

Ejemplo: NN_ex2.m

Author: D.Sc. Aboud Barsekh Onji **Institution:** Universidad Anáhuac México
- Facultad de Ingeniería **Contact:** aboud.barsekh@anahuac.mx **ORCID:** 0009-0004-5440-8092

A continuación, se describen los comandos clave utilizados para la predicción de series de tiempo con LSTM en MATLAB (`trainnet`), explicando las opciones abordadas en el código y el flujo de trabajo para regresión.

1. Carga y Visualización de Datos

Función: `stackedplot`

Visualiza datos de series temporales multivariadas apiladas verticalmente. - **En el código:** Se usa para mostrar los 3 canales de la forma de onda en un solo gráfico, facilitando la comparación visual de tendencias.

2. Arquitectura de la Red (LSTM para Regresión)

Para predicción de series de tiempo (regresión), la arquitectura difiere ligeramente de la clasificación. - **Capa Salida:** Se elimina `softmaxLayer` y la capa final totalmente conectada (`fullyConnectedLayer`) debe tener el tamaño de la salida deseada (en este caso, igual a la entrada: 3 canales). - **Modo de Salida LSTM:** - `sequenceInputLayer(3)`: Define la entrada de 3 canales. - `lstmLayer`: Aprende dependencias temporales.

3. Configuración del Entrenamiento

Función: `trainingOptions("adam", ...)`

Define la estrategia de aprendizaje de la red neuronal para regresión.

- **SequencePaddingDirection="left":**
 - **Importante:** En RNNs/LSTMs, el *padding* (relleno) puede afectar la predicción final. Rellenar a la izquierda asegura que los valores más recientes (a la derecha) sean reales y no ceros de relleno, lo cual es crucial para la predicción precisa del siguiente paso.
 - **Shuffle="every-epoch":** Mezcla las secuencias para evitar sesgos de orden.
 - **Plots="training-progress":** Visualización en tiempo real.
 - **Verbose=false:** Oculta la salida detallada en la consola para mantenerla limpia.
-

4. Entrenamiento de la Red

Función: `trainnet(X, T, net, loss, options)`

- **Loss Function (mse):**
 - **mse (Mean Squared Error):** Es la función de pérdida estándar para problemas de regresión, calculando el promedio de los errores al cuadrado entre la predicción y el valor real. Minimizar este error ajusta la red para predecir valores continuos.
-

5. Predicción de Ciclo Cerrado (Closed Loop Forecasting)

Estrategia:

En lugar de predecir todo de una vez (Open Loop), se predice paso a paso, retroalimentando la predicción de la red como entrada para el siguiente paso.

1. **resetState(net):** Reinicia el estado interno de la LSTM antes de predecir una nueva secuencia.
2. **Inicialización:** Se alimenta la red con los datos de entrada conocidos (**X**) para actualizar su estado interno hasta el punto de inicio de la predicción (`predict(net, X)`).
3. **Bucle de Predicción (for loop):**
 - Se predice el paso **t** usando el paso **t-1** (que fue una predicción del modelo anterior).
 - Se actualiza el estado (`net.State`) en cada iteración.
 - Esto permite generar secuencias futuras indefinidamente.
4. **Consideración:** El error tiende a acumularse con el tiempo en predicciones de ciclo cerrado largas, ya que cada nueva predicción se basa en una aproximación anterior.