new}\right) & \dots & k\left(x_1^{new}, x_n^{new}\right) \\ k\left(x_2^{new}, x_1^{new}\right) & \dots & k\left(x_2^{new}, (x_n^{new}\right) \\ \vdots & \vdots & \vdots \\ k\left(x_n^{new}, x_1^{new}\right) & \dots & k\left(x_n^{new}, x_n^{new}\right) \end{pmatrix}$$

Choosing the usual prior mean function $m(x) = 0$ to simplify the matrix, we can then sample values of f at inputs X_{new} from the GP by sampling from a multivariate normal distribution:

$$f_{new} = \left(f\left(x_1^{new}\right), \dots, f\left(x_n^{new}\right)\right)^T$$

The joint distribution$p(f, f_{new})$ is given by:

$$\begin{pmatrix} f \\ f_{new} \end{pmatrix} \sim N\left[\begin{pmatrix} m(X) \\ m(X_{new}) \end{pmatrix}, \begin{pmatrix} K\left(X, X\right) & K\left(X, X_{new}\right) \\ K\left(X_{new}, X\right) & K\left(X_{new}, X_{new}\right) \end{pmatrix}\right]$$

where $K(X, X)$, represents the kernel evaluated at X, $K(X_{new}; X_{new})$ is the covariance matrix between the new points, $K\left(X, X_{new}\right)$ is the covariance matrix between the observed points and the new values, and $K(X_{new}; X)$ is the covariance matrix between the new points and the observed points. The mean function can be any function, but the covariance function must be positive definite, that is

$$\sum_j \sum_i v_i K(x_i, x_j) v_j \geq 0, \quad \textit{for all} \quad v_i,\ x_i \tag{2.9}$$

To complete the specification of the prior we need to specify the mean and covariance functions. Any positive definite covariance function can be chosen; for $K(.,.)$ are the following, see Table 1:

The distribution predicted by the GP can be determined by the conditional rules of mul-

Kernel	$K(x,y)$
Exponential	$\sigma^2 \exp\left(-\frac{\lvert x-y\rvert}{l}\right)$
Squared Exponential	$\sigma^2 \exp\left(-\frac{1}{2}\frac{\lvert x-y\rvert^2}{l^2}\right)$
Matérn	$\sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\lvert x-y\rvert}{l}\right)^{\nu} \kappa_\nu \left(\frac{\lvert x-y\rvert}{l}\right)$
Brownian Motion	$\min(x,y)$

Table 2.1: Covariance functions

tivariate Gaussian distribution:

$$f_{new}|f, X, y \sim N\left(\mathbb{E}\left(f_{new}\right), \mathbb{C}ov\left(f_{new}\right)\right) \tag{2.10}$$

where

$$\mathbb{E}\left(f_{new}\right) = m\left(X_{new}\right) + K\left(X_{new}, X\right) K^{-1}\left(X, X\right)\left[f - m\left(X\right)\right] \tag{2.11}$$

and

$$\mathbb{C}ov\left(f_{new}\right) = K\left(X_{new}, X_{new}\right) - K\left(X_{new}, X\right) K^{-1}\left(X, X\right) K\left(X, X_{new}\right)$$

For a Gaussian likelihood:

$$y_i = f(x_i) + \epsilon_i \quad , \quad \epsilon_i \sim N\left(0, \sigma_\epsilon^2\right) \quad , \quad y|f \sim N\left(f(x_i), \sigma_\epsilon^2 I\right) \tag{2.12}$$

where I is the identity matrix. The noise can be included in the covariance function, as follows:

$$K\left(f(x_i), f(x_j)\right) = K\left(x_i, x_j\right) + \delta_{ij}\sigma_\epsilon^2 \tag{2.13}$$

where δ_{ij} is the Kronecker delta, and σ_ϵ^2 is the noise variance between layers.
The uncertainty is now present in the observations, and the joint distribution over the

unknown data and the known data is augmented in the covariance equation by

$$\begin{pmatrix} f \\ f_{new} \end{pmatrix} \sim N\left(\begin{pmatrix} m\left(X\right) \\ m(X_{new}) \end{pmatrix}; \begin{pmatrix} K\left(X,X\right)+\sigma_{\epsilon}^{2}I & K\left(X,X_{new}\right) \\ K\left(X_{new},X\right) & K\left(X_{new},X_{new}\right) \end{pmatrix}\right)$$

The marginal distribution is given by

$$f_{new}|X_{new},X,f \sim N\left(\mathbb{E}\left(y_{new}\right),\mathbb{C}ov\left(y_{new}\right)\right) \tag{2.14}$$

where

$$\mathbb{E}\left(y_{new}\right)=m\left(X_{new}\right)+K\left(X_{new},X\right)\left(K\left(X,X\right)+\sigma_{\epsilon}^{2}I\right)^{-1}\left[y-m\left(X\right)\right]$$

and

$$\mathbb{C}ov\left(y_{new}\right)=K\left(X_{new},X_{new}\right)-K\left(X_{new},X\right)\left(K\left(X,X\right)+\sigma_{\epsilon}^{2}I\right)^{-1}K\left(X,X_{new}\right)$$

In this case the integrals required to infer a posterior, $p\left(f_{new}|X_{new},X,f\right)$, are tractable.

2.3 Deep Neural Networks and Gaussian Processes

In this section, we consider fully-connected ANNs with layers numbered from $l=0$ (imput) to $l=L-1$ (output), each containing, $N_0,\ldots,N_{L-1}$ neurons, and with a Lipschitz, twice differentiable nonlinearity activation function $\phi:\mathbb{R}\to\mathbb{R}$, with bounded second derivative. For each $x\in\mathbb{R}^{d_{in}}$ denote the imput to the network ($x=(x_1,\ldots,x_{d_{in}})$), and $z^{(l)}\in\mathbb{R}^{d_{out}}$ denote its output. We use $z_i^{(l)}(x),h_i^{(l)}(x)$ to represent the pre- and post-activation functions at layer l with input x, also, let $h_i^{(0)}=x$. The parameters consist of the connection matrices $W_{ij}^{(l)}\in\mathbb{R}^{N_l\times N_{l+1}}$ and bias vectors $b_i^{(l)}\in\mathbb{R}^{N_{l+1}}$ for $l=0,\ldots,L-1$; which are independent and randomly selected, with zero mean and variances $\frac{\sigma_w^2}{N_l}$, and σ_b^2, respectively.

Now we are going to establish the relationship between a single-hidden layer neural networks, and Gaussian processes. Suppose that $z_j^{(l)}(x)$ is a Gaussian process with mean and

covariance functions $\mu^{(l)}(x)$, $K^{(l)}(x, x')$, respectively, i.e.

$$z_j^{(l)}(x) = \begin{pmatrix} z_j^{(l)}(x) \\ z_j^{(l)}(x') \end{pmatrix} \sim GP\left(\mu^{(l)}(x), K^{(l)}\left(x, x'\right)\right)$$

where

$$\mu^{(l)}(x) = \mathbb{E}\left\{\mathbf{z}_j^{(l)}(x)\right\} = \begin{pmatrix} \mu(x) \\ \mu(x') \end{pmatrix}$$

and

$$K = K^{(l)}\left(x, x'\right) = \mathbb{C}ov\left\{z_j^{(l)}(x), z_j^{(l)}(x')\right\} = \begin{pmatrix} K^{(l-1)}(x, x) & K^{(l-1)}(x, x') \\ K^{(l-1)}(x', x) & K^{(l-1)}(x', x') \end{pmatrix}$$

The $i-$th component of the network output, $z_i^{(1)}$, is computed as:

$$z_i^{(1)}(x) = b_i^{(1)} + \sum_{j=1}^{N_1} W_{ij}^{(1)} h_j^{(1)}(x), \quad h_j^{(1)}(x) = \phi\left(b_j^{(0)} + \sum_{k=1}^{d_{in}} W_{ij}^{(0)} x_k\right) \tag{2.15}$$

Note that there is a dependency on the input data vector x, and also since the weight and bias are considered i.i.d., the post-activations $h_j^{(1)}$ and $h_k^{(1)}$ are independent for $j \neq k$. Also, $z_i^{(1)}(x)$ is a sum of i.i.d terms, it follows from the Central Limit Theorem that in the limit of infinite width $N_1 \to \infty$, $z_i^{(1)}(x)$ will be Gaussian distributed, Lee, et. al (2018).

Suppose that $z_j^{(l)}(x)$ is a Gaussian processes, i.i.d for every j, and that $h_j^{(l)}(x)$, also are independent and identically distributed. Then after $l-1$ steps, the recurrence relation for a feedforward network is defined as

$$z_i^{(l)}(x) = b_i^{(l)} + \sum_{j=1}^{N_l} W_{ij}^{(l)} h_j^{(l)}(x), \quad h_j^{(l)}(x) = \phi\left(z_j^{(l-1)}(x)\right) \tag{2.16}$$

Prior on weights:

$$b_j^{(l)} | \sigma_b^2 \sim N\left(0, \sigma_b^2\right), \quad W_{ij}^{(l)} | \Sigma_W \sim N\left(0, \Sigma_W\right) \tag{2.17}$$

Prior on hyperparameters:

$$\sigma_b^2|\alpha,\beta \sim IG\left(\alpha,\beta\right) \quad \Sigma_W|\nu,R \sim IG\left(\nu,R\right) \tag{2.18}$$

Note, $z_i^{(l)}(x)$ is a sum of i.i.d. random terms so that, as $N_l \to \infty$, any finite collection $\{z_i^{(l)}(x), \quad l = 1,\ldots,L\}$ will have joint multivariate Gaussian distribution, i.e,

$$\mathbf{z}_i^{(l)}(x) \sim GP(\mu^{(l)}(x), K^{(l)}\left(x,x'\right)),\ \mathbf{z}_i^{(l)}(x) = \left(z_i^{(l)}(x_1),\ldots,z_i^{(l)}(x_n)\right) \tag{2.19}$$

A general equation is now established to approximate the covariance for a bivariate Gaussian process:

$$\begin{aligned} K^{(l)}\left(x,x'\right) &= \mathbb{E}\left\{\left[z_i^{(l)}(x) - \mathbb{E}\left(z_i^{(l)}(x)\right)\right]\left[z_i^{(l)}(x') - \mathbb{E}\left(z_i^{(l)}(x')\right)\right]\right\} \\ &= \mathbb{E}\left\{\left[b_i^{(l)} + \sum_{j=1}^{N_l} W_{ij}^{(l)} h_j^{(l)}(x)\right]\left[b_i^{(l)} + \sum_{j=1}^{N_l} W_{ij}^{(l)} h_j^{(l)}(x')\right]\right\} \\ &= \sigma_b^2 + \Sigma_W \sum_{j=1}^{N_l} \mathbb{E}\left[h_j^{(l)}(x) h_j^{(l)}(x')\right] \\ &= \sigma_b^2 + \Sigma_W \sum_{j=1}^{N_l} \mathbb{E}\left[\phi\left(z_j^{(l-1)}(x)\right)\phi\left(z_j^{(l-1)}(x')\right)\right] \end{aligned} \tag{2.20}$$

where the calculation of the expectation is a two dimensions Gaussian integral:

$$\begin{aligned} \mathbb{E}\left(\phi\left(z_j^{(l-1)}(x)\right)\phi\left(z_j^{(l-1)}(x')\right)\right) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} &\phi\left(z_j^{(l-1)}(x)\right)\phi\left(z_j^{(l-1)}(x')\right) \\ &\times p\left(z_j^{(l-1)}(x), z_j^{(l-1)}(x')\right) \\ &\times dz_j^{(l-1)}(x) dz_j^{(l-1)}(x') \end{aligned} \tag{2.21}$$

Since:

$$\begin{pmatrix} z_j^{(l)}(x) \\ z_j^{(l)}(x') \end{pmatrix} \sim GP\left(\mu^{(l)}(x), K^l\left(x,x'\right)\right) \tag{2.22}$$

then

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\phi\left(z_j^{(l-1)}(x)\right)\phi\left(z_j^{(l-1)}(x')\right)\frac{1}{2\pi|K|^{\frac{1}{2}}}$$
$$\times\exp\left\{-\frac{1}{2}\left(z_j(x)^{(l-1)}-\mu(x),\quad z_j^{(l-1)}(x')-\mu(x')\right)K^{-1}\begin{pmatrix}z_j(x)^{(l-1)}-\mu(x)\\ z_j^{(l-1)}(x')-\mu(x')\end{pmatrix}\right\}$$
$$\times dz_j^{(l-1)}(x)dz_j^{(l-1)}(x')$$

