



UNIVERSIDAD NACIONAL AUTÓNOMA DE HONDURAS
CARRERA DE INGENIERÍA EN SISTEMAS
Asignatura Inteligencia Artificial
Asignación 02 Clasificación de Latidos del Corazón usando
Redes Neuronales Profundas

Descripción

El proyecto consiste en la implementación de un Red Neuronal Profunda para la clasificación de los latidos del corazón humano. Este es un problema de clasificación multiclase ya que se cuenta con 3 posibles tipos de latidos (**Normal (N)**, **Anormal (A)**, **Inclasificable (U)**). Los conjuntos de datos que deberá usar para entrenar su red se proporcionan junto a este documento, note que para poder realizar el entrenamiento deberá codificar las etiquetas utilizando la técnica conocida como **One Hot Encoding** que se explica en el video adjunto. **No modifique los archivos `x_train` y `x_test`** y realice directamente el entrenamiento con ellos (sino la revisión automática no se podrá realizar correctamente).

Una vez que tenga una red entrenada deberá almacenar sus parámetros e hiperparámetros en un archivo en formato JSON nombrado **`params.json`**, también se proporciona un ejemplo del formato de este archivo junto con este documento. El archivo contiene un arreglo nombrado **`dnn_layers`** con los datos de las capas de su red neuronal (menos la capa de entrada que no tiene parámetros), **`n`** es el número de unidades de la capa y **`activation`** el nombre de la función de activación que usó en dicha capa que puede ser **`relu`**, **`tanh`**, **`sigmoid`** o **`softmax`**, **`w`** es la matriz W para dicha capa representada como un arreglo de arreglos, cada arreglo dentro de dicho arreglo representa una fila de la matriz W. Finalmente, de manera similar a **`w`**, **`b`** contiene el vector b para cada capa. Evidentemente, los tamaños de W y b deben ser los adecuados según el valor de **`n`** para cada capa y la función **`softmax`** sólo debería ser usada en la última capa.

No es necesario que se incluya la información de la capa de entrada pues esta es constante.

Recursos

[Video sobre clasificación multiclase.](#)

[Dataset y archivo `params.json` de ejemplo.](#)

Requisitos de Presentación

- El principal archivo a entregar será el archivo `params.json`, este será inicialmente el único archivo a revisar por parte del docente. A manera de respaldo, también deberá entregar un archivo `modelo.py` donde estará todo el código en python que utilizó para entrenar la red, éste sólo se revisará en caso de que el docente considere necesario.
- Antes de enviar su trabajo es muy importante que pueda revisar la correctitud del mismo a través de la aplicación *autograder*; el autograder sólo revisará el archivo `params.json`, creará una red neuronal con los parámetros que ahí estén y probará su precisión con datos de prueba (diferentes a los proporcionados, para comprobar que no haya sobreajuste). Si el archivo se lee correctamente, el autograder le proporcionará la precisión de su red con los datos de prueba y una nota preliminar de acuerdo a la precisión alcanzada.
<https://rp-autograder.herokuapp.com/>
- La fecha máxima de entrega está definida en el campus virtual.
- El ejercicio se desarrollará de forma individual.
- Para la entrega del ejercicio solamente deberá subir los archivos `params.json` y `modelo.py` mediante el enlace proporcionado en el campus virtual, no se preocupe por colocarle su nombre, este se coloca automáticamente según el usuario que lo subió.
- Una vez que todas las tareas sean entregadas se usará un software evaluador que comparará su código con el del resto de sus compañeros, si su código es *demasiado similar*, se considerará plagio.
- El plagio en cualquiera de sus formas se penalizará con una nota de 0.
- No cumplir con alguno de los puntos anteriores implicará una disminución en su nota, favor si alguno no está claro, consultar en clase.

Elaborado por: Raúl José Palma Mendoza.