**Manual de uso: Automatización de pruebas para “User-Specific” usando MATLAB R2020b**

|  |  |
| --- | --- |
| Código | QNNEMG-WindSize-1 |
| Versión: | 1.0.0 |
| Fecha de la versión: | 03/02/2021 |
| Creado por: | Danny Díaz |
| Nivel de confidencialidad: | Público |

**Historial de modificaciones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Creado por** | **Descripción de la modificación** |
| 04/02/2021 | 1.0.0 | Danny Díaz | Versión inicial |
|  |  |  |  |
|  |  |  |  |

Tabla de contenido

[Inicio rápido 2](#_Toc63319164)

[Descripción del proceso de prueba y de la tarea de automatización 4](#_Toc63319165)

[Parámetros de la función parametrizada 6](#_Toc63319166)

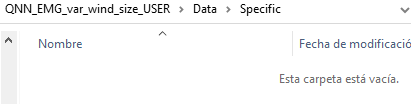
[Descripción de los script BATCH usados para automatizar los procesos 7](#_Toc63319167)

# Inicio rápido

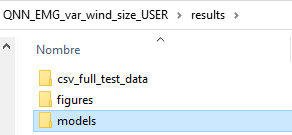
**Pre-requisitos**

* MATLAB R2020b
* Verificar que existan en la carpeta “modelo” (QNN\_EMG\_var\_wind\_size\_USER), las subcarpetas:

Data/Specific sin ningún dato de usuario:



Y sobre todo results:



Con csv\_full\_test\_data vacía

Con figures vacía

Con models vacía

GIT suele omitir subir a GitHub carpetas vacías por lo que puede ser necesario generar estas carpetas de forma manual solo en esta carpeta “modelo”

**Procedimiento**:

Abrir el CMD en la carpeta contenedora y ejecutar “0\_replicate\_with\_user\_data.bat 1 10”, esta instrucción copiará la carpeta modelo y le pondrá los datos respectivos de los usuarios: user1,2,3,4,5,6,7,8,9,10.



Esto genera dos carpetas con cada dato de usuario en “Data/Specific”:



Luego para ejecutar los experimentos del 27 al 32 (esto está fijo en el .bat) con:

* 3 épocas
* Para los tamaños de ventana 100 y 105
  + Cada tamaño de ventana con strides 20,30 y 40
* Desde el usuario 1 hasta el usuario 2

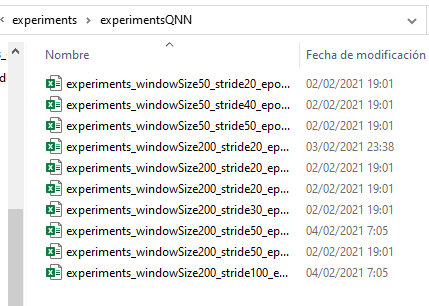
Se ejecuta en el CMD:

1\_execute\_matlab\_instruction\_for\_folder.bat 3 "[100 105]" "[20 30 40]" 1 2



Luego de ejecutar esta instrucción abrirá una instancia en user1 ejecutando el training con los parámetros indicados y luego de 15 segundos abrirá otra instancia en user2 ejecutando el training los parámetros indicados.

Al final los resultados son guardados en la misma carpeta de usuario dentro en la carpeta “results”. Y escribirá un Excel afuera de esa carpeta, en la carpeta “experiments\experimentsQNN”:

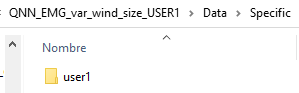


Cada Excel posee los datos de los usuarios en cada una de sus hojas.

# Descripción del proceso de prueba y de la tarea de automatización

**Situación del proceso**

Para ejecutar una prueba para un usuario X, se requiere tener en la carpeta “Data/Specific” el dato correspondiente a ese usuario.



Y luego en la instancia de MATLAB ejecutar el script correspondiente a un experimento con los parámetros codificados manualmente.

Los procesos identificados a automatizar son:

1. Crear una nueva carpeta con el dato de un usuario X, 306 veces.
2. Ejecutar un script para cada usuario con los parámetros requeridos de cada experimento (inicialmente seis: 27, 28, 29, 30, 31, 32.

