Theory behind Inverse Problems PINNs

- Paul Louis Escapil Inchauspe

- Artemio Araya

- Daniel Sanchez

- Agustín Soto

- Elías Hawks

Thanks to Professor María Thomsen Solis for her classes on Heat Transfer, Thermodynamics and Fluid Dynamics, where most of the examples to be used in this document come from. Also for redirecting the importance of the concepts and the theory developed by her on the application of technical tools or computational algorithms.

Books

- Fluid Dynamics: https://drive.google.com/file/d/1xQiR_MQ8985s9JjL-MyOrO82j8WHGdPq/view?usp=sharing
- Electricity and Magnetism for Mathematicians: https://drive.google.com/file/d/1OswdBat51L2Tv3adim_KmgX47nN1pODG/view?usp=sharing

Índice

The objective of this document is to help future participants in the directed research workshop of the Adolfo Ibáñez University.

As well as explaining the math behind inverse problems. AI algorithms are not black boxes.

Under a statistical methodology, an initial or null hypothesis is not rejected unless empirical evidence shows otherwise.

1. IA and Industrial Process

1.1. Thermodynamics

- https://colab.research.google.com/drive/1JUzUBPEuqnKYCxo6BxNY5sAM3XhbzxeH?usp=sharing
- https://colab.research.google.com/drive/12tM7aDL9_stv7w-v2CeNoZcpgN9NFBf5?usp=sharing

1.1.1. Mathematical representation of Elements of Thermodynamics

1.2. Heat Transfer

- https://colab.research.google.com/drive/1q7A5HT_-1zoukSDZyjal9FPLG_imlm6J?usp=sharing

1.2.1. Mathematical Representation of Heat Transfer Elements

1.3. Electromagnetism

- https://colab.research.google.com/drive/1Fb-Uq1kwUgKcxmRui3k7GBmjz3833q1u?usp=sharing
- https://colab.research.google.com/drive/1hafIuEqEhiioZpQajJWUt6aSsPjIFx2V?usp=sharing

1.3.1. Mathematical Representation of Elements of Electromagnetism

1.4. Fluid Dynamics

- https://colab.research.google.com/drive/1C82mvWWrylb3vxYrxfU0TRjgb6NntUfd?usp=sharing

- https://colab.research.google.com/drive/1ftpYxW6-W1iypHdhIPFn7MFofvb26gNm?usp=sharing
- https://colab.research.google.com/drive/1bC4P4eJ6GHxDjwqHh73IT9ubplW9VA99?usp=sharing

1.4.1. Mathematical representation of Elements of Fluid Dynamics

1.5. Convection

- Convección Interna https://colab.research.google.com/drive/16DrJ7VjDnYzaS0USNcz8upAiVFYQG5HF?usp=sharing

  Convección Forzada Externa, Interna y Natural

- https://colab.research.google.com/drive/19Zq49drvMXavEuzHfWjyTLdhy7crHzZs?usp=sharing

  Convección Externa

- https://colab.research.google.com/drive/1IAfHsnjMZhQe0GETiYm_dv6Kplqkph5e?usp=sharing

  Ecuación de difusión de calor

- https://colab.research.google.com/drive/1-wCq3TP-9sM7eR7pZ8GUVish4iuIQRMw?usp=sharing

  Transferencia de Calor

- https://colab.research.google.com/drive/1q7A5HT_-1zoukSDZyjal9FPLG_imlm6J?usp=sharing

  Momentum

- https://colab.research.google.com/drive/1PhjGNHVIGK0Jek4NyEskEKfb3Ihl9LZy?usp=sharing

  Cinemática

- https://colab.research.google.com/drive/1f8_eslVUXN0-8IDR3ASJoHXNIMRTH7oL?usp=sharing

  Estática

- https://colab.research.google.com/drive/1E2HEK9027jVSRXRMAz5m2LPH0_wtlFtE?usp=sharing

1.5.1. Representación matemática de Elementos de Convección

2. Probability Theory

2.0.1. Algebra of Random Variables

Suma de Variables Aleatorias

1

Multiplicación de Variables Aleatorias

2

2.0.2. Hipótesis Testing and Estimation Methods

2.0.3. Maximum Likelihood Method

3. Applications

3.1. Finite Element Method

3.2. Estimation of Vector and Scalar Fields

3.2.1. Neural Networks

Let a random vector function $\vec{Y_{M\{...\}}}$, where for a input pattern vector $\vec{s}$ we have a random vector $\vec{Y_M}$ with a multivariate normal distribution $\mathcal{N}_{\mathcal{M}}(\vec{\mu}, \vec{\sigma})$ with constant variance $V\{\vec{Y_M}\}$.

