Investigación sobre RNN en PyTorch y dataset Sunspots

Análisis del funcionamiento de capas en torch.nn

RNN

Input esperado La capa torch.nn.RNN recibe como entrada un tensor que representa una secuencia de datos a procesar en el tiempo. El formato por defecto es (longitud_secuencia, lote, cantidad_entradas); si se activa batch_first=True, pasa a ser (lote, longitud_secuencia, cantidad_entradas). En la documentación, se indica:

- input: tensor de forma (L, N, H_in) cuando batch_first=False o (N, L, H_in) si batch_first=True, donde L es la longitud de la secuencia, N es el tamaño del lote y H_in es el número de características por paso. (docs.pytorch.org)

Output devuelto La capa retorna dos cosas:

- output: secuencia completa de estados ocultos, con forma (L, N, D × hidden_size) si no se utiliza batch_first, o (N, L, D × hidden_size) si sí. Aquí, D es 2 si la RNN es bidireccional, o 1 en caso contrario. (docs.pytorch.org)
- h_n: último estado oculto para cada capa (y dirección), con forma (D × num_layers, N, hidden_size).

Parámetros de configuración principales

- **input_size**: cantidad de características que aporta cada punto de la secuencia (H_in).
- **hidden_size**: tamaño del vector oculto (dimensionalidad del estado que se propaga entre pasos).
- **num_layers**: cuántas capas recurrentes están apiladas (e.g. num_layers=2 genera dos RNN conectadas en serie).
- **nonlinearity**: tipo de función de activación entre capas, puede ser 'tanh' o 'relu'.
- **bias**: indica si se usan términos de sesgo para las transformaciones lineales.
- **batch_first**: invierte el orden de las dimensiones de entrada/salida, colocando el lote primero.
- **dropout**: aplica abandono entre capas recurrentes para regularización.
- **bidirectional**: permite procesar la secuencia en ambas direcciones, duplicando los valores de salida por paso temporal. (docs.pytorch.org)

Dataset Sunspots

Descripción

El dataset utilizado corresponde al **Daily total sunspot number (versión 2.0)**, disponible en el sitio oficial del **WDC-SILSO, Royal Observatory of Belgium**:

- Sunspot Number Dataset – SIDC

Este archivo contiene el número total diario de manchas solares calculado con la fórmula:

$$R = N_s + 10 \times N_g$$

donde:

- N_s = número de manchas individuales.
- N_g = número de grupos de manchas solares observados en todo el disco solar.

El rango temporal disponible va desde el **1 de enero de 1818** hasta el mes más reciente (con valores provisionales en los últimos 3–6 meses). Antes de 1818 no se incluyen datos diarios, ya que las observaciones eran demasiado escasas y se recopilan solo promedios mensuales o anuales.

Un ejemplo de línea del archivo CSV es:

```
1818;01;01;1818.001; -1; -1.0; 0;1
```

Significado de las columnas

1. **Año** → año de la observación (ejemplo: 1818).
2. **Mes** → mes de la observación (ejemplo: 01).
3. **Día** → día de la observación (ejemplo: 01).
4. **Fecha fraccionaria** → año expresado en formato decimal (ejemplo: 1818.001 representa aproximadamente el 1 de enero de 1818).
5. **Número diario total de manchas solares** → valor positivo indica el número estimado de manchas solares. El valor -1 marca un dato faltante.
6. **Desviación estándar diaria** → dispersión de los valores individuales reportados por distintas estaciones.
7. **Número de observaciones** → cuántas estaciones contribuyeron a calcular el valor diario.
8. **Indicador definitivo/provisional** → 1 indica que el valor es definitivo; 0 significa que es provisional y aún puede ser revisado.

Consideraciones adicionales

- Desde la **versión 2.0 (julio de 2015)**, ya no se utiliza el factor de escala de Zúrich (0.6). En su lugar, la serie se mantiene en la escala de los conteos modernos crudos, para homogeneizar los datos históricos y actuales.
- Los valores de error cambian según la época:
  - Antes de 1981, las desviaciones estándar se estimaron con un modelo autorregresivo basado en la distribución de Poisson.
  - Desde 1981 en adelante, corresponden a la desviación estándar real calculada sobre las observaciones de distintas estaciones.