**Automatización**

Primero se parametriza todos lo necesario de cada experimento en una misma función tanto para el entrenamiento:

QNN\_train\_emg\_Exp\_Replay\_SxWx.m



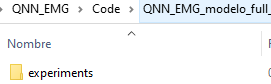
Como para la validación:

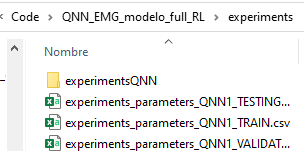
QNN\_validation\_Exp\_Replay\_SxWx.m

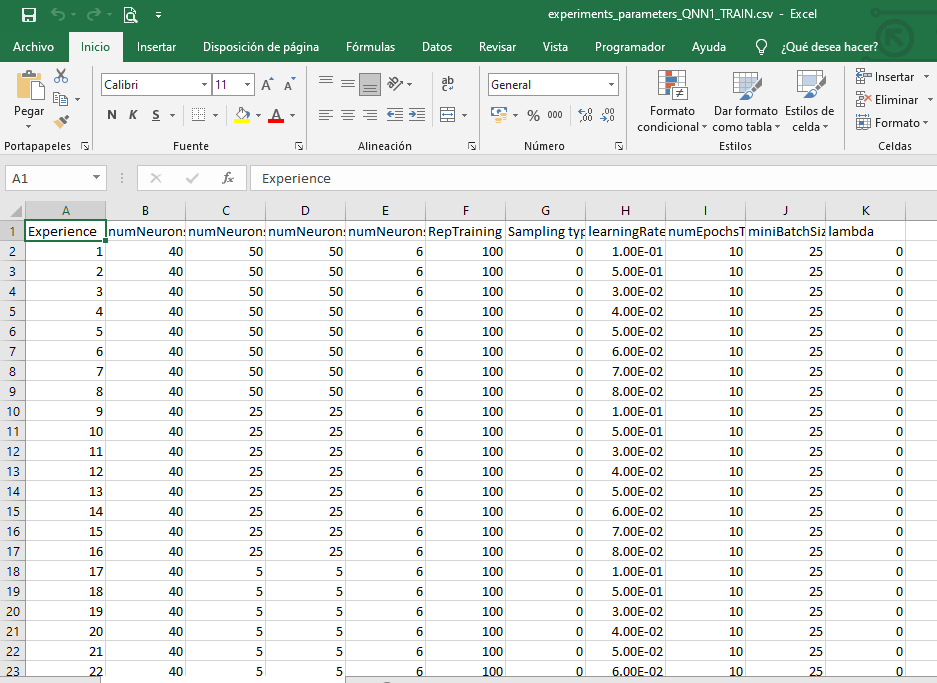


En la validación la variable “numRealEpochs” solo se usa para el almacenamiento de datos.

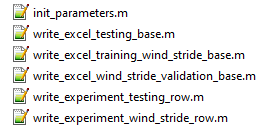
Los datos de los experimentos se sacan del CSV respectivo de entrenamiento, validación o testeo escritos literalmente en “experiments”:







Las siguientes funciones son llamadas de forma interna por los scripts de validación y testeo. Se usan para leer del csv guandando en variables y otros para escribir cabeceras y filas en un nuevo archivo Excel:



Todos se ubican en la carpeta experiments.

## Parámetros de la función parametrizada

* USER\_ID: int

Es el número del usuario. Ejms: 1,3,5,306

* experiment\_begin: int

En el CSV de expériments, el identificador del experimento de inicio. Ejm: 27

* experiment\_end: int

En el CSV de expériments, el identificador del experimento de inicio. Ejm: 32

* make\_validation\_too: bool

Si es verdadero hace también validación usando rangedown=26+repTraining del entrenamiento, caso contrario la validación no se ejecuta.

* write\_excel: bool

Si es verdadero escribirá resultados en experiments, caso contrario no escribirá esas filas

* windows\_sizes: array

Vector con todos los tamaños de ventana a testear. Ejm: [200 250 300]

* strides: array

Vector con todos los strides para probar con cada uno de los tamaños de ventana anteriormente descritos. Ejm: [20 40 50]

* numRealEpochs: int

Número de épocas. Es decir, número de veces que se le mostrará los mismos datos al modelo. Ejm: 10

**Ejemplo de llamada a la función**,

Si tengo en specific al usuario 288 y quiero probar del experimento 27 al 30 definidos en el Excel, con validación y escribiendo en excel. Para solamente el tamaño de ventana 200 y strides 100, 50. Finalmente, se le muestra dos veces los mismos datos (real-epochs es 2)

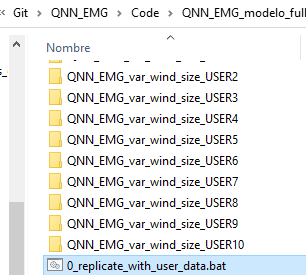
**QNN\_train\_emg\_Exp\_Replay\_SxWx(288, 27, 30, true, true, [200], [50 100], 2)**

## Descripción de los script BATCH usados para automatizar los procesos

Los procesos identificados a automatizar son:

1. Crear una nueva carpeta con el dato de un usuario X, 306 veces.

Para esta tarea se usa el script “0\_replicate\_with\_user\_data.bat” escrito en lenguaje BATCH:



Pide dos parámetros:

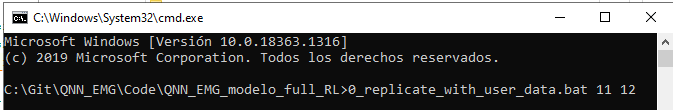
* %1: int

Es el número del usuario de inicio (desde que usuario se va a generar y copiar datos)

