TALLER ENTREGABLE · REDES RECURRENTES

Tu propio caso con GRU y LSTM

Escoge una base de datos con secuencias, plantea un problema real y resuélvelo con una GRU y una LSTM

Dato	Detalle
Tema	Redes recurrentes: GRU y LSTM (libro Deep Learning de la Biblioteca BIT, Parte IV)
Modalidad	En parejas o individual
Datos	De libre elección, con los requisitos de la sección 3
Herramientas	Python, PyTorch, pandas, NumPy, matplotlib (scikit-learn para la línea base)
Entregables	Un notebook ejecutado y un informe en PDF (máximo 6 páginas), en una sola entrega
Fecha de entrega	_____
Puntaje	100 puntos (rúbrica en la sección 9)

1. De qué se trata

En el libro viste dos casos de redes recurrentes en logística: avisar a tiempo que el refrigerador de un furgón está fallando (GRU) y estimar cuánto falta para que llegue un envío (LSTM). En este taller **tú escoges el caso**: buscas una base de datos donde el orden de los datos importe, planteas una pregunta que le sirva a alguien para decidir algo y la respondes con una GRU y una LSTM, comparadas de forma justa contra una línea base sin red.

Lo que se califica no es que la red gane, sino que el trabajo sea **honesto y riguroso**: datos bien escogidos, partición sin fugas, comparación justa y conclusiones respaldadas con cifras. Si la línea base empata o gana, ese también es un resultado válido, siempre que lo expliques.

2. Objetivos de aprendizaje

1. Reconocer cuándo un problema es secuencial y cuándo una red recurrente tiene sentido.
2. Buscar, evaluar y documentar una fuente de datos real (origen, licencia, calidad).
3. Preparar secuencias para una red recurrente sin fugas de información entre entrenamiento, validación y prueba.
4. Entrenar una GRU y una LSTM con el mismo protocolo y compararlas contra una línea base sin red.
5. Traducir el resultado en una decisión o un impacto concreto para quien usaría el modelo.

3. Escoger el caso y los datos

3.1 Dos tipos de problema permitidos

Tipo	Qué se predice	Ejemplos de temas	Línea base mínima
A. Pronóstico de una serie de tiempo	Uno o varios valores futuros a partir de los valores pasados	Demanda o ventas, consumo de energía, tráfico, calidad del aire, viajes en bicicleta compartida, llegadas a un servicio	Ingenua: el último valor, o el valor del mismo momento del ciclo anterior (ayer, la semana pasada)
B. Clasificación de secuencias	Una categoría para una secuencia completa (o un aviso antes de un evento)	Falla de una máquina a partir de sensores, actividad humana con acelerómetros, sentimiento de un texto, señales médicas o de audio	Clase mayoritaria y un modelo sin orden (por ejemplo, regresión logística sobre promedios de la secuencia o bolsa de palabras)

3.2 Requisitos mínimos de los datos

- **El orden importa.** Si mezclar el orden de los pasos no cambiaría la respuesta, el problema no es para una red recurrente.
- **Tamaño suficiente.** Tipo A: al menos 2 000 pasos de tiempo (o varias series que sumen eso). Tipo B: al menos 1 000 secuencias etiquetadas. Si tienes menos, justifica por qué igual sirve.
- **Fuente verificable.** Autor o entidad, enlace, fecha de consulta y **licencia o condiciones de uso** que permitan usarla en un trabajo académico.
- **Sin datos personales sensibles** sin anonimizar. Si los datos son de una empresa donde trabajas, necesitas permiso y debes quitar todo lo que identifique a personas o clientes.
- **Que no sea un ejemplo de tutorial ya resuelto** (por ejemplo, la serie de pasajeros de aerolínea de 144 meses o MNIST leído fila por fila). Si usas un conjunto muy conocido, tu aporte debe ser claro: otra pregunta, otra decisión de negocio u otro análisis.
- **Se puede reproducir:** el notebook descarga o lee los datos desde un archivo incluido en la entrega o con un enlace estable.

3.3 Dónde buscar

Algunos lugares con datos abiertos. Revisa siempre la licencia de cada conjunto, no la del sitio:

- **datos.gov.co**: portal de datos abiertos de Colombia (movilidad, energía, salud, clima, comercio).
- **UCI Machine Learning Repository**: conjuntos clásicos con descripción y referencia; tiene series de tiempo y datos de sensores.
- **Kaggle Datasets** y **Hugging Face Datasets**: muchos conjuntos de series, sensores y texto; la licencia cambia de uno a otro.
- Portales de datos abiertos de ciudades y entidades (por ejemplo, sistemas de bicicletas compartidas, operadores de energía, redes de calidad del aire).