Rango temporal

El archivo diario v2.0 abarca desde el **1 de enero de 1818** hasta el último mes transcurrido. Los últimos 3 a 6 meses están clasificados como provisionales. Antes de 1818 no existen datos diarios; solo se registran promedios mensuales o anuales debido a la escasez de observaciones.

Utilidad para predicción de series temporales

El conjunto es útil porque:

- Presenta un ciclo solar claro de aproximadamente 11 años que permite estudiar periodicidad y estacionalidad.
- Ofrece resolución diaria, lo que asegura el análisis de dependencias tanto de corto como de largo plazo mediante arquitecturas como RNN, LSTM o GRU.
- Incluye desviación estándar y número de observaciones por día, lo que permite ponderaciones y análisis de incertidumbre en el entrenamiento.

Carga del dataset

Se lee el archivo CSV con separador ; usando pandas. Tras la carga:

- Los valores faltantes marcados con -1 se deben eliminar o imputar antes de la normalización y el entrenamiento.
- Se deben construir ventanas de longitud fija, entendidas como bloques contiguos de observaciones usados como entrada. La longitud de la ventana es un hiperparámetro que determina cuánta historia se aprovecha en cada ejemplo.

Visualización inicial

La serie diaria, o agregada a nivel mensual, se debe graficar para identificar:

- Picos y valles cíclicos con una duración aproximada de 11 años.

- Huecos correspondientes a días con valores -1 en caso de no haber sido tratados.
- Cambios en la varianza, apoyándose en la desviación estándar diaria para identificar valores atípicos.

Normalización de los datos

Antes del entrenamiento:

- Se deben eliminar o imputar los días con -1.
- La normalización debe realizarse mediante Min–Max o Z-score sobre los valores válidos.
- Cuando se utilicen ventanas deslizantes, la normalización debe calcularse únicamente con los datos de entrenamiento y aplicarse luego a validación y prueba, con el fin de evitar fugas de información.

Extra

Ventanas temporales

Las ventanas temporales son subsecuencias consecutivas extraídas de la serie para que la red aprenda dependencias en el tiempo. En este contexto, cada ventana incluye un número fijo de observaciones pasadas que sirven como entrada para predecir el valor siguiente o un conjunto de valores futuros. La longitud de la ventana es un hiperparámetro crítico:

- Ventanas cortas capturan relaciones locales, pero pierden información sobre patrones de largo plazo.
- Ventanas largas abarcan más historia, lo que permite aprender ciclos como el solar (~11 años), aunque incrementan la complejidad del modelo y el riesgo de problemas de gradientes que desaparecen o explotan.

Bidireccionalidad

En PyTorch, al activar bidirectional=True, la RNN procesa la secuencia en dos direcciones: de inicio a fin y de fin a inicio. Esto produce dos vectores ocultos por cada paso temporal, que se concatenan en la salida. En el caso del dataset de manchas solares, la bidireccionalidad ofrece una ventaja al permitir que el modelo incorpore simultáneamente información del pasado y del futuro en el análisis de cada punto de la serie, lo cual puede enriquecer la representación de patrones cíclicos.

> **Nota:** No aplica ya que por no existe inferencia, no añade más información del contenido del pasado de la info.

Dropout

El parámetro dropout introduce aleatoriamente la desactivación de conexiones entre capas durante el entrenamiento. Su función es reducir el sobreajuste, obligando a la red a no depender en exceso de conexiones específicas. En modelos con múltiples capas recurrentes o con ventanas largas, el dropout es fundamental para mejorar la capacidad de generalización, evitando que el modelo memorice el ruido propio de los datos.

> **Nota:** No aplica ya que por defecto tenemos 1 capa para RNN.

Packed Sequences (no aplica)

Cuando las secuencias de entrada tienen longitudes variables (por ejemplo, ventanas de distinta duración debido a la limpieza de datos faltantes), PyTorch ofrece utilidades como pack_padded_sequence y pack_sequence. Estas funciones permiten procesar de manera eficiente las secuencias sin necesidad de rellenarlas artificialmente hasta una longitud común, optimizando la memoria y el cálculo.

> **Nota:** No aplica ya que se usa una ventana de longitud fija.

Referencias

- RNN Documentation (PyTorch)
- LSTM Documentation (PyTorch)
- Dataset Sunspots (WDC-SILSO)
- Información detallada del dataset diario (SILSO)