* %2: int

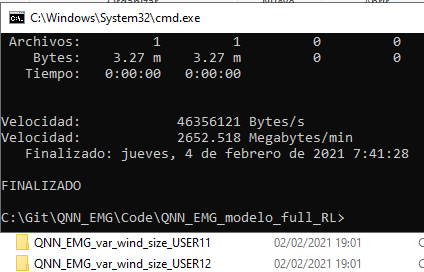
Es el número del usuario de fin (desde que usuario se va a generar y copiar datos)

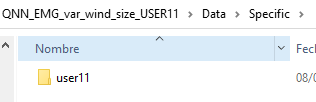
Ejemplo de uso:

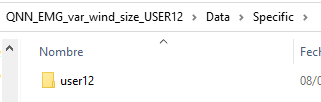


Resultado:

Se creó la carpeta de user11 y user12 con sus respectivos datos





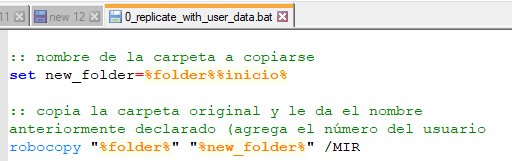


Se ejecuta un ciclo desde %1 hasta %2 y se crean carpetas de manera secuencial del usuario.

Se requiere una carpeta modelo, es decir, sin nada en “Data/Specific” y ningún resultado guardado en la carpeta “results”:

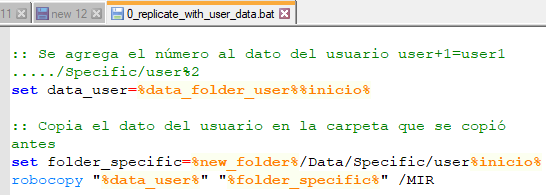


Se copia la carpeta sin resultados o datos de usuario:



A la carpeta modelo se le agrega el número del usuario al final para evitar sobre escribir y poder identificarlo.

Luego se copia de Data Full el dato del usuario dentro de la carpeta recientemente copiada:

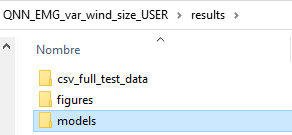


Se puede generar entonces de forma automática las 306 carpetas de los usuarios con su dato respectivo

IMPORTANTE: Verificar que existan en la carpeta “modelo”, las subcarpetas:

Data/Specific

Y sobre todo results:



Con csv\_full\_test\_data vacía

Con figures vacía

Con models vacía

GIT suele omitir subir a GitHub carpetas vacías por lo que puede ser necesario generar estas carpetas de forma manual

1. Ejecutar un script para cada usuario con los parámetros requeridos de cada experimento (inicialmente seis: 27, 28, 29, 30, 31, 32.

Para esto se usa el script BATCH “1\_execute\_matlab\_instruction\_for\_folder.bat”.

Cuyos parámetros son:

:: %1 -> numEpochs (veces que se muestra los datos)

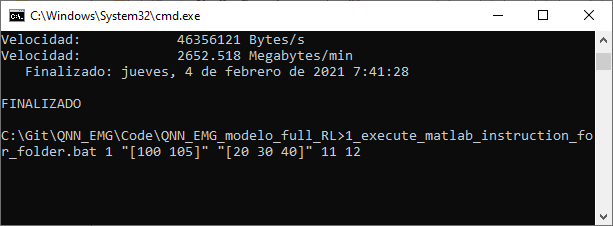
:: %2 -> window\_sizest en vector dentro de string "[100 105]"

:: %3 -> stridest en vector dentro de string "[20 30 40]"

:: %4 -> usuario inicial

:: %5 -> hasta usuario final

Ejemplo de uso en el CMD:

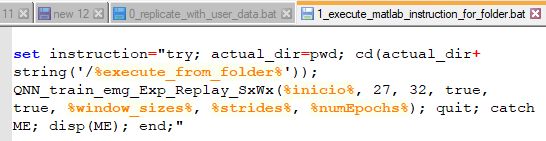


La instrucción ejecuta 1 epoca para los tamaños de ventana 100 y 105, y para cada tamaño de ventana prueba los strides 20,30 y 40. Estas instancias son abiertas en las carpetas “…USER11” y “…USER12”.

Ejemplo de solo ejecutar el usuario 288: 1\_execute\_matlab\_instruction\_for\_folder.bat 1 "[200]" "[20 30 40]" 288 288

Se ejecuta desde el experimento 27 al experimento 32 definidos en los CSV de TRAINING y VALIDATION.

Para cambiar esto se debe cambiar los parámetros dentro del archivo .bat:



Específicamente en esta sección:



El número de inicio debe ser menor al segundo porque se ejecutan los experimentos de manera secuencial. La instancia de MATLAB se cerrará automáticamente