

**MANUAL DE USUARIO SOFTWARE DE ENTRENAMIENTO
PARA REDES NEURONALES (MODELOS
SECUENCIALES)**

VERSIÓN ALFA

**DESARROLLADORES:
ALEJANDRO TORO OSSABA
JUAN JOSE JARAMILLO TIGREROS**

25/11/2020

INDICE

	Pag.
1. CONSIDERACIONES Y OBSERVACIONES DE LA VERSIÓN	3
2. LIBRERIAS NECESARIAS PARA EL USO DEL SOFTWARE	4
3. EJECUCIÓN DEL SOFTWARE	4
4. INTERFAZ PRINCIPAL	4
5. DIÁLOGO DE CONFIGURACIÓN	5
5.1 PESTAÑA DE ARQUITECTURA	5
5.2 PESTAÑA DE ENTRENAMIENTO.....	6

1. CONSIDERACIONES Y OBSERVACIONES DE LA VERSIÓN

- Esta es una versión inestable de prueba (Versión Alfa), utilizada para validar el potencial del software, sus funcionalidades, realizar pruebas y verificar fallos. La intención general es el desarrollo de un software amigable con personas que tienen pocos conocimientos o conocimientos muy básicos acerca de modelos neuronales, para que estas puedan desarrollar y entrenar modelos neuronales para sistemas inteligentes de una forma sencilla e intuitiva. Además, posteriormente se busca integrarla a otras interfaces que permite recolectar los datos enviados por una manilla de adquisición electromiografica, guardarlos y graficarlos, para el posterior entrenamiento de los modelos neuronales.
- Esta versión puede presentar errores en algunos procesos.
- Esta versión no posee todas las funcionalidades y parámetros de configuración posibles para los modelos neuronales. En futuras versiones se pretende agregar mayor cantidad de opciones de configuración que permitan personalizar aún más los modelos y el entrenamiento de los mismos, así como también mejorar la presentación de la información en la interfaz para facilitar su análisis.
- Esta versión no fue compilada en un archivo ejecutable (exe) ya que es una versión de prueba. Por estos motivos es necesario contar con Python en la maquina donde se vaya a correr el software.
- Cuando se está entrenando un modelo, en cada iteración el software automáticamente realiza un checkpoint o punto de guardado del modelo, este se guarda en la carpeta checkpoint contenida dentro del directorio del software. Este archivo es el checkpoint del último modelo entrenado y está en formato h5.
- En el directorio del software hay dos dataset para probar el software. El primero es dataset_prueba_Tx200_4Canales_4Clases, el cual contiene datos aleatorios y solo es de utilidad para interactuar y familiarizarse con el software; y dataset_EMG_Tx200_4Canales_5Clases, el cual contiene una grabación de 5 gestos adquiridos mediante una manilla electromiografica.
- En el directorio del software también se puede encontrar el código dataSetGenerator.py el cual permite generar dataset de diferentes tamaños con números aleatorios, esto sirve nada más para validar el funcionamiento del software.
- El archivo para correr el software es **Interfaz.py**.

2. LIBRERIAS NECESARIAS PARA EL USO DEL SOFTWARE

- PyQt5.
- PyQt Graph.
- Pathlib.
- Numpy.
- Matplotlib.
- Pickle.
- Math.
- Tensorflow 2.2 o superior.
- Keras.

3. EJECUCIÓN DEL SOFTWARE

Para ejecutar el software se requiere Python 3.5 a 3.8.

Abrir la carpeta **Software de Entrenamiento de Redes neuronales**, en esta ejecutar el archivo de Python **Interfaz.py**.

4. INTERFAZ PRINCIPAL



Ilustración 1. Interfaz principal para monitoreo del entrenamiento.