Let

$$K^{-1}=\left(\sqrt{K}^T\right)^{-1}\left(\sqrt{K}\right)^{-1} \tag{2.23}$$

Consider the following transformation:

$$\begin{pmatrix}\xi_{i_1}\\ \xi_{i_2}\end{pmatrix}=\left(\sqrt{K}\right)^{-1}\begin{pmatrix}z_j(x)-\mu(x)\\ z_j(x')-\mu(x')\end{pmatrix}\Rightarrow\left(\sqrt{K}\right)\begin{pmatrix}\xi_{i_1}\\ \xi_{i_2}\end{pmatrix}=\begin{pmatrix}z_j(x)-\mu(x)\\ z_j(x')-\mu(x')\end{pmatrix}$$

To simplify the notation, the following variable change is made:

$$z_j(x)=z_j(x)^{(l-1)},\quad z_j(x')=z_j(x')^{(l-1)} \tag{2.24}$$

$$\sqrt{K}\xi_{i_1}=z_j(x)-\mu(x)\Rightarrow z_j(x)=\sqrt{K}\xi_{i_1}+\mu(x)\Longleftrightarrow\xi_{i_1}=\frac{z_j(x)-\mu(x)}{\sqrt{K}}\sim N(0,1)$$

and

$$\sqrt{K}\xi_{i_2}=z_j(x')-\mu(x')\Rightarrow z_j(x')=\sqrt{K}v+\mu(x')\Longleftrightarrow\xi_{i_2}=\frac{z_j(x')-\mu(x')}{\sqrt{K}}\sim N(0,1)$$

The Jacobian of the transformation is:

$$J=\begin{pmatrix}\frac{\partial z_j(x)}{\partial\xi_{i_1}} & \frac{\partial z_j(x)}{\partial\xi_{i_2}}\\ \frac{\partial z_j(x')}{\partial\xi_{i_2}} & \frac{\partial z_j(x')}{\partial\xi_{i_2}}\end{pmatrix}=\begin{pmatrix}\sqrt{K} & 0\\ 0 & \sqrt{K}\end{pmatrix}\Rightarrow|J|=|K|^{\frac{1}{2}} \tag{2.25}$$

then

$$\begin{aligned}
&\frac{1}{2\pi|K|^{\frac{1}{2}}}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\phi\left(\sqrt{K}\xi_{i_1}+\mu(x)\right)\phi\left(\sqrt{K}\xi_{i_2}+\mu(x')\right)\\
&\times\frac{1}{2\pi}\exp\left(-\frac{1}{2}\left(\xi_{i_1}^2+\xi_{i_2}^2\right)\right)|J|d\xi_{i_1}d\xi_{i_2}\\
&=\frac{1}{\sqrt{2\pi}}\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\phi\left(\sqrt{K}\xi_{i_1}+\mu(x)\right)\phi\left(\sqrt{K}\xi_{i_2}+\mu(x')\right)\\
&\times\exp\left(-\frac{1}{2}\xi_{i_1}^2\right)\exp\left(-\frac{1}{2}\xi_{i_2}^2\right)d\xi_{i_1}d\xi_{i_2}\\
&=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\phi\left(\sqrt{K}\xi_{i_1}+\mu(x)\right)\exp\left(-\frac{1}{2}\xi_{i_1}^2\right)d\xi_{i_1}\\
&\times\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\phi\left(\sqrt{K}\xi_{i_2}+\mu(x')\right)\exp\left(-\frac{1}{2}\xi_{i_2}^2\right)d\xi_{i_2}
\end{aligned}$$

The approximation of the last integral depends on the choice of the sigmoid function $\phi(.)$.

In the case of deep neural networks, the transfer function $\phi(z)$ is a bounded function where all moments are bounded. Then we can apply Central Limit Theorem to show that the stochastic process is a Gaussian process, Christopher (1996) [6] and Lee (2018) [7].

Cho and Saul [19] developed a new family of covariance functions which allows computing the correlation between two vectors $x, z \in \mathbb{R}^{N_l}$. They define the $n-$th order arc-cosine kernel function via the integral representation:

$$K^{(l)}(x,z)=2\int\frac{1}{(2\pi)^{\frac{N_l}{2}}}\exp\left(-\frac{\|w\|^2}{2}\right)\Theta(w\,.x)\Theta\left(w\,.z\right)\left(w\,.x\right)^l\left(w\,.z\right)^l dw, \tag{2.26}$$

where $\Theta\left(t\right)=\frac{1}{2}\left(1+sign(t)\right)$ denote the Heaviside step function. The integral representation (2.26) allows the kernel of covariance functions to be positive definite, and that the dependence between x y z, can be written as:

$$K^{(l)}\left(x,z\right)=\frac{1}{\pi}\|x\|^l\|z\|^l J_l(\theta) \tag{2.27}$$

where all the angular dependence is captured by $J_l(\theta)$. The angular dependence is given by:

$$J_l(\theta) = (-1)^l (\sin\theta)^{2l+1} \left(\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\right)^l \left(\frac{\pi-\theta}{\sin\theta}\right) \tag{2.28}$$

In particular, when $l = 0$, we have the angle between x and z. When $l > 0$, the angular dependence is more complicated. Some terms of $J_l(\theta)$, are shown:

$$J^0(0) = \theta, \quad J^1(\theta) = \sin\theta + (\pi-\theta)\cos\theta,$$
$$J^2(\theta) = 3\sin\theta\cos\theta + (\pi-\theta)\left(1 + 2\cos^2\theta\right)$$

The arc-cosine kernel for $l = 0$, is represented by:

$$K^{(0)} = 1 - arccos^{-1}\left(\frac{xz}{\|x\|\|z\|}\right) \tag{2.29}$$

Neural network models are strongly related to the kernel of functions defined in (2.26), when considering the inner product between the different outputs of the neural network as:

$$\phi(x)\phi(z) = \sum_{i=1}^{m} \Theta(w_i.x)\Theta\left(w_i.z\right)\left(w_i.x\right)^l\left(w_i.z\right)^l \tag{2.30}$$

where w_i denote $i-$th row of the weight matrix W and m is the number of output units. In the limit, it can be seen that the equation (2.30) is equivalent to (2.26), Cho and Saul (2009):

$$\lim_{m\to\infty} \frac{2}{m}\phi(x)\phi(z) = K^{(l)}(x,z) \tag{2.31}$$

Also, Cho and Saul [19] proved that for a vector of inputs $x = (1, x_1, \ldots, x_{N_l})$, their characteristics can be mapped by means of a nonlinear transformation $\phi(x)$, using kernel functions:

$$K^{(l)}(x,z) = \phi\left(\phi\left(\ldots\phi(x)\right).\phi\left(\ldots\phi(z)\right)\right) \tag{2.32}$$

The iterated equation (2.32) mimics a multilayer neural network, for example for a one-layer neural network, $K(x,z) = \phi(x)\phi(z)$. Cho and Saul (2009) define a recursive kernel through a new mapping of features through compositions such as $\phi(\phi(x))$. In the case of a linear kernel $K(x,z) = xz$, the composition is $\phi(\phi(x)) = \phi(x) = x$, and for homogeneous polynomial kernels $K(x,z) = (xz)^{N_l}$; the composition is:

$$K(x,z) = \phi(\phi(x)).\phi(\phi(z)) = (\phi(x)\phi(z))^{N_l} = \left((x.z)^{N_l}\right)^{N_l} = (x.z)^{N_l^2} \tag{2.33}$$

Then we consider the composition of l layers based on the iterate defined in (2.32), applying a mathematical induction the inductive step is given by:

$$K^{(l)}(x,z) = \frac{1}{\pi}\left[K^{(l-1)}(x,x)K^{(l-1)}(z,z)\right]^{\frac{l}{2}} J_l\left(\theta^{(l)}\right) \tag{2.34}$$

where $\theta^{(l)}$ is the angle between x and y in the feature space:

$$\theta^{(l)} = \cos^{-1}\left\{\frac{K^{(l)}(x,z)}{\left[K^{(l)}(x,x)K^{(l)}(z,z)\right]^{\frac{1}{2}}}\right\} \tag{2.35}$$

Recently, a linear rectified function $ReLU(t) = \max(0;t)$, was successfully used in neural networks as it carries the neuron signal better. ReLUs have the desirable property that they do not require input normalization, as Krizhevsky in [20]. To compute the given integral (2.21), a rectified linear sigmoidal can be used, Hazan in [21], which results in an analytical kernel given by

$$K_{ReLU}\left(x_i,x_j\right) = \frac{\|x_i\|\|x_j\|}{\pi}\sin\left(cos^{-1}\left(\rho_{ij}\right)\right) + \left(\pi - cos^{-1}\left(\rho_{ij}\right)\right)\rho_{ij}$$

where

$$\rho_{ij} = \frac{\langle x_i,x_j\rangle}{\|x_i\|\|x_j\|}, \quad \langle x_i,x_j\rangle = \int\int \phi(x_i)\phi(x_j)p\left(x_i,x_j\right)dx_i dx_j$$

To compute the entries of $K(.,.)$, let $\phi(t) = ReLU(t)$,

$$\begin{pmatrix} z_1' \\ z_2' \end{pmatrix} \sim N(0, K), \quad 0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad K = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$$

then

$$\mathbb{E}_{z_1', z_2'}\left(\phi\left(z_1'\right)\phi\left(z_2'\right)\right) = h(\sigma_1, \sigma_2, \rho) = \frac{\sigma_1\sigma_2}{\pi}\sin\left(\cos^{-1}(\rho)\right) + \rho\left(\pi - \cos^{-1}(\rho)\right)$$

and $K_{ReLU}^{(1)}$ is given by

$$\begin{pmatrix} h\left(\sqrt{1+\alpha}\|x_i\|, \sqrt{1+\alpha}\|x_i\|, \frac{\alpha}{1+\alpha}\right) & h\left(\sqrt{1+\alpha}\|x_i\|, \sqrt{1+\alpha}\|x_j\|, \frac{\alpha}{1+\alpha}\frac{\langle x_i, x_j\rangle}{\|x_i\|\|x_j\|}\right) \\ h\left(\sqrt{1+\alpha}\|x_i\|, \sqrt{1+\alpha}\|x_j\|, \frac{\alpha}{1+\alpha}\frac{\langle x_i, x_j\rangle}{\|x_i\|\|x_j\|}\right) & h\left(\sqrt{1+\alpha}\|x_j\|, \sqrt{1+\alpha}\|x_j\|, \frac{\alpha}{1+\alpha}\right) \end{pmatrix}$$

Iterating successively, we obtain:

$$K_{ReLU}^{(2)}\left(x_i, x_j\right) = \mathbb{E}_{z_1, z_2}\left(\phi(z_1)\phi(z_2)\right), \quad \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \sim N(0, K)$$

is a recursive equation of h with the appropriate parameters, see Hazan[21].

On the other hand, using GP prior functions allows Bayesian inference to be made precisely to obtain predictions and estimate the uncertainty in deep neural network models. The estimation of the parameters is not required through training based on a gradient-type algorithm.

Under the Bayesian statistical approach, the weights and biases of the network are generated following a probability distribution $p(\mathbf{W}|\theta)$, where ($\mathbf{W} = (W, b)$, represents the weights and bias, and $\theta = (\alpha, \beta, \nu, R) \sim p(\theta)$ represents the hyperparameters, which can be integrated.

$$p(\mathbf{W}) = \int p(\mathbf{W}|\theta)p(\theta)d\theta \tag{2.36}$$

Given a training sample $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, where $\mathbf{x} = (x_1, \ldots, x_n)^T$ and $\mathbf{y} = (y_1, \ldots, y_n)^T$ denote the input and output of the network, respectively. Suppose now that we want to make a prediction with a test data x_{new} using priors over funtions rather than weights $\mathbf{z}(x) = (z_1, \ldots, z_n)$ restricted to input values $\mathbf{x}$. Then

$$\begin{aligned} p\left(z_{new}|\mathcal{D}, x_{new}\right) &= \int p\left(z_{new}|\mathbf{z}, \mathbf{x}, x_{new}\right) p\left(\mathbf{z}|\mathcal{D}\right) d\mathbf{z} \\ &= \frac{1}{p(\mathbf{y})} \int p\left(z_{new}, \mathbf{z}|x_{new}, \mathbf{x}\right) p\left(\mathbf{y}|\mathbf{z}\right) d\mathbf{z} \end{aligned} \tag{2.37}$$

where $\mathbf{y} = (y_1, \ldots, y_n)^T$ are the targets on the training set, $p(\mathbf{y})$ is a marginal likelihood, and $p\left(\mathbf{y}|\mathbf{z}\right)$ corresponds to observation noise. We will assume a noise consisting of a Gaussian $(y|z \sim N(0, \sigma_\epsilon^2))$.