Antes de escribir código

Llena la **ficha del caso** (sección 7) con tus datos. Si no puedes llenar alguna fila, sobre todo la licencia o la decisión que depende del modelo, busca otra base de datos.

4. Lo que debes hacer

El notebook debe seguir estos pasos, con estos títulos. En cada paso, **explica con tus palabras y con cifras** lo que ves; un resultado sin lectura no suma puntos.

Paso 1. El problema

Quién tiene el problema, qué decisión depende del modelo y qué cuesta equivocarse (en dinero, tiempo, riesgo o personas afectadas). Termina con **la pregunta del caso** en una frase, como en los casos del libro.

Paso 2. Los datos

Origen, autores, licencia, cómo se descargan, tamaño, qué es cada columna, cada cuánto se mide y qué periodo cubre. Indica qué columnas **no** pueden entrar al modelo porque no se conocerían en el momento de predecir.

Paso 3. Análisis descriptivo

Antes de cualquier red: gráficas de la serie o de ejemplos de secuencias de cada clase, valores faltantes, valores extremos, patrones (estacionalidad, tendencia, ciclos) y desbalance de clases. Mínimo **tres gráficas** con su lectura.

Paso 4. Preparación sin fugas

1. **Partición.** Tipo A: en el tiempo (lo más antiguo para entrenar, lo siguiente para validar, lo más reciente para probar). Tipo B: por entidad (máquina, persona, documento), para que la misma entidad no aparezca en dos particiones. Explica tu elección.
2. **Ventanas o secuencias.** Cuánta historia lee la red y qué predice. Justifica el largo con lo que viste en el descriptivo.
3. **Escala.** Con estadísticos calculados **solo con entrenamiento**.
4. **Faltantes.** Qué haces con ellos y por qué.
5. Imprime la forma de los tensores de cada partición: (ejemplos, pasos, variables).

Paso 5. Línea base sin red neuronal

La línea base mínima de tu tipo de problema (sección 3.1), con su métrica en validación. Si se te ocurre una regla mejor, que es la que usaría alguien en la práctica, agrégala.

Paso 6. GRU y LSTM con el mismo protocolo

Una red con una capa recurrente y una capa lineal de salida. La parte de la red se ve así:

```
  self.rnn = nn.GRU(input_size=N_VARIABLES, hidden_size=32, batch_first=True)
  self.salida = nn.Linear(32, N_SALIDAS)   # de la memoria a la predicción
```

`N_VARIABLES` es cuántos datos entran en cada paso y `N_SALIDAS` cuántos valores o clases predice la red.

- Después cambia `nn.GRU` por `nn.LSTM`, con el **mismo tamaño, semilla, datos y protocolo**. Recuerda que la LSTM devuelve su estado final como una pareja (h, c).
- Entrena con la validación: guarda el mejor estado, usa parada temprana y grafica las curvas de pérdida. **La mejor época no puede ser la última permitida.**
- Cuenta los parámetros de las dos redes y explica de dónde sale la diferencia.
- Si cambias el tamaño de la memoria, el largo de la ventana u otra decisión, hazlo **mirando solo la validación** y deja registro de lo que probaste en una tabla.

Paso 7. Evaluación (una sola vez, en prueba)

- Tipo A: MAE (y otra métrica que justifiques) por horizonte, para la línea base, la GRU y la LSTM; error en momentos especiales (picos, festivos, cambios de régimen).
- Tipo B: exactitud, precisión, sensibilidad, F1 por clase y matriz de confusión para la línea base, la GRU y la LSTM.
- Una gráfica de casos concretos: un tramo de la serie con lo real y lo pronosticado, o ejemplos de secuencias bien y mal clasificadas.
- **Análisis de errores:** ¿dónde se equivoca más la mejor red y por qué crees que pasa?

Paso 8. De la predicción a la decisión

Convierte el resultado en algo que entienda quien toma la decisión: pesos ahorrados, horas, alertas a tiempo, falsas alarmas por semana, personas atendidas. Declara tus supuestos de costo en una tabla y justifícalos. Si escoges un umbral, escógelo en validación.

Cierra con **un párrafo para quien tomaría la decisión**, sin términos técnicos: qué recomiendas (línea base, GRU o LSTM), qué gana, en qué casos falla y qué vigilarías si se pone en marcha.