1. **Grafica del costo durante el entrenamiento:** permite observar la gráfica del costo durante el entrenamiento mientras el mismo progresa.
2. **Grafica de la exactitud durante el entrenamiento:** permite observar la gráfica de la exactitud durante el entrenamiento mientras el mismo progresa.
3. **Valor del costo durante el entrenamiento:** indica el valor actual del costo.
4. **Valor de la exactitud durante el entrenamiento:** indica el valor actual de la exactitud.
5. **Costo y exactitud en el set de prueba:** indica el costo y la exactitud obtenidos en el dataset de validación o prueba. Por defecto el programa reserva el 20% del total de los datos cargados para realizar la validación del modelo. Estos valores solo se muestran al usuario al final del entrenamiento.
6. **Progreso del entrenamiento:** indica en porcentaje el progreso del entrenamiento, además indica cuantas iteraciones han transcurrido del total.
7. **Botón de acceso a la configuración:** permite acceder al dialogo de configuración del modelo.
8. **Botón para detener el entrenamiento:** permite detener el entrenamiento.
9. **Botón para iniciar el entrenamiento:** permite inicial el entrenamiento. Primero debe configurarse el modelo para poder habilitar el botón de inicio.

5. DIÁLOGO DE CONFIGURACIÓN

5.1 PESTAÑA DE ARQUITECTURA

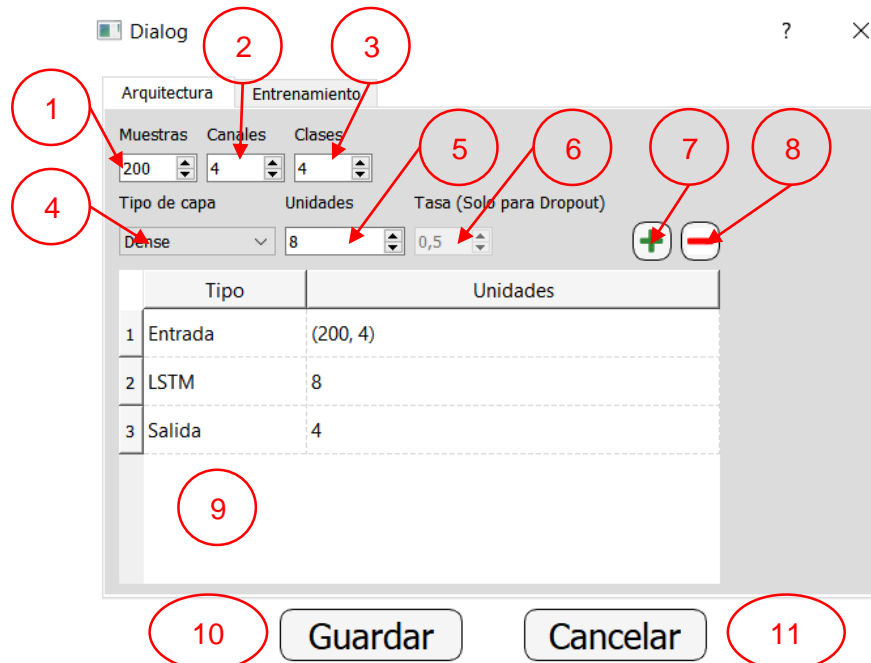


Ilustración 2. Pestaña para configurar la arquitectura de la red neuronal en el dialogo de configuración.

1. **Numero de muestras (Tx):** establecer el número de muestras en la ventana.

2. **Número de canales (entradas):** número de canales o entradas en el dataset en cada una de las muestras.
3. **Numero de clases:** número de clases para clasificación.
4. **Tipo de capa:** tipo de capa a añadir en el modelo. Se puede seleccionar entre Dense, la cual corresponde a una capa oculta convencional; LSTM, la cual corresponde a una capa recurrente; y Dropout, la cual corresponde a una estrategia de regularización para evitar el sobreajuste.
5. **Número de unidades en la capa:** seleccionar el número de unidades o neuronas en la capa. Solo es válido para las capas Dense y LSTM.
6. **Tasa de dropout:** permite establecer el porcentaje de desactivación de unidades en la capa, por ejemplo, un valor de 0.5 indica que en cada pasada por la capa se desactivan aleatoriamente el 50% de las unidades en la capa.
7. **Agregar capa:** permite agregar la capa con la configuración establecida.
8. **Eliminar capa:** permite eliminar una capa del modelo. Para esto se debe seleccionar la fila dando click en el numero en la tabla de arquitectura de la red. Como mínimo la red debe tener su capa de entradas, una capa oculta y la capa de salidas.
9. **Arquitectura de la red:** permite observar cada una de las capas en el modelo neuronal junto con sus dimensiones. Permite la selección de capas para su eliminación.
10. **Guardar configuración:** guarda la configuración establecida.
11. **Cancelar configuración:** cancela la configuración y conserva la configuración por defecto o en caso de haber realizado una configuración previa, conserva esta.