The importance of choosing priors over functions implies that $z_1, \ldots, z_n, z_{new}$ are generated from a Gaussian processes

$$\begin{pmatrix} z \\ z_{new} \end{pmatrix} \sim GP \left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} K\left(\mathcal{D}, \mathcal{D}\right) & K\left(x_{new}, \mathcal{D}\right) \\ K\left(\mathcal{D}, x_{new}\right) & K\left(x_{new}, x_{new}\right) \end{pmatrix} \right]$$

Then the integral in (2.37) can be obtained exactly, since the marginal distribution is:

$$z_{new}|\mathcal{D}, x_{new} \sim N\left(\mu_{post}, K_{post}\right) \tag{2.38}$$

where

$$\mu_{post} = K\left(x_{new}, \mathcal{D}\right)\left(K\left(\mathcal{D}, \mathcal{D}\right) + \sigma_\epsilon^2 I_n\right)^{-1} y$$

and

$$K_{post} = K\left(x_{new}, x_{new}\right) - K\left(x_{new}, \mathcal{D}\right)\left(K\left(\mathcal{D}, \mathcal{D}\right) + \sigma_\epsilon^2 I_n\right)^{-1} K^T(x_{new}, \mathcal{D})$$

where I_n is the $n \times n$ identity. The predicted distribution of $z_{new}|\mathcal{D}, \mathbf{x}$ is clearly determined.Deep neural network training works using a Bayesian approach. The covariance function is determined by choosing a prior Gaussian process. In this case, the model depends on the depth, non-linearity of the transfer function, weights and biases.

Another recently work from [22] gives a new proof for fully connected neural networks equivalence to a Gaussian process. The author shows that equivalence works well under random weights and biases, and where input dimension, output dimension, and depth are kept fixed, while the hidden layer widths tend to infinity.

2.3.1 GP-NN Theorem

Theorem 1: We consider a fully connected network with the form of 2.2 using ReLU activation functions with the following equation

$$z_i^{(l)}(x) = b_i^{(l)} + \sum_{j=1}^{N_{l-1}} W_{ij}^{(l)} h_j^{(l-1)}(x) \tag{2.39}$$

where:

$$h_j^{(l)}(x) = \phi\left(z_i^{(l-1)}(x)\right) \tag{2.40}$$

a distribution parameters given by:

$$b_j^{(l)}|\sigma_b^2 \sim N\left(0, \sigma_b^2\right), \quad W_{ij}^{(l)}|\Sigma_W \sim N\left(0, \Sigma_W\right) \tag{2.41}$$

and,

$$\sigma_b^2|\alpha, \beta \sim IG\left(\alpha, \beta\right) \quad \Sigma_W|\nu, R \sim IG\left(\nu, R\right) \tag{2.42}$$

For $l = 1, ..., D$ hidden layers

Then there exist strictly increasing width functions $k_{l:\mathbb{N}\mapsto\mathbb{N}}$ such that $K_1 =_1 (N), \ldots, K_D = K_D(N)$, and for any countable input set $(x[i])_{i=1}^{\infty}$, the distribution of the output of the network converges in distribution to a Gaussian process as $n \to \infty$ [23].

Proof

We have to prove that $z^{(K+1)}(x[i])_{i=1}^{\infty}$ converges weakly to a Gaussian process when $N \to \infty$. Take a arbitrary finite collection of the inputs $(x[i])_{i\in I}$ of the a finite subset I of $\mathbb{N}$.

In order to continue with the proof, we need the following auxiliary result:

Lemma 1 (Normal recursion). If the activations of a previous layer are normally distributed with moments:

$$\mathbb{E}\left[z_i^{(l-1)}(x)\right] = 0$$
$$\mathbb{E}_i^{(l-1)}(x) z_j^{(l-1)}\left(x'\right) = \delta_{i,j} K\left(x, x'\right)$$

Then under the recursion 2.3.1 and as $N \to \infty$ the activations of the next layer converge in distribution to a normal distribution with moments

$$\mathbb{E}\left[z_i^{(l)}(x)\right] = 0$$
$$\mathbb{E}\left[z_i^{(l)}(x) z_j^{(l)}\left(x'\right)\right] = \delta_{i,j}\left[\Sigma_w \mathbb{E}_{(\epsilon_1,\epsilon_2)\sim\mathcal{N}(0,K)}\left[\phi\left(\epsilon_1\right)\phi\left(\epsilon_2\right)\right] + \sigma_b^2\right]$$

where K is a 2×2 matrix containing the input covariances.

Proposition 1 Let $\varepsilon > 0$, and $x[1], \ldots, x[n] \in \mathbb{R}^M$. If $H_k = 2^{H_{k+1}^2}$ for $k = 1, \ldots, D-1$, then for H_D sufficiently large, we have

$$d\left(z^{(D+1)}(\mathbf{x}), Z(\mathbf{x})\right) \leq \varepsilon,$$

where $Z(\mathbf{x})$ is a mean-zero multivariate normal random variable.

So, for every x[i], we can build a $z^{(D+1)}(x)$ with a multivariate normal distribution because each input is multivariate normal distribution by previous lemma. Analogously, for each layer by Central Limit Theorem and the previous lemma we can set multivariate normal distribution for all z_i. Moreover when $N \to \infty$ we can say that H_D sufficiently large where N_l represents the width for some $l-$layer. So, for the metric d defined as follows:

$$d\left(x[i], x[i]'\right) = \sum 2^{-i} \min\left(1, \left|x[i] - x[i]'\right|\right) \quad \forall x[i], x[i]' \in \mathbb{R}^{\mathbb{N}}$$

and using Proposition 1 we get that there exist strictly increasing functions z_i. Then the output variables for the D-1 layer converges weakly to a multivariate gaussian distribution which means that $(z_i^{(D+1)}(x))$ with $i \in I$ is a Gaussian Process where the covariance matrix

associated to the proposed Gaussian Process in 2.20.

2.4 Electrocefalogram at Alzheimer's diseases

Alzheimer's diseases are very common in society. One way to detect them is by using an Electroencephalogram (EGG) test. An Electroencephalogram records brain activity in electrical signals received by sensors placed in a helmet. The signals are recorded by a machine and plotted for later analysis. Currently, an EGG represents a diagnostic test with high precision and an affordable price. However, the complexity and lack of knowledge about human brain behavior cause uncertainty about this type of disease. For example, an early diagnosis of Alzheimer's must be treated correctly. New mechanisms are currently being used to improve short-term detection [24] and [25].

In the area of machine learning, models have also been developed for the detection of Alzheimer's. A classification task can determine whether or not a patient has the disease. However, the input data is not usually the pure signal since an EEG is a function of time. Some models, such as recurrent neural networks or classical models, can work with this type of data. In this work, instead, a transformation of the time series to a frequency domain is made [26]. There are several methods for data transformation, and here we have used the following:

Higuchi Fractal Dimension (HFD)

The Higuchi Fractal Dimension (HFD) is the slope value of a real value function or time series. This approximation is obtained by fitting the function through the least squares method. To compute the HFD [27] of an N-sample data sequence $x(1), x(2), ..., x(N)$, the data set is divided into a klength sub-data set as,

$$x_k^m : x(m), x(m+k), x(m+2k), \ldots, x\left(m + \left[\frac{N-m}{k}\right] \cdot k\right) \tag{2.43}$$

Where [] is Gauss' notation, k is constant, and m $= 1, 2, \ldots,$ k. The length $L_m(k)$ for each sub-data set is then computed as,

$$L_m(k) = \left\{ \left[\frac{\left[\frac{N-m}{k}\right]}{\sum_{i=1}^{k} |x(m+ik) - x(m+(i-1)\cdot k)|} \right] \frac{N-1}{\left[\frac{N-m}{k}\right] \cdot k} \right\} /k \tag{2.44}$$

The mean of $L_m(k)$ is computed to find the HFD as,

$$HFD = \frac{1}{K} \sum_{M=1}^{K} L_m(k) \tag{2.45}$$

Lempel Ziv Complexity (LZC)

[28] defines LZC a complexity metric related to the number of distinct substrings and the rate of occurrence in a given string. There are several works where this metric is used to solve problems in test signs mainly related with EGG signals. Also, this technique has many method to used it. The same method that [29] provided is used for this work.

The flag to be analyzed is changed into grouping whose components are some images. The twofold arrangement is basic to build: the information values underneath or break even with the cruel have the image "1" and the values over the cruel have the image "0". This algorithm gives the number of distinct patterns contained in the given finite sequence S = s1, s2, ..., sn [30] and [31]. The calculation of c(n) (Lempel-Ziv complexity) proceeds on diagram.

This method uses comparison and accumulation so the computation of c(n) is easy to calculate.

The Lempel-Ziv of the totally random sequence of length n consisting of two different symbols with equal probabilities is

$$b(n) = \frac{n}{\log_2(n)}$$

If we divide the complexity of the sequence by the complexity b(n) of the random sequence, we get the normalized LempelZiv C(n), which does not depend on the length of the sequence

when n is large

$$C(n) = \frac{c(n)}{b(n)}$$

Power Spectral Density (PSD)

Decomposes the signal of the function in terms of the frequencies that make up that signal. There are some kinds of descompositions depend on the type of signal using Fourier analysis. When the time of the decomposition domain is finite, it is possible to calculate the density between the time and the energy of the signal. This decomposition is known as the energy spectral density [32].

In math notation

Let $\{y(t); t = 0, \pm 1, \pm 2, ..\}$ be a discrete-time data sequence of random variables with:

$$\mathbb{E}\{y(t)\} = 0 \quad , \tag{2.46}$$

The covariance function of $r(k)$ is defined as:

$$r(k) = \mathbb{E}\{y(t)y * (t-k)\}. \tag{2.47}$$

The PSD is defined as the Discrete-time Fourier Transform of the covariance sequence [33]:

$$\phi(w) = \sum_{k=-\infty}^{\infty} r(k)e^{-iwk}. \tag{2.48}$$

It is the Fourier transform of the biased estimate of the autocorrelation sequence. The periodogram for a signal x_n is defined as:

$$\hat{P}(f) = \frac{\Delta t}{N}\left|\sum_{n=0}^{N-1} x_n e^{-i2\pi fn}\right|^2, \tag{2.49}$$

with $-1/2\Delta t < f \leq 1/2\Delta t$, where Δt is the sampling interval, N is the length of the signal and f is the frequency in Hz [34].

Relative Power (RP)

The idea is to transform the EEG signals into a power spectrum using Fourier transforms. The relative power value is a ratio between the selected and full frequency ranges [35].

The relative power of each given band/sum of power from 1 to 45 Hz was calculated by

$$RP\left(f_1, f_2\right) = \frac{P\left(f_1, f_2\right)}{P(1,45)} \times 100\% \tag{2.50}$$

where $P(\cdot)$ indicates the power, $RP(\cdot)$ indicates the relative power, and f_1, f_2 indicate the low and high frequency, respectively.

The ratios of power for different frequency bands in each electrode was computed for possible pairs of frequency bands. The relative power for each band and the ratios of power for different frequency bands were averaged in each region.

Sample Entropy (SE)

It is a logarithmic function that relates two consecutive points in the time sequence [36]. As [37] describes this method, an general definition of Sample Entropy is given here. Let EEG signal $x(n)$:

- The partial correlation sums is:

$$C_r^m(i) = \frac{1}{N-m-1} \operatorname{count}[\{n\} \mid n \quad \neq i \& \max_{i=1,m} |x_{n+1} - x_{i+1}| < r \Bigg] \tag{2.51}$$

where N - size of the calculation window, m - number of space dimension, n - current number of the element, r-radius of the hypersphere [37].

- Averaging of the correlation sums:

$$\theta^m(r) = \frac{1}{N-m} \sum_{i=1}^{N-m} C_r^m(i) \tag{2.52}$$

- Calculation of the Sample Entropy:

$$SE(m, r, N) = \ln \frac{\theta^m(r)}{\theta^{m+1}(r)} \tag{2.53}$$

Chapter 3

State of the Art

GPs are used to model operational data because they are flexible, robust to outliers, and estimate calibrated uncertainty. The data output will be modeled by a multivariate GP called Deep Gaussian Processes (DGP), which is a generalization of a multilayer neural network viewed as a GP in the limit width. Currently, there is several researches about DGP as a probabilistic model and using bayesian inference for predictive tasks[38] .

Salimbeni and Deisenroth in [39] used an inference algorithm variational doubly stochastic algorithm that does not force independence between layers. Also, they demonstrated that a DGP model could be used effectively for many data points. They provide strong empirical evidence of the inference scheme for DGPs and show that they work well in practice in both classification and regression.