5. Reglas del taller

- Semilla fija para PyTorch y NumPy (por ejemplo, `SEMILLA = 42`).
- **La prueba se usa una sola vez**, al final. Todas las decisiones se toman con validación.
- Ninguna variable que no se conocería en el momento de predecir puede entrar a la red.
- Código documentado: comentarios por bloques numerados y en las líneas importantes. Claridad antes que eficiencia; nada de código que no se explique.
- Todo debe correr con «Reiniciar y ejecutar todo» sin errores, en CPU, en un tiempo razonable (orientativo: menos de 15 minutos). Si necesitas más, reduce los datos y dilo.
- Cada decisión se justifica: por qué estos datos, este largo de ventana, este tamaño de red, esta métrica, estos costos.
- Puedes usar asistentes de IA para programar, pero **debes entender y poder explicar cada línea**. Indica en el informe para qué los usaste.

6. Qué se entrega (una sola entrega)

1. **Notebook ejecutado** (`.ipynb`) con todas las salidas visibles y los 8 pasos de la sección 4.
2. **Informe en PDF** de máximo 6 páginas: la ficha del caso, el problema, los datos, el método, la tabla de resultados, la decisión y las respuestas a las preguntas de reflexión.
3. **Los datos** (si pesan menos de 50 MB) o el código que los descarga, más un archivo `fuente.txt` con el enlace, la fecha de consulta y la licencia.

Nombre de los archivos: `taller_rnn_<apellido1>_<apellido2>.ipynb`, `.pdf` y `.zip`.

7. Ficha del caso (va al inicio del informe)

Campo	Tu respuesta
Nombre del caso	
Tipo de problema (A o B)	
Quién usaría el modelo y para qué decisión	
Fuente de los datos (entidad, enlace, fecha de consulta)	
Licencia o condiciones de uso	
Tamaño (filas, pasos, secuencias, periodo)	
Qué entra a la red en cada paso	
Qué predice la red	
Partición (fechas o entidades)	
Línea base	
Métrica principal y por qué	
Resultado principal en prueba (línea base, GRU, LSTM)	

8. Preguntas de reflexión (responde en el informe)

1. ¿Qué información aprovecha la red recurrente que no aprovecha tu línea base? Muéstralo con un ejemplo concreto de tus datos.
2. ¿Qué fuga de información habría aparecido si partías los datos al azar? ¿Cómo la evitaste?
3. ¿Qué diferencia encontraste entre la GRU y la LSTM? Con tus datos, ¿justifica la LSTM su mayor número de parámetros?
4. ¿Qué tendría que pasar para que tu modelo dejara de servir en la práctica (cambios en los datos, en el negocio, en la medición)? ¿Cómo lo detectarías?
5. ¿Qué limitaciones tienen tus datos y cómo afectan lo que puedes concluir?

9. Rúbrica de calificación

Criterio	Qué se espera para el puntaje completo	Puntos
Elección del caso y los datos	Problema secuencial con una decisión clara; datos que cumplen los requisitos de la sección 3; ficha del caso completa, con fuente y licencia.	15
Análisis descriptivo	Al menos tres gráficas pertinentes con lecturas que citan cifras; faltantes, extremos, patrones o desbalance identificados.	10
Preparación sin fugas	Partición correcta para el tipo de problema; escala con estadísticos de entrenamiento; ventanas justificadas; formas impresas.	15
Línea base	Línea base mínima bien implementada y con su métrica; regla adicional si aplica.	10
GRU y LSTM	Redes correctas; mismo protocolo; mejor estado por validación; curvas; mejor época que no es la última; parámetros contados y explicados.	15
Evaluación y análisis de errores	Métricas adecuadas en prueba para los tres métodos; gráfica de casos concretos; análisis de dónde y por qué falla la mejor red.	15
Decisión e impacto	Resultado traducido a la decisión con supuestos explícitos; umbral (si hay) escogido en validación; párrafo claro para quien decide.	10
Informe y código	Reflexiones respondidas; código documentado que corre de principio a fin; uso de IA declarado.	10
Total		**100**

Penalizaciones: fuga de información que infla el resultado (−20); decidir algo mirando la prueba (−15); datos sin fuente o sin licencia verificable (−10); notebook que no corre de principio a fin (−10); problema que no es secuencial (−10); resultados sin lectura (−2 por resultado, hasta −10).

10. Lista de chequeo antes de entregar

- La ficha del caso está completa, con licencia.
- El notebook corre con «Reiniciar y ejecutar todo» y tiene los 8 pasos con sus títulos.
- La partición es en el tiempo (tipo A) o por entidad (tipo B), y la escala usa solo entrenamiento.
- La prueba se usó una sola vez y la mejor época no fue la última.
- La GRU y la LSTM tienen el mismo tamaño, semilla y protocolo.
- Cada resultado tiene debajo una lectura con cifras.
- El informe tiene la decisión, los supuestos y las respuestas a las cinco preguntas.