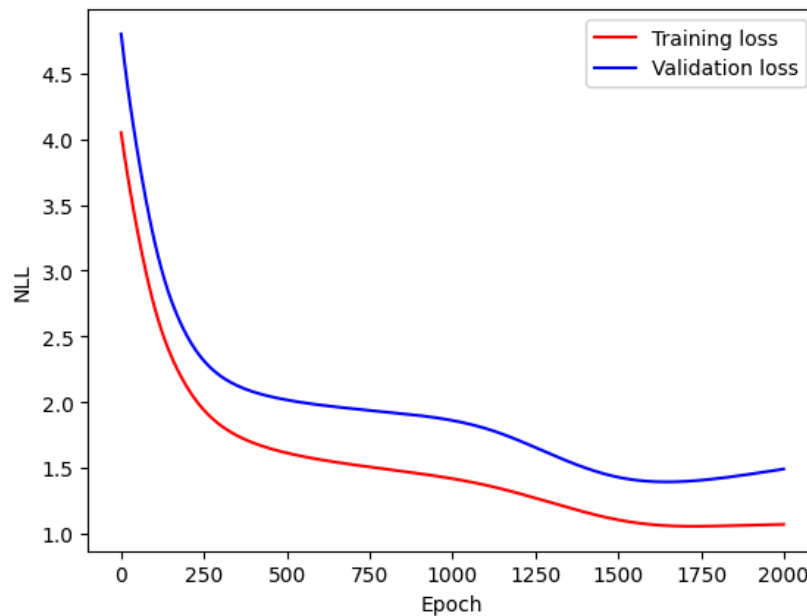


Laboratorio 3

Deep Learning

Repositorio: <https://github.com/TheDeloz-v2/DL-LAB3>

Ejercicio 1 - Construyendo una Red Neuronal Recurrente



1. *¿Qué interpretación le da a la separación de las gráficas de training y validation?*

La separación entre las curvas de pérdida de entrenamiento y validación sugiere cierta divergencia entre cómo el modelo aprende los datos de entrenamiento en comparación con cómo generaliza a nuevos datos. Se observa que la línea de entrenamiento muestra una disminución constante, por otro lado, la línea de validación también disminuye pero se estabiliza y permanece por encima de la de entrenamiento. Esto puede ser una señal de que el modelo está comenzando a sobreajustarse.

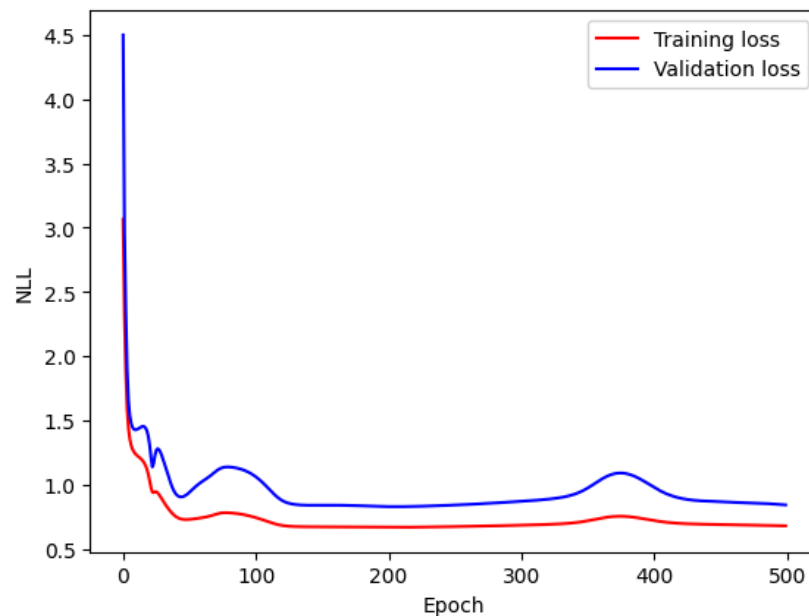
2. *¿Cree que es un buen modelo basado solamente en el loss?*

Como se dijo en el inciso anterior, al observar la estabilidad y la diferencia entre el loss de entrenamiento y validación se puede deducir que probablemente existen problemas de sobreajuste. Sin embargo, aunque las gráficas muestran una tendencia decreciente que es generalmente buena, el loss no captura todo sobre el rendimiento del modelo.

