Escuela de ingeniería y ciencias.

Campus **Monterrey**

Unidad de formación **TC3006C.102**

Inteligencia artificial avanzada para la ciencia de datos II

Semana 8: Generative Learning

Equipo **3**

Grupo: **102**

Regina Reyes Juárez - **A01275790**
Nadia Salgado Alvarez - **A01174509**
Gilberto Angel Camacho Lara - **A01613895**
Santiago Miguel Lozano Cedillo - **A01198114**

18 de noviembre

Semana 8: Generative Learning

1. Introducción, exploración, explicación y limpieza de datos

El dataset Articles.csv contiene un total de 2,692 instancias y 4 columnas: Article, Date, Heading y NewsType. Cada fila corresponde a una noticia publicada, su fecha, su titular y la categoría a la que pertenece (business o sports).

Durante la exploración verificamos que:

- No existen valores nulos en ninguna columna.
- El dataset está relativamente balanceado:
    - sports: 1,408 instancias
    - business: 1,284 instancias

La columna que usamos para el modelo generativo es Heading, que contiene el titular de cada noticia.
También analizamos la longitud de los titulares, obteniendo:

- Longitud promedio: 8.15 palabras
- Mínimo: 3 palabras
- Máximo: 19 palabras

Esto confirma que los titulares son secuencias cortas, ideales para entrenar un modelo generativo tipo LSTM.
Ejemplos de titulares:

- "sindh govt decides to cut public transport fares..."
- "asia stocks up in new year..."
- "hong kong stocks open 0.66 percent lower..."

Limpieza aplicada:

- Solo conservamos las columnas Heading y NewsType, eliminando columnas irrelevantes para el modelo generativo. También tokenizamos, creamos vocabulario, agregamos tokens especiales <bos>, <eos>, <pad> y <unk> y convertimos los titulares a secuencias numéricas de longitud fija.

Este preprocesamiento prepara los datos para entrenar un modelo de lenguaje con LSTM.

2. Desarrollo del Modelo de Deep Learning

El modelo desarrollado es un modelo de lenguaje basado en LSTM, diseñado para predecir secuencias palabra por palabra. Su arquitectura se compone de una capa de embedding que transforma los índices del vocabulario en vectores densos, seguida de una red LSTM capaz de capturar dependencias temporales en el texto, y finalmente una capa lineal que proyecta cada estado oculto hacia el espacio del vocabulario para producir las predicciones. Durante la implementación se tomaron decisiones intencionales orientadas a garantizar estabilidad y desempeño. En particular, se utilizó inicialización Xavier uniforme tanto en los pesos del embedding como en la capa lineal.

Para evaluar el impacto de la arquitectura y justificar la elección final del modelo, se entrenaron dos configuraciones distintas. El Modelo A, una versión pequeña con embedding de 64 dimensiones, un LSTM de 128 unidades y una sola capa, se utilizó como línea base. Su sencillez permite entrenamientos rápidos y menor riesgo de sobreajuste, pero su capacidad representacional es limitada. En contraste, el Modelo B incrementó la complejidad incorporando embeddings de 128 dimensiones, un LSTM de 256 unidades dividido en dos capas y un dropout de 0.3, lo cual le permite capturar patrones más complejos del lenguaje.

Al comparar ambas configuraciones mediante métricas estándar como loss y perplexity en los conjuntos de validación y prueba, el Modelo B mostró un mejor desempeño general, evidenciando que su mayor capacidad y regularización adicional lograron un modelo más expresivo. Con base en estos resultados cuantitativos y en pruebas cualitativas de generación de texto, se seleccionó el Modelo B como la configuración óptima.

3. Resultados e interpretación

Durante el entrenamiento, ambos modelos mostraron mejoras progresivas, pero con diferencias marcadas en su capacidad para modelar el lenguaje. El Modelo A, de menor tamaño, redujo su pérdida de entrenamiento de 7.35 a 5.54 y alcanzó una perplejidad final de 255.42. En validación, disminuyó la pérdida de 6.39 a 5.84 con una PPL de 346.95, lo que indica que logró capturar patrones básicos del lenguaje, aunque con un límite evidente en su capacidad representacional.

Modelo B, más grande y con dropout, mostró un desempeño superior desde etapas tempranas del entrenamiento. Su pérdida de entrenamiento descendió de 6.82 a 5.40 con una PPL final de 222.76, mientras que en validación redujo la pérdida de 6.30 a 5.79, alcanzando una PPL de 328.29. En el conjunto de prueba obtuvo la perplejidad más baja: 321.92. Estas mejoras sugieren que el aumento en la dimensionalidad de embeddings, el tamaño del estado oculto y la profundidad del LSTM permiten capturar dependencias más complejas del texto, y que el uso de dropout fue fundamental para evitar sobreajuste pese al incremento en parámetros.