Zhao in [40] used a state-space model for the DGP regression. They constructed a DGP by placing a GP prior transformed into the length and magnitude scales at each hierarchy level. They used a posterior maximum estimation procedure based on filtering algorithms and demonstrated the performance using non-stationary synthetic and gravitational wave signals. Other work about Deep Gaussian Process was in [41] where they designed a model to approximate expectation propagation. A new method using backpropagation algorithm for learning was proposing by the authors in the Bayesian context. In classification tasks, there is also some important contribution by the use of Deep Gaussian Process. For instance, Jayashree [42] use a DGP model for Text classification, but they

fond some limitations of their model such as over-fitting and model architecture complexity.

Gordon and Izmailov [43] showed that deep ensembles effectively estimate the marginal distributions. They also investigated priors on the considered functions by defining vague priors on the neural network weights. In addition, they demonstrate properties, generalize the models from a probabilistic perspective, and obtain results equivalent to those reproduced using a GP. Lee [7] demonstrated an equivalence between infinitely wide deep networs and GPs, developed computationally efficient methods to compute the covariance function of GPs, and connected GPs with the theory of signal propagation in random neural networks. Investigations in the same direction highlights the work of: Yang and Schoenholz [44], Matthews [23], Novak [45], Garriga-Alonso [46], Agrawal [15], Damianou and Lawrence [5], Khan [47], Hazan and Jaakkola [21].

On the other hand, there are several investigations and applications focused on the classification of electroencephalogram signals. In Gao[48], Gupta[49], Acharya[50], Lun[51] and Zou[52] used convolutional neural network models classify signals previously transformed into elliptical signals. Using this methodology, machine learning models got an accuracy about 90% in the data testing. Another interesting methodology is the one used in Demir [53] where a 2D network input is formed to form a graph and then classified using a machine learning model. Other techniques have been developed over the years to detect diseases associated with brain activity. For example, in Kumar [54], Motamedi [55] and [56] where they investigate the importance of the biological network through various models of neural networks.

Machine learning models mentioned above are some of the most popular in use today [57], [58], [59], [60], [61], [62] and [63]. However, there are a variety of other techniques for classifying EEG signals such as Gupta[64] which used PCA technique for Classification of Seizure and Seizure-Free EEG Signals. In Kulkarni [65], McBride [66], RuizGmez [67] and Buscema [68] they used Support Vector Machine (SVM) models which is a artificial intelligence model used to classification tasks. SVM works well for binary classification task like our current problem. Other model used is K-nearest neighbor mentioned at Bablani [69],

Nuru [70] and Li [71]. An interest investigation could be compare all these model against our Gaussian Process model. However, this is off limits of current research.

Probabilistic models are also used in the context of classifying EEG signals [72],[73], [74] and [75]. In particular, in Faul [76] and Owen [77] they design Gaussian processes for detecting diseases related to brain activity. Advantageously, GPs are parametric models that allow training models with small data sizes. On the other hand, Sarishvili [78] and Chiappa [79] they use other probabilistic models to analyze and classify EEG signals. Mainly, these systems are based on the Gaussian state-space model, where the bayesian analysis plays a fundamental role in including a priori information of the models. Bayesian models are inside of probabilistic but some important contributions need to be highlighted, for example [80], [81], [82], [83] and [84]. Moreover, in Seixas[85] created bayesian network to the diagnosis of dementia and alzheimers disease. Also, there are general works for modeling of imaging, genetics and diagnosis which show the several applications of these models such as in [86], [87], [88], [89], [90], [91] and [92].

Chapter 4

Methodology

4.1 Phases of Problem Solving

The main goal is to use the dataset mentioned above to classify the positive and negative cases of Alzheimer's disease. In this sense, we use the classification models detailed in Table (3). Each file was used to train the GP and NN models independently. That is, in total, results are obtained from 7 classification models (GP, NN = 5, NN = 10, NN = 50, NN = 100, NN = 500, and NN = 1000). It is proposed to compare the performance of GPs against neural networks changing some parameters such as network width, data-set size and activation function. In each model, the size of the training data for the learning process was changed; that is, n = 75, n = 100, n = 250 and n = 436 observations are taken to train the model for each of the data-set sizes. Similarly, all models are tested with the ReLu and Tanh activation functions. Also, in neural networks, the width of the network is changed to compare its performance against the Gaussian process. In Figure (2) there is a diagram that explains the experiment for each data-set.

4.1.1 Description of the Problem

Once the equivalence between INN and GP was theoretically demonstrated, an algorithm was created to verify this equivalence computationally. The first thing was to show that an infinite neural network (in width) is a Gaussian process. To do this, a data-set of Alzheimer's signs was taken to train 7 classification models. Six of them are neural networks with different width sizes in the hidden layers. The remaining model is a Gaussian process

that is Bayesian trained as mentioned in the previous section. The metrics used to validate the experiment are the mean square error and the classification accuracy. We want to demonstrate that the accuracy of the neural network with the highest number of neurons in the hidden layers is similar to the accuracy of the GP. Similarly, the mean square error of both models should be similar. Finally, the third part of the experiment is to repeat the previous steps varying the amount of input data to avoid over fitting the models.

4.1.2 Data Description

The dataset used corresponds to a previous investigation of Alzheimer's disease (AD) detection. AD is a progressive and irreversible brain disorder, which causes memory problems, slowly destroying it until losing the ability to develop simple tasks. An non-invasive method to study AD uses Electroencephalograms (EEGs) that register the electrical activity of the brain. However, the EEG raw signals are difficult to classify directly. Martínez (2021)[29] proposed to extract five significant features related to the main effects of AD on EEG signals. Therefore, for this study we only considered the five significant features mentioned in Table 4.1.

Notation	Feature Name
HFD	Higuchi Fractal Dimension
LZC	Lempel Ziv Complexity
PSD	Power Spectral Density
RP	Relative Power
SE	Sample Entropy

Table 4.1: Features extracted from the EEGs to identify the AD effects.

EEG signal analysis is usually performed using linear methods such as Fast Fourier Transform. However, alternative nonlinear models have been developed for analyzing EEG signals using chaos theory in recent years [93]. In this method, the time dimension is changed to the frequency dimension to explain in the same way the signal behavior. In the bellow sections, some of both methods used in this work for extract features as motioned in Table 4.1 are explained in details.

4.1.3 Analysis of the Problem

The experimental goal is to validate the performance of the proposed method in the diagnosis of Alzheimer's disease. Different experiments were executed, varying the size of the input set, to view the accuracy difference when the size of the input increase. As reference point, this section compares the results of the proposed method with multi-layer neuronal networks.

4.1.4 Algorithm Design

For the development of the algorithm, python 3 was used in Jupyter Notebook. Using popular machine learning libraries like Pandas, Skylearn and Tensorflow. The specifications of all the libraries can be found in the .1. Mainly, the algorithm consists of two parts. The first is where the functions that estimate the kernel under certain parameters established in this research are defined. On the other hand, the main function where the training of the models is executed and the results are stored.

Also, in this experiment the data of the significant features was cleaned up by randomly removing data to leave the same number of labels for each class in the dataset Anexo .1. In total, 514 observations were considered, of which 85% were for training data and the rest 15% for testing.

4.1.5 Testing

The predictive performance of Deep Gaussian Process (DGP) and Neural Networks (NN) models are evaluated using the following metric, Mean Squared Error (MSE), defined by:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4.1}$$

where $\hat{y}_i$ indicates the predicted value, y_i indicates the actual values, and n is the number of predictions. The metirc range is in $[0, \infty)$ and lower values indicate better performance. It can be seen in Figure 5.1 for Relative Power that in terms of error the GP with the Relu activation function is the best whereas the Neural Network with 5-width is better in Tanh. Also, the error results on the other features are shown in Table 5.1.

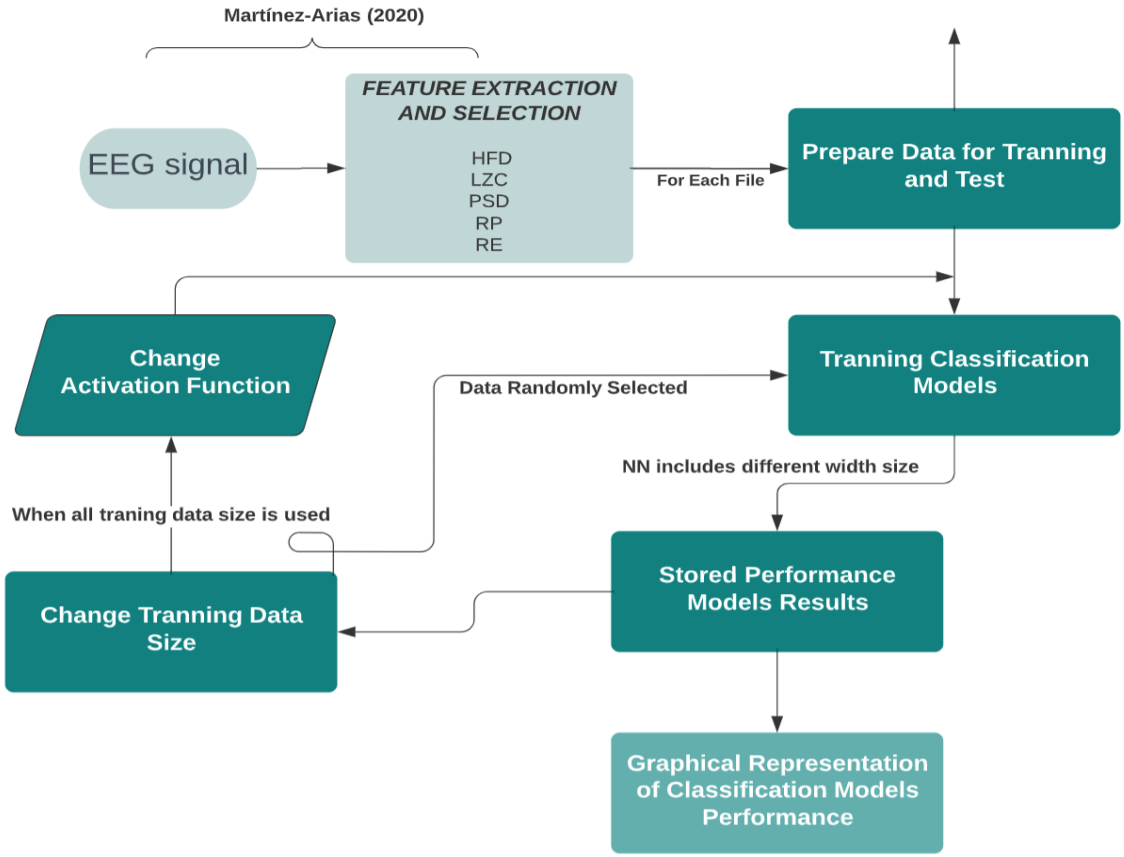

Figure 4.1: Experiment Description Diagram.

The metrics for all features and models were recorded in Table 5.1. In this case only was considered the results with the highest dataset size that is 436.

Chapter 5

Results and Discussion

In this section, only part of the results obtained from the performance of all the models has been included. The following details the accuracy and error of the models, but only for the Relative power technique and a data volume of 250 observations.

Datase	Model (ReLU)	Test Accuracy	Test MSE	Model (tanh)	Test Accuracy	Test MSE
LZC	GP	0.935	0.0564	GP	0.948	0.1092
	NN-10	0.935	0.0508	NN-10	0.883	0.0784
	NN-1000	0.961	0.0404	NN-1000	0.948	0.0594
RP	GP	0.987	0.0215	GP	0.974	0.0221
	NN-10	0.935	0.0414	NN-10	0.961	0.0248
	NN-1000	0.987	0.0243	NN-1000	0.974	0.0299
HFD	GP	1.000	0.0125	GP	0.987	0.0153
	NN-10	0.961	0.0243	NN-10	1.000	0.0152
	NN-1000	1.000	0.0114	NN-1000	1.000	0.0178
PSD	GP	0.974	0.0246	GP	0.987	0.0245
	NN-10	0.961	0.0244	NN-10	0.961	0.0215
	NN-1000	0.987	0.0157	NN-1000	0.974	0.0207
SE	GP	1.000	0.0125	GP	0.987	0.0153
	NN-10	0.961	0.0243	NN-10	1.000	0.0152
	NN-1000	1.000	0.0114	NN-1000	1.000	0.0178

Table 5.1: A Performance comparison between DGP and Neural Networks for all features.

