📚 **Resumen de la Clase 4 – RNNs, LSTM, Perplejidad y Generación de Texto**

### **🧩 Contenido Técnico**

#### **1. Redes Neuronales Recurrentes (RNN)**

* Se repasaron las RNN y su capacidad para procesar secuencias gracias a su **estado interno (memoria)**.
* Se vieron tres variantes según la entrada y salida: **many-to-one**, **many-to-many con igual tamaño**, y **many-to-many con distinto tamaño** (e.g. traducción).

#### **2. Limitaciones de las RNN**

* Problemas con **gradientes evanescentes** que impiden recordar información lejana en la secuencia.
* Esto dificulta el aprendizaje de dependencias de largo plazo.

#### **3. LSTM y GRU**

* **LSTM (Long Short-Term Memory)**: introduce puertas (de olvido, entrada, salida) que permiten mantener información relevante en el tiempo.
* **GRU (Gated Recurrent Unit)**: variante más simple y eficiente con resultados similares.

#### **4. Preparación de Datos**

* Se trabajó con cuentos de Borges: se **partieron en oraciones**, se tokenizó por palabras y se aplicó **padding** para igualar longitudes.
* Se construyeron ventanas de contexto fijas y se generaron múltiples ejemplos a partir de cada oración.

#### **5. Evaluación con Perplejidad**

* Se introdujo la **perplejidad** como métrica de evaluación de modelos de lenguaje.
* Representa cuántas palabras son "posibles" para continuar una secuencia según el modelo.
* Se relaciona con la **entropía cruzada**: perplexity = exp(crossentropy).

#### **6. Entrenamiento del modelo**

* Se implementó un modelo secuencial con:  
  + Capa de **embedding**.
  + Dos capas **LSTM**.
  + Capa densa final con **softmax**.
* Se entrenó usando **categorical crossentropy** y el optimizador **Adam**.
* Se calculó la perplejidad en validación con un **callback personalizado**.

#### **7. Generación de Texto**

* Se mostró cómo generar texto con el modelo usando una interfaz con **Gradio**.
* Se probaron diferentes inicios de oración, y el modelo completaba con estilo borgeano.

#### **8. Estrategias de generación**

* **Greedy Search**: elige siempre la palabra más probable.
* **Beam Search**: explora varias opciones a la vez y elige la secuencia más probable.
* **Muestreo con temperatura**: introduce aleatoriedad para lograr mayor creatividad. La **temperatura** controla cuánto se exploran opciones menos probables.

### **📌 Organización del curso**

* Se continúa con la implementación del modelo de predicción de palabras.
* Se anunció que el **desafío próximo** será entrenar un modelo por **caracteres** (no por palabras).
* Se planteó el análisis de cómo definir documentos, oraciones y tokens según el problema específico.
* Se valoró la importancia de elegir bien el **tamaño del contexto** y el **tipo de tokenización**.