Were exist a relation between the expected random vector $\vec{Y_M}$ with a deep neural network and the input pattern vector $\vec{s}$.

$$\vec{Y_{M(...)}}: \Re \to \text{Set of Random Variables}$$
$$\vec{s} \longmapsto T^{[n]}\big(\Theta^{[n]}\big(T^{[i]}(\Theta^{[i]} \cdot ... \cdot T^{[1]}\big(\Theta^{[1]}\vec{s} + \vec{b^{[1]}}\big)... + \vec{b^{[i]}}\big) + \vec{b^{[n]}}\big) = Y_M(\vec{s})$$

The coefficient matrices $\Theta^{[i]}$ and $b^{[i]}$ must have the proper dimensions that allow add and matrix multiplication, that is, the propagation of information through the neurons.

Deep Neural Network Expanded

$$h_\Theta(\vec{s}) = T^{[n]}\left(\begin{bmatrix} \theta^{(2)}_{(0,1)} & \cdots & \theta^{(2)}_{(0,N)} \\ \vdots & \ddots & \vdots \\ \theta^{(2)}_{(I,1)} & \cdots & \theta^{(2)}_{(N,I)} \end{bmatrix} T^{[i]}\left(\begin{bmatrix} \theta^{(i)}_{(0,1)} & \cdots & \theta^{(i)}_{(0,N)} \\ \vdots & \ddots & \vdots \\ \theta^{(i)}_{(I,1)} & \cdots & \theta^{(i)}_{(N,I)} \end{bmatrix} \cdots T^{[1]}\left(\begin{bmatrix} \theta^{(1)}_{(0,1)} & \cdots & \theta^{(1)}_{(0,N)} \\ \vdots & \ddots & \vdots \\ \theta^{(1)}_{(I,1)} & \cdots & \theta^{(1)}_{(N,I)} \end{bmatrix} \begin{bmatrix} s_1 \\ \vdots \\ s_n \end{bmatrix} + \begin{bmatrix} \theta^{(1)}_{(0,1)} \\ \vdots \\ \theta^{(1)}_{(L,1)} \end{bmatrix}\right) \cdots + \begin{bmatrix} \theta^{(i)}_{(0,1)} \\ \vdots \\ \theta^{(i)}_{(L,1)} \end{bmatrix}\right) + \begin{bmatrix} \theta^{(N)}_{(0,1)} \\ \vdots \\ \theta^{(N)}_{(L,1)} \end{bmatrix}\right) \tag{1}$$

$T^{[}i], \forall i = 1, ..., n$: No lineal Transformation

Maximum Likelihood Method

Let a random sample

$$\left\{ \mathbf{a^1}, \mathbf{a^2}, \ \dots \ , \mathbf{a^i}, \ \dots \ , \mathbf{a^m} \right\} = \left\{ \begin{bmatrix} \vec{Y_M}^{(1)} \\ s_1^{(1)} \\ \vdots \\ s_n^{(1)} \end{bmatrix} ; \begin{bmatrix} \vec{Y_M}^{(2)} \\ s_1^{(2)} \\ \vdots \\ s_n^{(2)} \end{bmatrix} ; \ \dots \ \begin{bmatrix} \vec{Y_M}^{(i)} \\ s_1^{(i)} \\ \vdots \\ s_n^{(i)} \end{bmatrix} \ \dots ; \ \begin{bmatrix} \vec{Y_M}^{(m)} \\ s_1^{(m)} \\ \vdots \\ s_n^{(m)} \end{bmatrix} \right\}$$

a colecction of m random vectors, where

$$Y_M^{(i)} \sim \mathcal{N}_{\mathcal{M}}(\vec{\mu}^{(i)}, \vec{\sigma^{(i)}}) \ , \forall i = 1, \dots m$$

By definition, remember that exist a relation between each input pattern vector $\vec{s}$ with the expected of the random variable $\vec{Y_M}$ with a deep neural network.

$$\forall \ i = 1, ..., m$$

$$E\{\vec{Y_{M\,(...)}}(\vec{s^i})\} = E\{\vec{Y_M}^{\,i}\} = T^{[n]}\big(\Theta^{[n]}\big(T^{[i]}(\Theta^{[i]} \cdot \dots \cdot T^{[1]}\big(\Theta^{[1]}\vec{s} + \vec{b^{[1]}}\big) \dots + \vec{b^{[i]}}\big) + \vec{b^{[n]}}\big)$$

Let a random experiment that consist in select a random sample previously defined. For a **random sample** observed, the likelihood function measure the joint density of having extracted that punctual sample in function of different possibilities of the learning vectors $\vec{\Theta}$ and b that generated it.

The objective of the likelihood function **for that sample** is, therefore, a way of ordering the possible generator parameters $\vec{\theta}$ of the system according to how best they fit in comparison with other learning parameters.