The results for the Relative Power feature is shown in Figure 5.6. Clearly the model that use the proposed Gaussian Process has the best test accuracy. This figure shows that the Neural Network models behave better with Tanh activation function than Relu on this feature while in the case of the GP there is no significant difference. Additionally, the

performance results on the other features are shown in Table 5.1.

Datase	Model (ReLU)	Test Accuracy	Test MSE	Model (tanh)	Test Accuracy	Test MSE
LZC	NN-50	0.909	0.091	NN-50	0.948	0.052
	NN-100	0.935	0.065	NN-100	0.935	0.065
	NN-500	0.935	0.060	NN-500	0.961	0.040
RP	NN-50	0.961	0.061	NN-50	0.961	0.050
	NN-100	0.948	0.044	NN-100	0.974	0.037
	NN-500	0.948	0.044	NN-500	0.961	0.033
HFD	NN-50	0.982	0.034	NN-50	0.974	0.024
	NN-100	0.980	0.025	NN-100	0.987	0.034
	NN-500	0.987	0.023	NN-500	0.987	0.023
PSD	NN-50	0.974	0.0246	NN-50	0.987	0.0245
	NN-100	0.961	0.0244	NN-100	0.961	0.0215
	NN-500	0.987	0.0157	NN-500	0.974	0.0207
SE	NN-50	1.000	0.0125	NN-50	0.987	0.0153
	NN-100	0.987	0.024	NN-100	0.987	0.015
	NN-500	0.987	0.011	NN-500	0.987	0.017

Table 5.2: Performance and error results for Neural Networks.

The results for neural network models with 50, 100, and 500 neurons for Relative Power are similar to those seen in the table. However, it should be noted accuracy for these models is slightly higher than the 5- and 10-neuron models, but lower than the 1000-neuron model. This makes sense as depth in theory, indicates greater accuracy. However, models with more neurons are more computationally complex as they consume more resources and take more time to train.

On the other hand, in most models, the tanh activation function provides better accuracy. In the GP this relationship is not maintained because there is evidence that the ReLu function fits better.

5.1 Model Error Results

The metric used to measure the error of the models was the mean square error was 4.1. It is set as -0.1 for the wrong class (healthy) and 0.9 for the correct class (infected). However, it was necessary to transform the label classes into a regression problem before measuring the error.

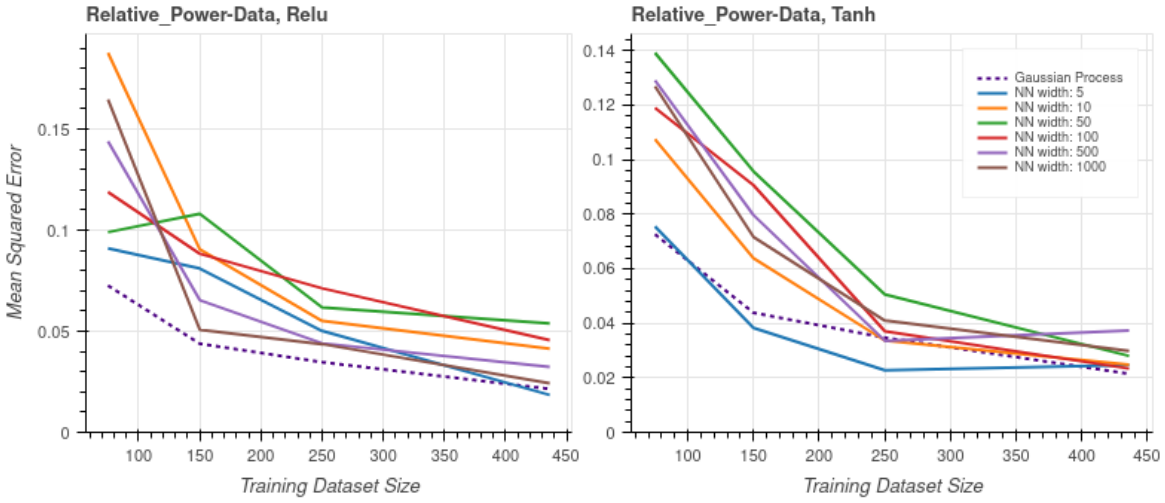

Figure 5.1: Relative Power Error Results

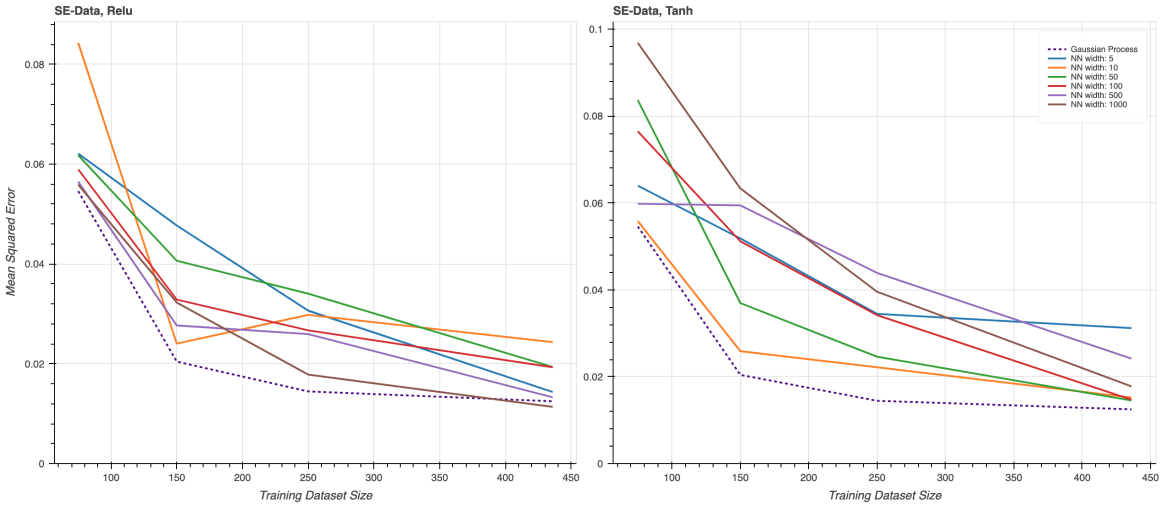

Figure 5.2: Sample Entropy Error Results

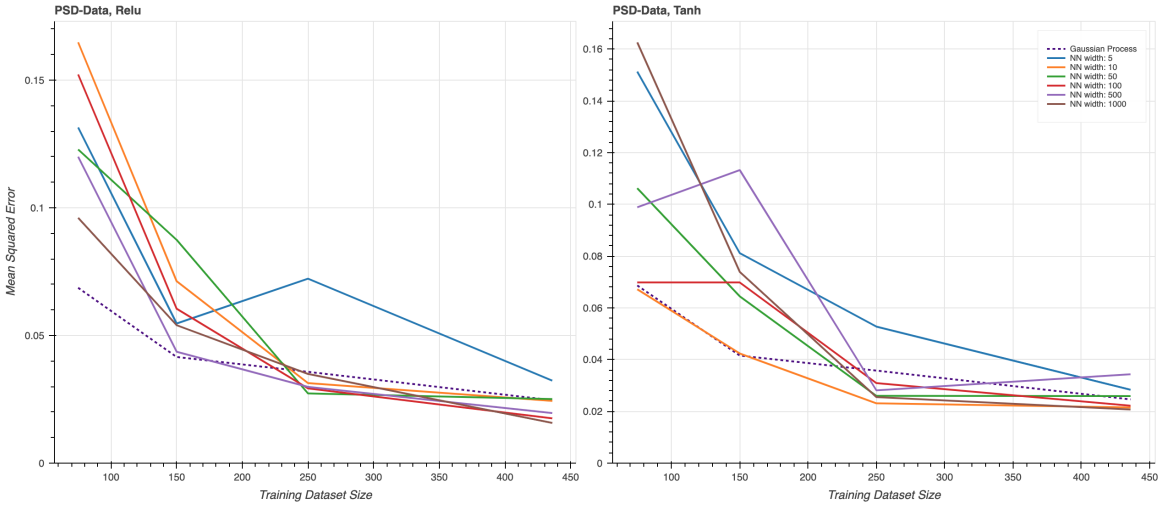

Figure 5.3: Power Spectral Density Error Results

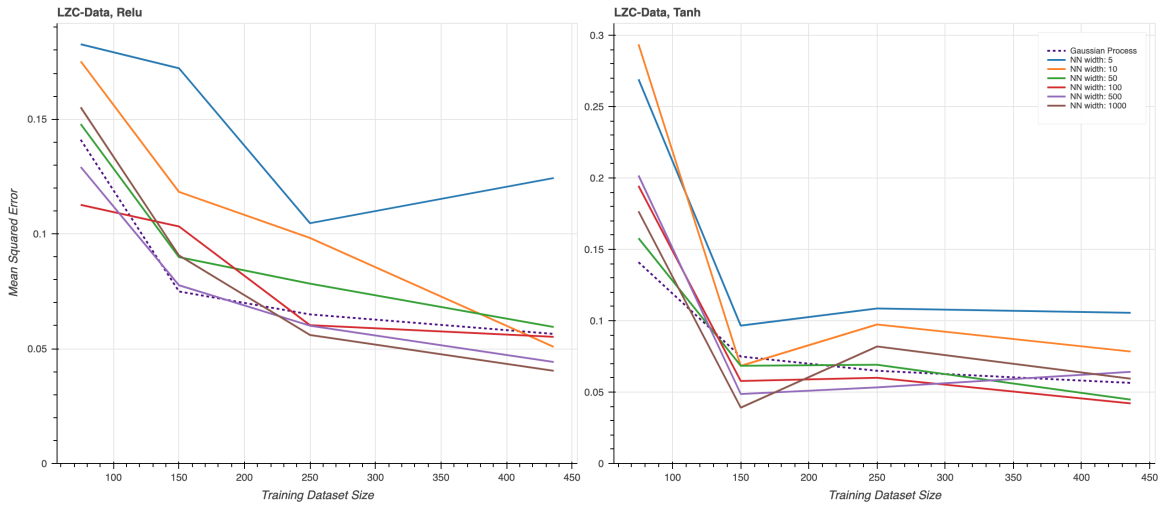

Figure 5.4: Lempel-Ziv Complexity Error Results

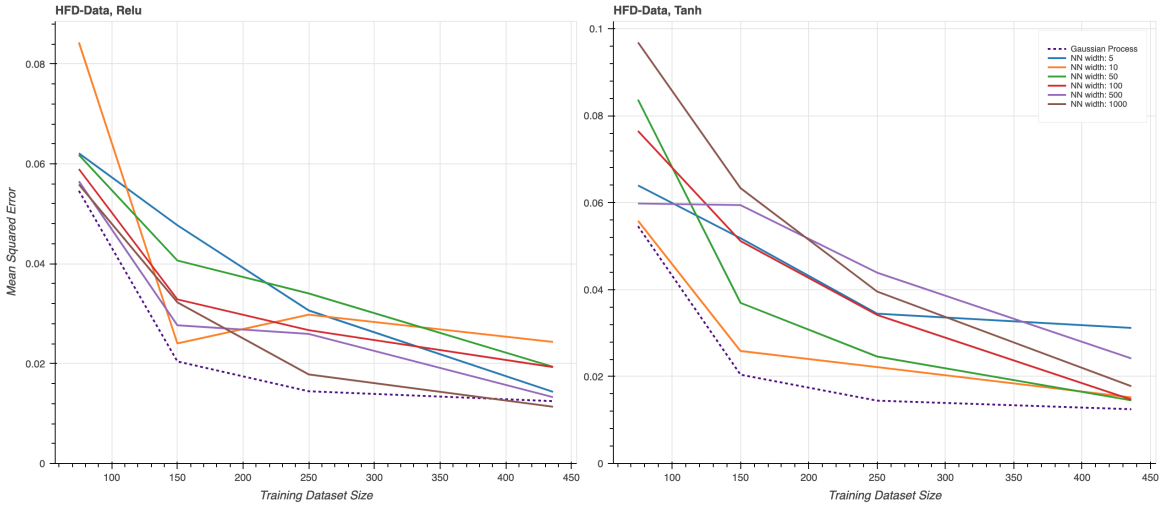

Figure 5.5: Higuchi Fractal Dimension Error Results

All models behave similarly when the number of training observations is increased, which indicates that with more observations, there should be less error. As we can see in the figures 5.1, 5.2, 5.3, 5.4 and 5.5, the model with the least error is the GP when all the observations are taken. Also, note that the error of the 1000-neuron network is close to the GP.