3. *¿Cómo deberían de verse esas gráficas en un modelo ideal?*

En un modelo ideal, las curvas de pérdida de entrenamiento y validación deberían ser relativamente cercanas y seguir una trayectoria similar a lo largo del tiempo. Ambas deberían disminuir y estabilizarse en valores bajos, lo que indica que el modelo no solo aprende bien sino que también generaliza bien a nuevos datos. Idealmente, llegar a converger y estabilizarse en un valor bajo sin una gran brecha entre ellas, demostrando la robustez y efectividad del modelo tanto en aprendizaje como en generalización.

Ejercicio 2 - Construyendo una Red Neuronal LSTM



1. *¿Qué modelo funcionó mejor? ¿RNN tradicional o el basado en LSTM? ¿Por qué?*

El modelo basado en LSTM funcionó mejor que el RNN tradicional. Básicamente se puede ver en el segundo gráfico generado que la pérdida es más baja que en la gráfica del RNN. Es clave mencionar que el LSTM también requirió menos iteraciones, siendo

un cuarto de las necesarias en el RNN, lo que significa que se lograron resultados mejores usando mucho menos recursos. Por lo tanto el modelo LSTM demostró ser más óptimo y eficiente.

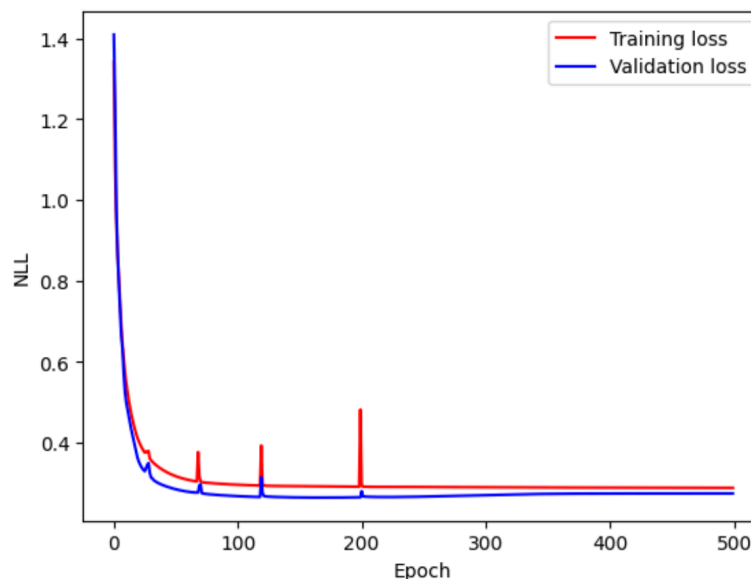
2. *Observen la gráfica obtenida arriba, ¿en qué es diferente a la obtenida a RNN? ¿Es esto mejor o peor? ¿Por qué?*

Lo primero que se observa es el rápido decrecimiento de las pérdidas en el modelo LSTM, tanto la de entrenamiento y validación disminuyen a un ritmo equiparable, luego llega un punto donde ambas se estabilizan y no muestran una brecha significativa, las fluctuaciones son mínimas y, como se mencionó anteriormente, la pérdida final del LSTM es menor.

3. *¿Por qué LSTM puede funcionar mejor con secuencias largas?*

El modelo LSTM funciona mejor con secuencias largas dado que su arquitectura implementa tanto una memoria de estado de celda, como diferentes puertas de entrada, olvido y salida. Estas puertas permiten que el LSTM pueda ir refrescando y mejorando su capacidad de retención de información relevante durante largos periodos de tiempo, mientras que también es capaz de desechar información irrelevante. El procesamiento de secuencias largas genera la necesidad de comprensión de contextos largos, por lo que el hecho de que pueda mantener información a lo largo del tiempo es crucial.