In simple terms, the maximum likelihood method says that if we look at that sample, it must be for a reason. Because it had to have been generated by the parameters that associate that sample with the highest joint density compared to any other possible combination of parameters of the model.

Likelihood function for a neural network that parameterizes the expectation of a multivariate normal distribution

Let u represent a deep neural network with 4 output variables.

$$\mathcal{L}(\Theta_{\vec{\text{Concatenation}}} \ |\mathbf{a^1}, \mathbf{a^2}, \ \dots, \mathbf{a^i}, \ \dots, \mathbf{a^m}) = \prod_{i=1}^{m} \mathcal{N}_M(\vec{Y^i}; u(\vec{s}), \vec{\sigma}), \sigma \in \Re$$

$$\exists \ \Theta_{\vec{\text{Concatenation}}max} \setminus \mathcal{L}(\Theta_{\vec{\text{Concatenation}}max} \ |\mathbf{a^1}, \mathbf{a^2}, \ \dots, \mathbf{a^i}, \ \dots, \mathbf{a^m})$$

$$= \text{máx} \left\{ \mathcal{L}(\Theta_{\text{Concatenation}} \ |\mathbf{a^1}, \mathbf{a^2}, \ \dots, \mathbf{a^i}, \ \dots, \mathbf{a^m}), \ \Theta_{\text{Concatenation}} \in M[\Re]_{nx1} \right\}$$

As in multiple linear regression, here we can show mathematically that maximizing the joint density of the sample is minimizing the "mean square.[er]ror of the predicted responses for each input pattern vector $\vec{s}$ and the desired response $\vec{y}$.

Demonstration

`https://www.statlect.com/fundamentals-of-statistics/multivariate-normal-distribution-maximu`

3.3. Solving the System of Equations: Gradient Descent & Backpropagation

Solving the System of Equations

It is well known that in order to find the concatenation of parameters that reduce the error of the predictions, or maximize the probability of the sample, those whose evaluation in the gradient function of said functions are very close to $\vec{0}$.

3.3.1. Neural Networks with Differential Equations(Physic Informed Neural Networks

- $T^{[i]}, \forall i \in \{1, ..., n\}$

Debido a composición interna de la red neuronal, el algoritmo del descenso de gradiente de la misma forma que se aplica en regresión lineal o logística no es suficiente para asegurar la convergencia y la efectiva realización del algoritmo.

Por esa razón, se utiliza el algoritmo backpropagation,para calcular la EVALUACIÓN en una matriz Θ en la función gadiente asociada a la función de verosimilitud definida arriba, ver ecuación.

En cada interación del algoritmo de descenso, se utilizará el algoritmo backpropagation para calcular este vector de números que ayudará a encontrar LOS PARÁMETROS Θ que maximizan la probabilidad conjunta de esa muestra puntual, suponiendo que efectivamente se cumplan todos los supuestos. Nosotros no conocemos cual es la relación determinista entre la probabilidad y la secuencia de observaciones, y es precisamente esa relación es la que podemos obtener con los parámetros.

La magnificiencia de la red neuronal es que permite aproximarse a un monton de funciones distintas, debido a su composición interna y numero de parámetros.

Es capaz de configurarse de tal forma de obtener una distinta elasticidad y elasticidad cruzada para distintos elementos de su dominio. Así como cambiar su derivada y derivada çruzada también". Pensando desde un punto de vista microeconomico, suponiendo que las salidas son la cantidad vendidad y los inputs son el precio.

Bajo los supuestos considerados, entonces tenemos.

Prueba de hipótesis

- H_0: La configuración de parámetros óptima $\Theta_{\text{Óptima}}$ que configuran la red neuronal efectivamente producen la relación determinista deseada $\mu(x; \vec{s})$.
- H_a: La configuración de parámetros óptima $\Theta_{\text{Óptima}}$ que configuran la red neuronal no reproduce la relación determinista deseada $\mu(x; \vec{s})$.

$$Verosimilitud = \prod_{i=1}^{m} N(h_\Theta(\vec{s_i}), \sigma)$$

Es demostrable matemáticamente que si la varianza de la colección de variables aleatorias es exactamente el mismo valor, entonces máximizar la probabilidad conjunta expresada en la función de arriba es igual a minimizar el error del cuadrático medio entre la predicción del modelo y la respuesta observada registrada en la base de datos.

Interpretación Económica Frecuentista Probabilística

Sea una colección de compradores y vendedores, propios de los compradores poseer una serie de elementos como una función de utilidad y un operador vectorial que permite definir sus preferencias. Además, sea su comportamiento racional, esto es,

La red neuronal asume que puede como no puede ex

Referencias