5.2 Model Performance Results

In the same way as the model error results, the classification accuracy is measured by transforming the classes into numerical values. The difference between classifying an infected is measured by comparing how many false positives and false negatives with those correctly classified. Analogously to what happens in a confusion matrix. In Fig 5.6, 5.9 and 5.10, the DGP fits better than the rest of the models, improving its accuracy as the size of the training data increases. On the other hand, neural networks maintain close to 95% accuracy on all features. In addition, as in the DGP, the neural networks improve when training with more data. The 5-neuron model has the worst performance, although it is close to 90

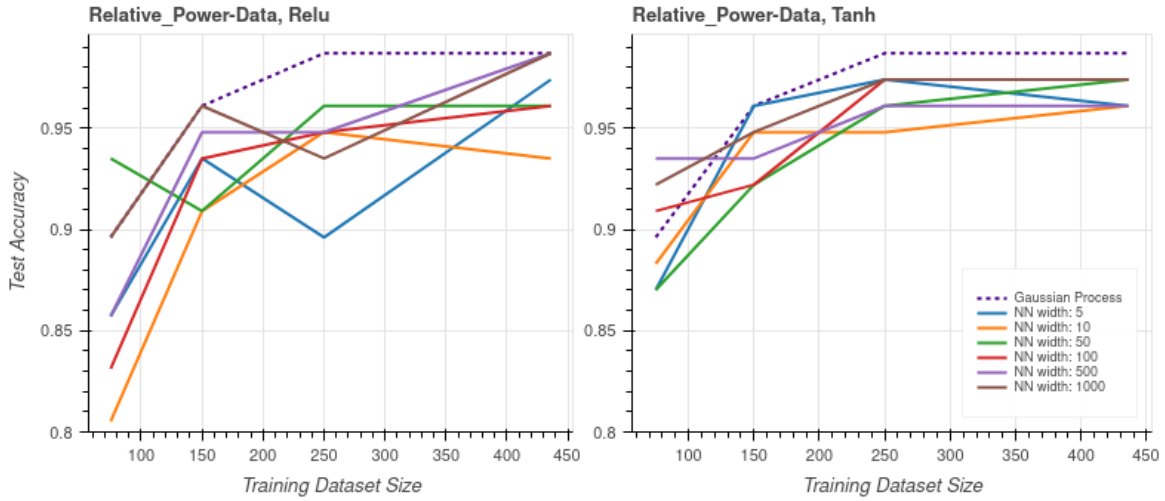

Figure 5.6: Relative Power Performance Results

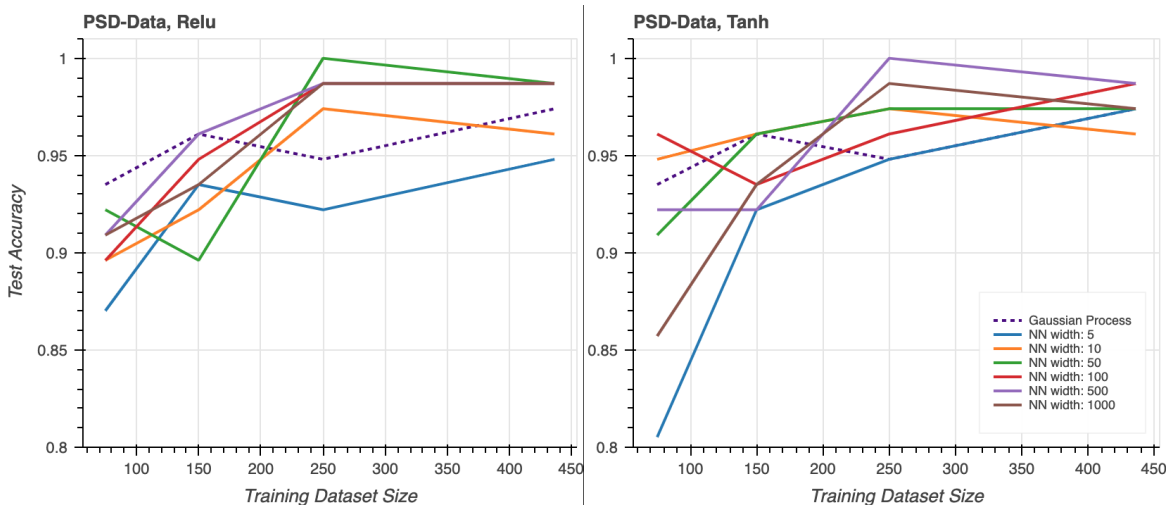

Figure 5.7: PSD Performance Results

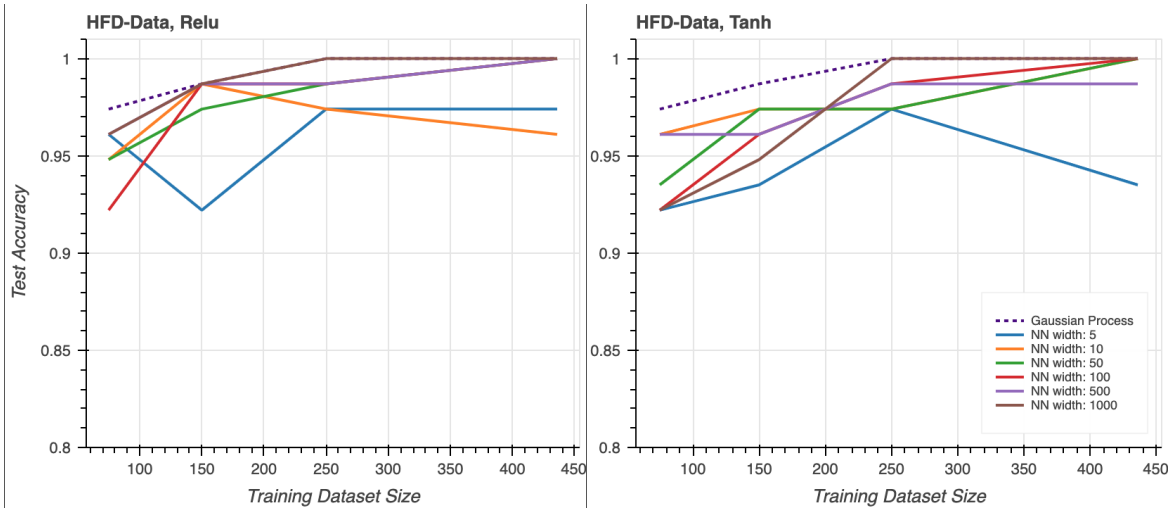

Figure 5.8: HFD Performance Results

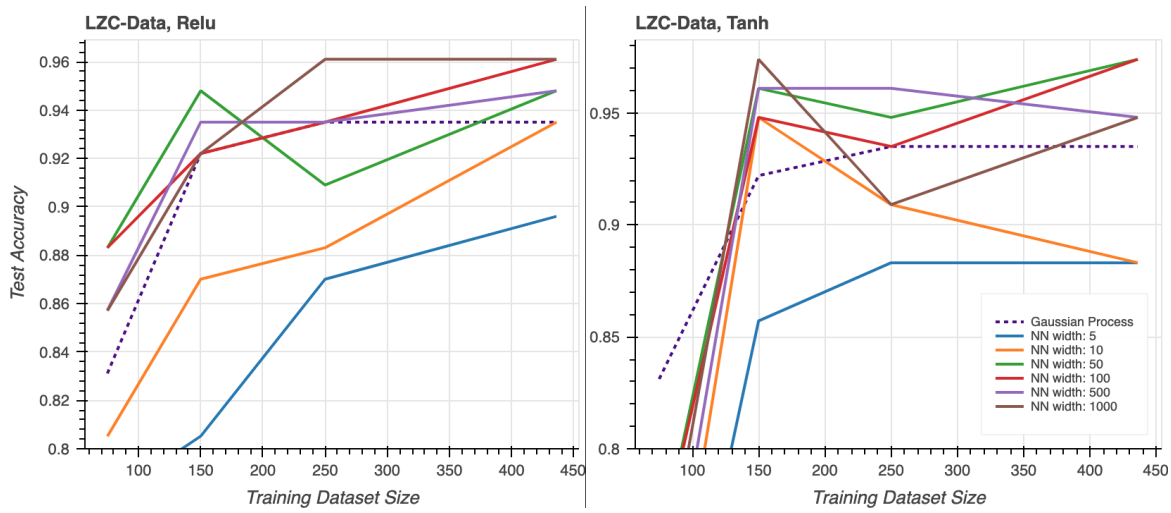

Figure 5.9: HFD Performance Results

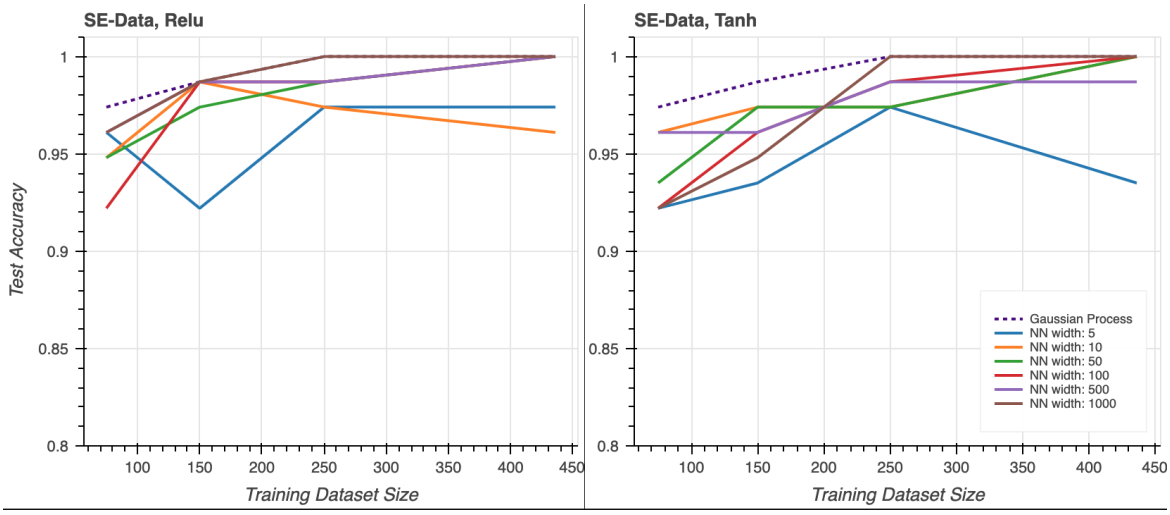

Figure 5.10: SE Performance Results

5.3 Interpretation

Both metrics indicate that the best model is the DGP. The classification accuracy is high because it is a simple binary classification task for these models. Activation functions do not directly influence the behavior of models. However, taking different dataset sizes affects the models' accuracy and error. The model closest to the Gaussian process is the network of 1000 neurons, as the theory says. This behavior occurs with all features, but using the full training dataset. Curiously, in Fig 5.1 the 5-neuron network is the model with the least error.

Chapter 6

Conclusions

Using a non stochastic gradient-based training we classified EEG Signals for the diagnosis of Alzheimer's Diseases to test the behavior of our GP, that uses a hierarchical Bayesian structure to model the weights and biases of a neural network, and compare it with neural networks varying their widths. A general formula was derived to evaluate the resulting integrals of Gaussian processes with non-linear transfer functions, and a kernel was obtained to update the covariance functions. The proposed methodology was applied to the classification of EEG signals for the diagnosis of Alzheimer's disease, considering five data sets and estimating the models varying size of the samples. In this study, it was shown that DGP can be used in supervised learning and classification tasks. As mentioned in Lee et al. (2018), we check that the GP behaves as a neural network with infinite number of neurons in the hidden layers. In general, all test accuracies were very high because the data-set consist on a limited number of patients and the work done by Martínez-Arias (2020) of feature extraction and selection.

We gave a mathematical proof in this work that demonstrates the equivalence between a connected neural network and a Gaussian process. Under certain random conditions, a new formula was given for the computation of the GP kernel. However, for kernel estimation, we must use a numerical method. In our experiments, we realize that inverse gamma distribution hyper-parameters fixed well for models learning process. The bias and weight variances are randomly choice using this hyper-parameters and it contributes to evaluate the model performance. The results show that GP classify well, and sometimes even better

than neural networks models.

For future researches, we will expected to prove our GP model to classify other data types such as crude data or images. Crude ECG signal data could be a interesting scenario to use GP model in order to classify positive cases of Alzheimer's disease. Also, analyze what role play distribution hyper-parameters in the model with other data types. Finally, the integral I_z compute in section 4 may be used to estimate the kernel using a different method that we proposed. This could be another research to see which alternative method works better, and to compare the GP performance results. We also can measure other model parameters to compare the performance of the two types of models: DGP and NN. The time it takes for the models to learn from the data. In the same way, add more data to analyze its behavior. Furthermore, the DGP can be compared with other artificial intelligence models such as SVM, K-Nearest Neighbors, or other types of NN. On the other hand, it also uses other datasets, such as classifying images, data with more classes, or regression tasks.