Parte 3 - Red Neuronal LSTM con PyTorch



1. *Compare las gráficas obtenidas en el LSTM "a mano" y el LSTM "usando PyTorch, ¿cuál cree que es mejor? ¿Por qué?*

Comparando las gráficas obtenidas en el LSTM "a mano" y el LSTM "usando PyTorch", se puede observar que la gráfica generada con PyTorch es superior. La principal razón es que en la gráfica de PyTorch, la pérdida de validación es más baja y sigue de cerca a la pérdida de entrenamiento a lo largo de las épocas, lo cual sugiere una mejor generalización del modelo. Además, la curva de entrenamiento y validación en PyTorch es más suave y estable, lo que indica un entrenamiento más robusto y menos susceptible a fluctuaciones y oscilaciones. En contraste, la gráfica "a mano" muestra oscilaciones y un aumento en la pérdida de validación después de ciertas épocas, lo cual podría ser indicativo de sobreajuste. Por lo tanto, el modelo LSTM entrenado usando PyTorch demuestra ser mejor debido a su desempeño más consistente y su capacidad superior para generalizar a nuevos datos.

2. *Compare la secuencia target y la predicha de esta parte, ¿en qué parte falló el modelo?*

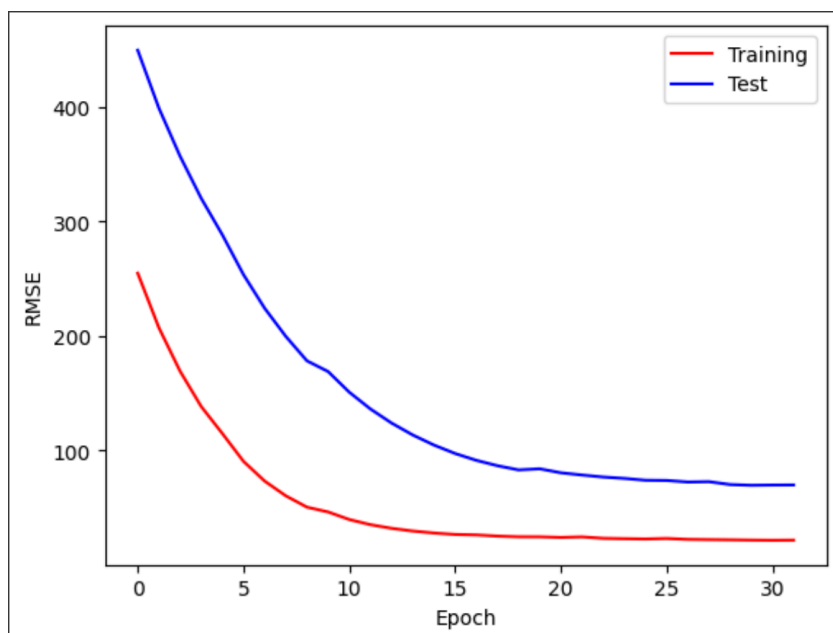
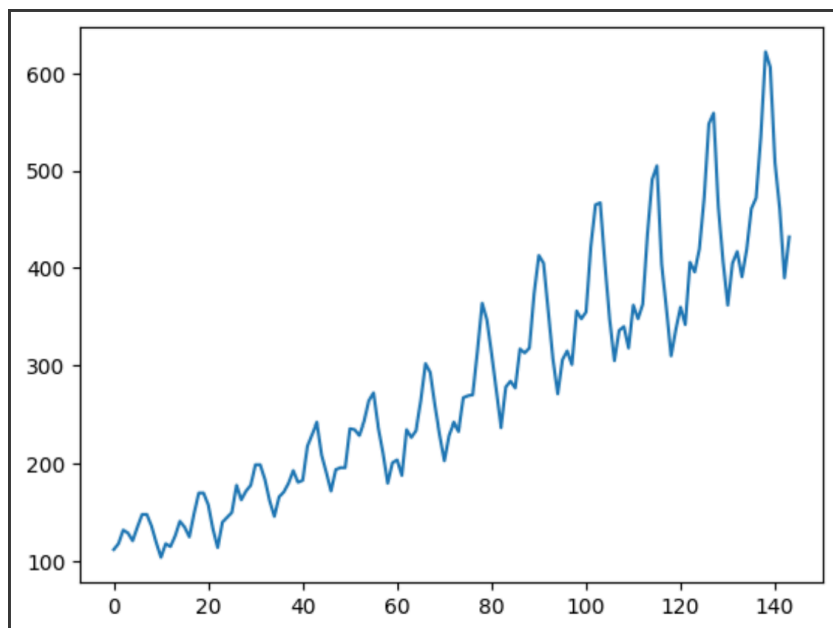
```
Secuencia Input:
['a', 'a', 'a', 'a', 'a', 'a', 'a', 'a', 'a', 'a', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b']
Secuencia Target:
['a', 'a', 'a', 'a', 'a', 'a', 'a', 'a', 'a', 'a', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'EOS']
Secuencia Predicha:
['a', 'a', 'a', 'a', 'a', 'a', 'a', 'a', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'EOS']
```