Finalmente, la comparación mediante la generación de titulares también confirma la ventaja del Modelo B: aunque aún aparecen tokens <unk> y frases de baja coherencia, esperable en modelos de pequeño tamaño, sus resultados presentan estructuras más estables y secuencias con mayor fluidez sintáctica que las esperadas para el Modelo A. En conjunto, las métricas cuantitativas y la evaluación cualitativa justifican la elección del Modelo B como modelo final, pues ofrece un mejor equilibrio entre capacidad de aprendizaje y generalización.

```
===== Entrenando Modelo A (emb64, h128, 1 capa) =====
Epoch 01 | Train Loss: 7.3541  PPL: 1562.61 | Val Loss: 6.3990  PPL: 601.23
Epoch 02 | Train Loss: 6.2182  PPL: 501.82 | Val Loss: 6.2607  PPL: 523.56
Epoch 03 | Train Loss: 6.0558  PPL: 426.57 | Val Loss: 6.1911  PPL: 488.36
Epoch 04 | Train Loss: 5.9372  PPL: 378.86 | Val Loss: 6.0889  PPL: 440.94
Epoch 05 | Train Loss: 5.8007  PPL: 330.53 | Val Loss: 5.9778  PPL: 394.59
Epoch 06 | Train Loss: 5.6892  PPL: 295.65 | Val Loss: 5.9221  PPL: 373.18
Epoch 07 | Train Loss: 5.6098  PPL: 273.10 | Val Loss: 5.8727  PPL: 355.19
Epoch 08 | Train Loss: 5.5429  PPL: 255.42 | Val Loss: 5.8492  PPL: 346.95

===== Entrenando Modelo B (emb128, h256, 2 capas, dropout) =====
Epoch 01 | Train Loss: 6.8237  PPL: 919.39 | Val Loss: 6.3022  PPL: 545.75
Epoch 02 | Train Loss: 6.0902  PPL: 441.52 | Val Loss: 6.1696  PPL: 477.99
Epoch 03 | Train Loss: 5.8866  PPL: 360.19 | Val Loss: 6.0111  PPL: 407.94
Epoch 04 | Train Loss: 5.7400  PPL: 311.07 | Val Loss: 5.9404  PPL: 380.07
Epoch 05 | Train Loss: 5.6518  PPL: 284.81 | Val Loss: 5.8818  PPL: 358.47
Epoch 06 | Train Loss: 5.5559  PPL: 258.77 | Val Loss: 5.8515  PPL: 347.76
Epoch 07 | Train Loss: 5.4753  PPL: 238.72 | Val Loss: 5.8160  PPL: 335.64
Epoch 08 | Train Loss: 5.4061  PPL: 222.76 | Val Loss: 5.7939  PPL: 328.29
```

```
TITULAR GENERADO 1: summer leads us reach against
TITULAR GENERADO 2: pcb <unk> <unk> in bounce to 1 bln day
TITULAR GENERADO 3: gladiators leaves economy india in on <unk>
TITULAR GENERADO 4: messi shares gas delays to zone last second <unk>
TITULAR GENERADO 5: sbp heat till <unk> online at doping day first
```

4. Conclusión

El desarrollo y evaluación de los modelos LSTM permitieron identificar la configuración más adecuada para la tarea de modelado de titulares. A través de un análisis comparativo entre dos arquitecturas, se observó que el aumento en la dimensionalidad del embedding, la profundidad del LSTM y el uso de dropout mejoraron de manera consistente tanto la perplejidad como la capacidad del modelo para generar secuencias más coherentes. Aunque ambos modelos lograron aprender patrones fundamentales del lenguaje, el Modelo B demostró un desempeño superior en los conjuntos de validación y prueba, además de producir textos de mayor fluidez.

Los resultados confirman que una mayor capacidad expresiva, acompañada de técnicas de regularización adecuadas, es clave para capturar relaciones complejas en tareas de lenguaje natural. Asimismo, el enfoque experimental permitió justificar de manera sólida la selección final del modelo, equilibrando rendimiento y generalización. En conjunto, el proyecto evidencia la efectividad de las arquitecturas LSTM para la generación de texto y

sienta las bases para futuras mejoras, como vocabularios más amplios, modelos más profundos o estrategias avanzadas de generación.

Referencias

- *News articles*. (2017, 30 abril). Kaggle.

  https://www.kaggle.com/datasets/asad1m9a9h6mood/news-articles

- Link a github: https://github.com/santiagolc02/deep_learning