Bibliography

[1] K. Choudhary, B. DeCost, C. Chen, A. Jain, F. Tavazza, R. Cohn, C. W. Park, A. Choudhary, A. Agrawal, S. J. L. Billinge, E. Holm, S. P. Ong, and C. Wolverton, "Recent advances and applications of deep learning methods in materials science," *npj Computational Materials*, vol. 8, no. 1, Apr. 2022. [Online]. Available: https://doi.org/10.1038/s41524-022-00734-6

[2] I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," *SN Computer Science*, vol. 2, no. 3, Mar. 2021. [Online]. Available: https://doi.org/10.1007/s42979-021-00592-x

[3] ——, "Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions," *SN Computer Science*, vol. 2, no. 6, Aug. 2021. [Online]. Available: https://doi.org/10.1007/s42979-021-00815-1

[4] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, Mar. 2021. [Online]. Available: https://doi.org/10.1186/s40537-021-00444-8

[5] A. C. Damianou and N. D. Lawrence, "Deep gaussian processes," 2012. [Online]. Available: https://arxiv.org/abs/1211.0358

[6] C. K. I. Williams, "Computing with infinite networks," in *Proceedings of the 9th International Conference on Neural Information Processing Systems*, ser. NIPS'96. Cambridge, MA, USA: MIT Press, 1996, p. 295–301.

[7] J. Lee, Y. Bahri, R. Novak, S. Schoenholz, J. Pennington, and J. Sohl-Dickstein, "Deep neural networks as gaussian processes," 10 2017.

[8] A. G. Wilson and P. Izmailov, "Bayesian deep learning and a probabilistic perspective of generalization," 2020. [Online]. Available: https://arxiv.org/abs/2002.08791

[9] C. A. I. S. . F.-D. R. Román, K., "Procesos gausianos profundos y redes neuronales infinitas para el análisis de señales eeg en la enfermedad de alzheimer." *Revista De Matemática: Teoría Y Aplicaciones*, vol. 29, no. 2, pp. 289–312, 2022.

[10] J. Schmidt, M. R. G. Marques, S. Botti, and M. A. L. Marques, "Recent advances and applications of machine learning in solid-state materials science," *npj Computational Materials*, vol. 5, no. 1, Aug. 2019. [Online]. Available: https://doi.org/10.1038/s41524-019-0221-0

[11] M. Kuss, "Gaussian process models for robust regression, classification, and reinforcement learning," Ph.D. dissertation, Technische Universität Darmstadt, 2006.

[12] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, and H. Shah, "Wide amp; deep learning for recommender systems," 2016. [Online]. Available: https://arxiv.org/abs/1606.07792

[13] M. Awad and R. Khanna, *Deep Neural Networks*, 01 2015, pp. 127–147.

[14] A. Radhakrishnan, M. Belkin, and C. Uhler, "Wide and deep neural networks achieve optimality for classification," 2022. [Online]. Available: https://arxiv.org/abs/2204.14126

[15] D. Agrawal, T. Papamarkou, and J. Hinkle, "Wide neural networks with bottlenecks are deep gaussian processes," 2020. [Online]. Available: https://arxiv.org/abs/2001.00921

[16] N. Cedeño, G. Carillo, M. J. Ayala, S. Lalvay, and S. Infante, "Analysis of chaos and predicting the price of crude oil in ecuador using deep learning models," in *Communications in Computer and Information Science*. Springer International

Publishing, 2021, pp. 318–332. [Online]. Available: https://doi.org/10.1007/978-3-030-90241-4_25

[17] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. The MIT Press, 2006.

[18] R. M. Neal, "Priors for infinite networks." University of Toronto, 1996.

[19] Y. Cho and L. Saul, "Kernel methods for deep learning," in *Advances in Neural Information Processing Systems*, Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, Eds., vol. 22. Curran Associates, Inc., 2009. [Online]. Available: https://proceedings.neurips.cc/paper/2009/file/5751ec3e9a4feab575962e78e006250d-Paper.pdf

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[21] T. Hazan and T. S. Jaakkola, "Steps toward deep kernel methods from infinite neural networks," *CoRR*, vol. abs/1508.05133, 2015. [Online]. Available: http://arxiv.org/abs/1508.05133

[22] B. Hanin, "Random neural networks in the infinite width limit as gaussian processes," 2021. [Online]. Available: https://arxiv.org/abs/2107.01562

[23] A. G. d. G. Matthews, M. Rowland, J. Hron, R. E. Turner, and Z. Ghahramani, "Gaussian process behaviour in wide deep neural networks," 2018. [Online]. Available: https://arxiv.org/abs/1804.11271

[24] N. Kulkarni and V. Bairagi, "Electroencephalogram based diagnosis of alzheimer disease," 01 2015.

[25] A. Tsolaki, D. Kazis, I. Kompatsiaris, V. Kosmidou, and M. Tsolaki, "Electroencephalogram and alzheimer's disease: Clinical and research approaches," *International journal of Alzheimer's disease*, vol. 2014, p. 349249, 04 2014.

[26] Q. Ma, M. Wang, L. Hu, L. Zhang, and Z. Hua, "A novel recurrent neural network to classify EEG signals for customers' decision-making behavior prediction in brand extension scenario," *Frontiers in Human Neuroscience*, vol. 15, Mar. 2021. [Online]. Available: https://doi.org/10.3389/fnhum.2021.610890

[27] A. H. H. Al-nuaimi, E. Jammeh, L. Sun, and E. Ifeachor, "Higuchi fractal dimension of the electroencephalogram as a biomarker for early detection of alzheimer's disease," vol. 2017, 07 2017.

[28] M. Aboy, R. Hornero, D. Abásolo, and D. Alvarez, "Interpretation of the lempel-ziv complexity measure in the context of biomedical signal analysis," *IEEE transactions on bio-medical engineering*, vol. 53, pp. 2282–8, 12 2006.

[29] M. A. P. Nathaly, "Study of artificial intelligence models applied to the analysis of electroencephalograms in alzheimer's disease," 2020.

[30] B. G. D. L. L.-G. . P. S. De Felice, C., "Abnormal oral vascular network geometric complexity in ehlers-danlos syndrome." *Oral surgery, oral medicine, oral pathology, oral radiology, and endodontics*, vol. 98(4), pp. 429–4, 02 2004.

[31] X.-S. Zhang, R. Roy, and E. Jensen, "EEG complexity as a measure of depth of anesthesia for patients," *IEEE Transactions on Biomedical Engineering*, vol. 48, no. 12, pp. 1424–1433, 2001. [Online]. Available: https://doi.org/10.1109/10.966601

[32] S. Sobana Rani, "Power spectral density in communication systems." 12 2016.

[33] R. L. M. P. Stoica, *Spectral analysis of signals.* New Jersey: Prentice Hall, 2005.

[34] M. Zainuddin Lubis and H. Manik, *Signal processing for power spectral density (PSD)*, 03 2016, pp. 15–25.

[35] K. Ko, H.-C. Yang, and K.-B. Sim, "Emotion recognition using eeg signals with relative power values and bayesian network," *International Journal of Control, Automation and Systems*, vol. 7, pp. 865–870, 10 2009.

[36] G. Jiang, S.-Z. Fan, M. Abbod, H.-H. Huang, J.-Y. Lan, F.-F. Tsai, H.-C. Chang, Y.-W. Yang, F.-L. Chuang, Y.-F. Chiu, K.-K. Jen, J.-F. Wu, and J.-S. Shieh, "Sample entropy analysis of eeg signals via artificial neural networks to model patients' consciousness level based on anesthesiologists experience," *BioMed Research International*, vol. 2015, pp. 1–8, 02 2015.

[37] Y. Zhivolupova and O. Tcvetkov, "The method for increasing of eeg signal sample entropy stability and its application for human state monitoring," vol. 776, 04 2017, pp. 519–525.

[38] G. Pleiss and J. P. Cunningham, "The limitations of large width in neural networks: A deep gaussian process perspective," 2021. [Online]. Available: https://arxiv.org/abs/2106.06529

[39] H. Salimbeni and M. Deisenroth, "Doubly stochastic variational inference for deep gaussian processes," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/8208974663db80265e9bfe7b222dcb18-Paper.pdf

[40] Z. Zhao, M. Emzir, and S. Särkkä, "Deep state-space gaussian processes," *Statistics and Computing*, vol. 31, no. 6, Sep. 2021. [Online]. Available: https://doi.org/10.1007/s11222-021-10050-6

[41] T. D. Bui, D. Hernández-Lobato, Y. Li, J. M. Hernández-Lobato, and R. E. Turner, "Deep gaussian processes for regression using approximate expectation propagation," 2016. [Online]. Available: https://arxiv.org/abs/1602.04133

[42] P. Jayashree and P. K. Srijith, "Evaluation of deep Gaussian processes for text classification," in *Proceedings of the Twelfth Language Resources and Evaluation*

Conference. Marseille, France: European Language Resources Association, May 2020, pp. 1485–1491. [Online]. Available: https://aclanthology.org/2020.lrec-1.185

[43] A. G. Wilson and P. Izmailov, "Bayesian deep learning and a probabilistic perspective of generalization," *CoRR*, vol. abs/2002.08791, 2020. [Online]. Available: https://arxiv.org/abs/2002.08791

[44] G. Yang and S. S. Schoenholz, "Mean field residual networks: On the edge of chaos," *CoRR*, vol. abs/1712.08969, 2017. [Online]. Available: http://arxiv.org/abs/1712.08969

[45] R. Novak, L. Xiao, J. Lee, Y. Bahri, G. Yang, J. Hron, D. A. Abolafia, J. Pennington, and J. Sohl-Dickstein, "Bayesian deep convolutional networks with many channels are gaussian processes," 2018. [Online]. Available: https://arxiv.org/abs/1810.05148

[46] A. Garriga-Alonso, C. E. Rasmussen, and L. Aitchison, "Deep convolutional networks as shallow gaussian processes," 2018. [Online]. Available: https://arxiv.org/abs/1808.05587

[47] M. Khan and S. Khan, "Khan et al 2019," 03 2019.

[48] Y. Gao, B. Gao, Q. Chen, J. Liu, and Y. Zhang, "Deep convolutional neural network-based epileptic electroencephalogram (eeg) signal classification," *Frontiers in Neurology*, vol. 11, 2020. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fneur.2020.00375

[49] S. Gupta, J. Meena, and O. P. Gupta, "Neural network based epileptic EEG detection and classification," *CoRR*, vol. abs/2111.03268, 2021. [Online]. Available: https://arxiv.org/abs/2111.03268

[50] U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, and H. Adeli, "Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals," *Computers in Biology and Medicine*, vol. 100, pp. 270–278, Sep. 2018. [Online]. Available: https://doi.org/10.1016/j.compbiomed.2017.09.017

[51] X. Lun, Z. Yu, T. Chen, F. Wang, and Y. Hou, "A simplified CNN classification method for MI-EEG via the electrode pairs signals," *Frontiers in Human Neuroscience*, vol. 14, Sep. 2020. [Online]. Available: https://doi.org/10.3389/fnhum.2020.00338

[52] X. Zou, S. Xu, S. Li, J. Chen, and W. Zou, "Optimization of the brillouin instantaneous frequency measurement using convolutional neural networks," *Opt. Lett.*, vol. 44, no. 23, pp. 5723–5726, Dec 2019. [Online]. Available: https://opg.optica.org/ol/abstract.cfm?URI=ol-44-23-5723

[53] A. Demir, T. Koike-Akino, Y. Wang, M. Haruna, and D. Erdogmus, "EEG-GNN: graph neural networks for classification of electroencephalogram (EEG) signals," *CoRR*, vol. abs/2106.09135, 2021. [Online]. Available: https://arxiv.org/abs/2106.09135

[54] N. Kumar and K. P. Michmizos, "A neurophysiologically interpretable deep neural network predicts complex movement components from brain activity," *Scientific Reports*, vol. 12, no. 1, Jan. 2022. [Online]. Available: https://doi.org/10.1038/s41598-022-05079-0

[55] S. Motamedi-Fakhr, M. Moshrefi-Torbati, M. Hill, C. M. Hill, and P. R. White, "Signal processing techniques applied to human sleep EEG signals—a review," *Biomedical Signal Processing and Control*, vol. 10, pp. 21–33, Mar. 2014. [Online]. Available: https://doi.org/10.1016/j.bspc.2013.12.003

[56] A. Craik, Y. He, and J. L. Contreras-Vidal, "Deep learning for electroencephalogram (EEG) classification tasks: a review," vol. 16, no. 3, p. 031001, apr 2019. [Online]. Available: https://doi.org/10.1088/1741-2552/ab0ab5

[57] D. Acharya, R. A. Sayyad, P. Dwivedi, A. Shaji, P. Sriram, and A. Bhardwaj, "EEG signal classification using deep learning," in *Advances in Intelligent Systems and Computing*. Springer Singapore, 2021, pp. 393–403. [Online]. Available: https://doi.org/10.1007/978-981-16-2709-5_30

[58] Y. Roy, H. Banville, I. Albuquerque, A. Gramfort, T. H. Falk, and J. Faubert, "Deep learning-based electroencephalography analysis: a systematic review," *Journal*

of Neural Engineering, vol. 16, no. 5, p. 051001, aug 2019. [Online]. Available: https://doi.org/10.1088/1741-2552/ab260c

[59] K. Aboalayon, M. Faezipour, W. Almuhammadi, and S. Moslehpour, "Sleep stage classification using EEG signal analysis: A comprehensive survey and new investigation," *Entropy*, vol. 18, no. 9, p. 272, Aug. 2016. [Online]. Available: https://doi.org/10.3390/e18090272

[60] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 6645–6649.