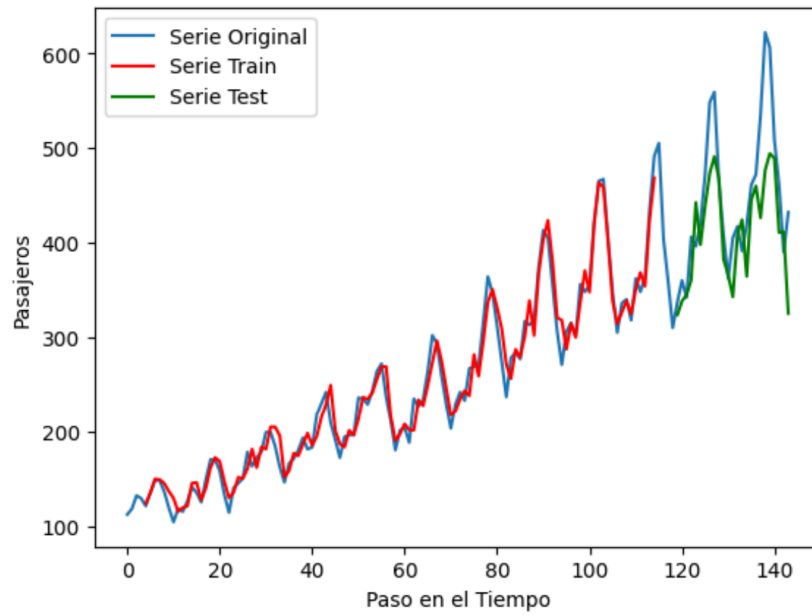
Al comparar la secuencia target y la secuencia predicha, se observa que el modelo falló en las posiciones 9 y 10. En la secuencia target, estas posiciones deberían ser 'a', pero el modelo predijo 'b' en ambas. Exceptuando estos errores, el resto de la secuencia predicha coincide con la target, incluyendo el símbolo de fin de secuencia 'EOS'. Por lo tanto, el fallo del modelo se encuentra en haber cambiado de 'a' a 'b' dos posiciones antes de lo esperado.

3. *¿Qué sucede en el código donde se señala "NOTA 1" y "NOTA 2"? ¿Para qué son necesarias estas líneas?*

En el código, las secciones señaladas como "NOTA 1" y "NOTA 2" son esenciales para alternar entre los modos de evaluación y entrenamiento del modelo neuronal. La línea `net.eval()` (NOTA 1) pone el modelo en modo de evaluación, desactivando funciones específicas de entrenamiento como Dropout y haciendo que Batch Normalization use estadísticas globales en lugar de las del batch actual. Esto garantiza una evaluación precisa del modelo. Por otro lado, la línea `net.train()` (NOTA 2) restaura el modo de entrenamiento del modelo, reactivando Dropout y configurando Batch Normalization para usar estadísticas del batch actual. Este cambio asegura que durante el entrenamiento se apliquen correctamente las técnicas de regularización y normalización, manteniendo la integridad del proceso de aprendizaje.

Parte 4 - Segunda Red Neuronal LSTM con PyTorch





Resultado Final

```
[51] 1
      2 print()
      3 print("La fraccion de abajo muestra su rendimiento basado en las partes visibles de este laboratorio")
      4 tick.summarise_marks() #
      5
```

La fraccion de abajo muestra su rendimiento basado en las partes visibles de este laboratorio

158 / 158 marks (100.0%)

[] 1 Empieza a programar o a [crear código](#) con IA.