5.2 PESTAÑA DE ENTRENAMIENTO

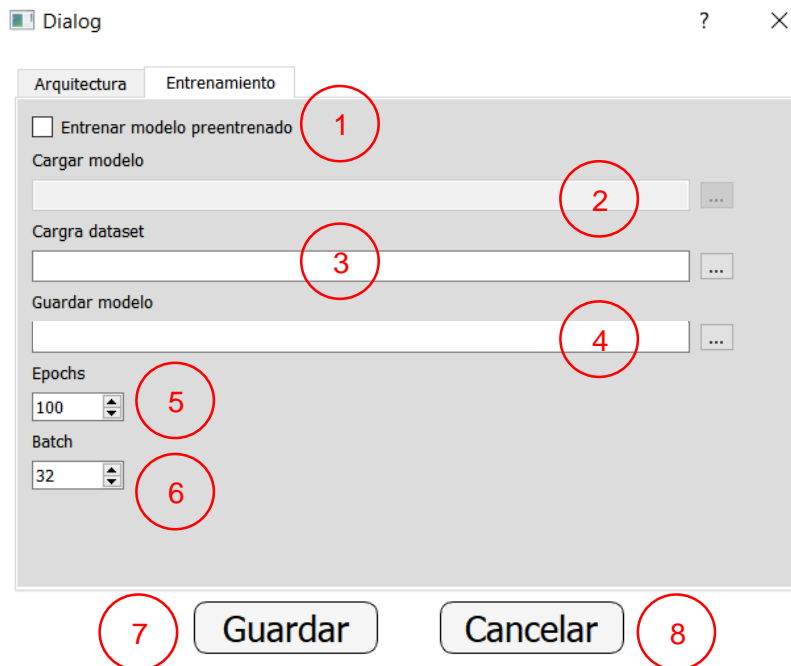


Ilustración 3. Pestaña para configurar el entrenamiento en el dialogo de configuración.

1. **Definir si se utiliza modelo preentrenado:** chequear esta casilla permite cargar un modelo existente para realizar un entrenamiento sobre el mismo.
2. **Cargar modelo preentrenado:** si la casilla de modelo preentrenado esta chequeada permite cargar un modelo existente seleccionándolo desde el explorador de archivos. El modelo debe estar en formato h5
3. **Cargar datos para el entrenamiento:** permite seleccionar desde el explorador de archivos el dataset para entrenar el modelo. El dataset debe estar en **formato.pkl**. Su estructura debe ser una tupla que contiene dos listas, la primera es una lista de m elementos, donde cada elemento es una ventana de datos de tamaño $Muestras(Tx) \times Entradas$; la segunda lista contiene m donde cada elemento es una etiqueta correspondiente a la clase de la ventana que se encuentra en el mismo índice de la primera lista.
4. **Dirección para guardar el modelo:** permite definir la dirección donde se guardará el modelo una vez haya completado el entrenamiento. El modelo se guarda en extensión h5.
5. **Cantidad de Epochs (Iteraciones):** permite definir la cantidad de iteraciones para el entrenamiento.
6. **Tamaño del Batch (Lote):** define el tamaño de lote que se utilizara en cada una de las iteraciones (Epochs) para realizar la técnica de gradiente descendiente por mini lotes.
7. **Guardar configuración:** guarda la configuración establecida.
8. **Cancelar configuración:** cancela la configuración y conserva la configuración por defecto o en caso de haber realizado una configuración previa, conserva esta.

El botón de guardar y cancelar es común a ambas pestañas, por tanto, si se presiona guardar se guardará la configuración definida en ambas pestañas.