[61] H. Greenspan, B. van Ginneken, and R. M. Summers, "Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1153–1159, 2016.

[62] Y. Zhao and L. He, "Deep learning in the EEG diagnosis of alzheimer's disease," in *Computer Vision - ACCV 2014 Workshops*. Springer International Publishing, 2015, pp. 340–353. [Online]. Available: https://doi.org/10.1007/978-3-319-16628-5_25

[63] F. C. Morabito, M. Campolo, C. Ieracitano, J. M. Ebadi, L. Bonanno, A. Bramanti, S. Desalvo, N. Mammone, and P. Bramanti, "Deep convolutional neural networks for classification of mild cognitive impaired and alzheimer's disease patients from scalp eeg recordings," in *2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*, 2016, pp. 1–6.

[64] A. Gupta, P. Singh, and M. Karlekar, "A novel signal modeling approach for classification of seizure and seizure-free eeg signals," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 5, pp. 925–935, 2018.

[65] N. Kulkarni and V. Bairagi, "Extracting salient features for eeg-based diagnosis of alzheimer's disease using support vector machine classifier," *IETE Journal of Research*, vol. 63, pp. 1–12, 10 2016.

[66] J. C. McBride, X. Zhao, N. B. Munro, C. D. Smith, G. A. Jicha, L. Hively, L. S. Broster, F. A. Schmitt, R. J. Kryscio, and Y. Jiang, "Spectral and complexity analysis

of scalp EEG characteristics for mild cognitive impairment and early alzheimer's disease," *Computer Methods and Programs in Biomedicine*, vol. 114, no. 2, pp. 153–163, Apr. 2014. [Online]. Available: https://doi.org/10.1016/j.cmpb.2014.01.019

[67] S. J. Ruiz-Gómez, C. Gómez, J. Poza, M. Martínez-Zarzuela, M. A. Tola-Arribas, M. Cano, and R. Hornero, "Measuring alterations of spontaneous EEG neural coupling in alzheimer's disease and mild cognitive impairment by means of cross-entropy metrics," *Frontiers in Neuroinformatics*, vol. 12, Oct. 2018. [Online]. Available: https://doi.org/10.3389/fninf.2018.00076

[68] M. Buscema, F. Vernieri, G. Massini, F. Scrascia, M. Breda, P. M. Rossini, and E. Grossi, "An improved i-FAST system for the diagnosis of alzheimer's disease from unprocessed electroencephalograms by using robust invariant features," *Artificial Intelligence in Medicine*, vol. 64, no. 1, pp. 59–74, May 2015. [Online]. Available: https://doi.org/10.1016/j.artmed.2015.03.003

[69] A. Bablani, D. R. Edla, and S. Dodia, "Classification of EEG data using k-nearest neighbor approach for concealed information test," *Procedia Computer Science*, vol. 143, pp. 242–249, 2018. [Online]. Available: https://doi.org/10.1016/j.procs.2018.10.392

[70] N. Isa, A. Amir, M. Ilyas, and M. Razalli, "The performance analysis of k-nearest neighbors (k-nn) algorithm for motor imagery classification based on eeg signal," *MATEC Web of Conferences*, vol. 140, p. 01024, 01 2017.

[71] M. Li, H. Xu, X. Liu, and S. Lu, "Emotion recognition from multichannel EEG signals using k-nearest neighbor classification," *Technology and Health Care*, vol. 26, pp. 509–519, Jul. 2018. [Online]. Available: https://doi.org/10.3233/thc-174836

[72] S. Arafat, M. Dohrmann, and M. Skubic, "Classification of coronary artery disease stress ecgs using uncertainty modeling," in *2005 ICSC Congress on Computational Intelligence Methods and Applications*, 2005, pp. 4 pp.–.

[73] M. Lorenzi, M. Filippone, G. B. Frisoni, D. C. Alexander, and S. Ourselin, "Probabilistic disease progression modeling to characterize diagnostic uncertainty:

Application to staging and prediction in alzheimer's disease," *NeuroImage*, vol. 190, pp. 56–68, Apr. 2019. [Online]. Available: https://doi.org/10.1016/j.neuroimage.2017.08.059

[74] G. B. Frisoni, D. Altomare, D. R. Thal, F. Ribaldi, R. van der Kant, R. Ossenkoppele, K. Blennow, J. Cummings, C. van Duijn, P. M. Nilsson, P.-Y. Dietrich, P. Scheltens, and B. Dubois, "The probabilistic model of alzheimer disease: the amyloid hypothesis revised," *Nature Reviews Neuroscience*, vol. 23, no. 1, pp. 53–66, Nov. 2021. [Online]. Available: https://doi.org/10.1038/s41583-021-00533-w

[75] G. Mirzaei, A. Adeli, and H. Adeli, "Imaging and machine learning techniques for diagnosis of alzheimer's disease," *Reviews in the Neurosciences*, vol. 27, no. 8, pp. 857–870, Dec. 2016. [Online]. Available: https://doi.org/10.1515/revneuro-2016-0029

[76] S. Faul, G. Gregorcic, G. Boylan, W. Marnane, G. Lightbody, and S. Connolly, "Gaussian process modeling of EEG for the detection of neonatal seizures," *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 12, pp. 2151–2162, Dec. 2007. [Online]. Available: https://doi.org/10.1109/tbme.2007.895745

[77] L. L. W. Owen, T. A. Muntianu, A. C. Heusser, P. M. Daly, K. W. Scangos, and J. R. Manning, "A gaussian process model of human electrocorticographic data," *Cerebral Cortex*, vol. 30, no. 10, pp. 5333–5345, Jun. 2020. [Online]. Available: https://doi.org/10.1093/cercor/bhaa115

[78] A. Sarishvili, J. Winter, H. Luhmann, and E. Mildenberger, "Probabilistic graphical model identifies clusters of EEG patterns in recordings from neonates," *Clinical Neurophysiology*, vol. 130, no. 8, pp. 1342–1350, Aug. 2019. [Online]. Available: https://doi.org/10.1016/j.clinph.2019.04.708

[79] S. Chiappa, "Analysis and classification of eeg signals using probabilistic models for brain computer interfaces," 01 2006.

[80] A. Alexiou, V. D. Mantzavinos, N. H. Greig, and M. A. Kamal, "A bayesian model for the prediction and early diagnosis of alzheimer's disease,"

Frontiers in Aging Neuroscience, vol. 9, Mar. 2017. [Online]. Available: https://doi.org/10.3389/fnagi.2017.00077

[81] P. R. Pinheiro, A. K. A. d. Castro, and M. C. D. Pinheiro, "A multicriteria model applied in the diagnosis of alzheimer's disease: A bayesian network," in *2008 11th IEEE International Conference on Computational Science and Engineering*, 2008, pp. 15–22.

[82] Y. Sun, S. Lv, and Y. Tang, "Construction and application of bayesian network in early diagnosis of alzheimer disease's system," in *2007 IEEE/ICME International Conference on Complex Medical Engineering*, 2007, pp. 924–929.

[83] S. R. B. Shree and H. S. Sheshadri, "Diagnosis of alzheimer's disease using naive bayesian classifier," *Neural Computing and Applications*, vol. 29, no. 1, pp. 123–132, Jul. 2016. [Online]. Available: https://doi.org/10.1007/s00521-016-2416-3

[84] S. Liu, Y. Song, W. Cai, S. Pujol, R. Kikinis, X. Wang, and D. Feng, "Multifold bayesian kernelization in alzheimer's diagnosis," in *Advanced Information Systems Engineering*. Springer Berlin Heidelberg, 2013, pp. 303–310. [Online]. Available: https://doi.org/10.1007/978-3-642-40763-5_38

[85] F. L. Seixas, B. Zadrozny, J. Laks, A. Conci, and D. C. M. Saade, "A bayesian network decision model for supporting the diagnosis of dementia, alzheimers disease and mild cognitive impairment," *Computers in Biology and Medicine*, vol. 51, pp. 140–158, Aug. 2014. [Online]. Available: https://doi.org/10.1016/j.compbiomed.2014.04.010

[86] N. K. Batmanghelich, A. Dalca, G. Quon, M. Sabuncu, and P. Golland, "Probabilistic modeling of imaging, genetics and diagnosis," *IEEE Transactions on Medical Imaging*, vol. 35, no. 7, pp. 1765–1779, 2016.

[87] F. C. Stingo, M. Guindani, M. Vannucci, and V. D. Calhoun, "An integrative bayesian modeling approach to imaging genetics," *Journal of the American Statistical Association*, vol. 108, no. 503, pp. 876–891, Sep. 2013. [Online]. Available: https://doi.org/10.1080/01621459.2013.804409

[88] T. Chekouo, F. C. Stingo, M. Guindani, and K.-A. Do, "A bayesian predictive model for imaging genetics with application to schizophrenia," *The Annals of Applied Statistics*, vol. 10, no. 3, Sep. 2016. [Online]. Available: https://doi.org/10.1214/16-aoas948

[89] Y. Xin, J. Sheng, M. Miao, L. Wang, Z. Yang, and H. Huang, "A review of imaging genetics in alzheimer's disease," *Journal of Clinical Neuroscience*, vol. 100, pp. 155–163, Jun. 2022. [Online]. Available: https://doi.org/10.1016/j.jocn.2022.04.017

[90] P. Lu, , and O. Colliot, "Multilevel modeling with structured penalties for classification from imaging genetics data," in *Graphs in Biomedical Image Analysis, Computational Anatomy and Imaging Genetics*. Springer International Publishing, 2017, pp. 230–240. [Online]. Available: https://doi.org/10.1007/978-3-319-67675-3_21

[91] Y. Song, S. Ge, J. Cao, L. Wang, and F. S. Nathoo, "A bayesian spatial model for imaging genetics," *Biometrics*, vol. 78, no. 2, pp. 742–753, Apr. 2021. [Online]. Available: https://doi.org/10.1111/biom.13460

[92] L. Zhang, M. Guindani, and M. Vannucci, "Bayesian models for functional magnetic resonance imaging data analysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 7, no. 1, pp. 21–41, Nov. 2014. [Online]. Available: https://doi.org/10.1002/wics.1339

[93] A. Kalauzi, T. Bojić, and A. Vuckovic, "Modeling the relationship between higuchi's fractal dimension and fourier spectra of physiological signals," *Medical biological engineering computing*, vol. 50, pp. 689–99, 05 2012.

Appendices

.1 Appendix 1.

- **Data preparation:**

```
import pandas as pd
from sklearn.preprocessing import StandardScaler



scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
X_train_flat, Y_train_reg = prep_data(X_train, Y_train)
X_test_flat, Y_test_reg = prep_data(X_test, Y_test)

def train_random_sample(X_train,Y_train,n_data,din,dout):
    S=tf.concat([X_train, Y_train],1)
    Random_S=tf.stack(random.sample(list(S),n_data),0)
    X_train_F=tf.split(Random_S,[din,dout],1)[0]
    Y_train_F=tf.split(Random_S,[din,dout],1)[1]
    return X_train_F,Y_train_F
```

Data Load:

```
#Load dataset
dataset = pd.read_csv('Sample_Entropy.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
X_train, X_test, Y_train, Y_test = train_test_split(X, y,
    test_size=0.15, random_state=0)
```