📚 **Resumen de la Clase 3 – RNN, LSTM, GRU y Modelos de Lenguaje**

### **🧩 Contenido Técnico**

#### **1. Word Embeddings y transición a secuencias**

* Se repasó el concepto de **word embeddings**, que representan palabras en un espacio vectorial en función de su contexto.
* Se introdujo la necesidad de representar el **orden secuencial** de las palabras para capturar mejor la estructura del lenguaje.

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

* Se explicó que las RNN tienen un **estado interno** que permite procesar información secuencial.
* Se presentaron esquemas de entrada/salida: **many-to-one**, **many-to-many**, etc.
* Las RNN pueden predecir la siguiente palabra, clasificar sentimientos, reconocer entidades, traducir secuencias y más.

#### **3. Problema del Gradiente Evanescente**

* Las RNN sufren de **vanishing gradients**, dificultando el aprendizaje de relaciones a largo plazo.
* Se explicó cómo entrenar secuencias largas equivale a tener redes muy profundas, donde los gradientes se diluyen.

#### **4. RNN Bidireccionales**

* Se presentaron las **RNN bidireccionales**, que procesan la secuencia en ambos sentidos (izquierda→derecha y derecha→izquierda).
* Esto permite incorporar contexto futuro y pasado, útil para tareas como desambiguación de palabras.

#### **5. Implementación en Keras**

* Se mostró cómo implementar una RNN simple en **Keras** con un ejemplo de juguete: predecir la suma de 3 números.
* Se entrenó una red secuencial con una capa RNN y una densa.
* Se exploró cómo hacer la red **bidireccional** con la clase Bidirectional.

#### **6. LSTM – Long Short-Term Memory**

* Se introdujo la arquitectura **LSTM** como solución al gradiente evanescente.
* Se explicaron sus componentes:  
  + **Memoria a corto plazo (H)** y **memoria a largo plazo (C)**.
  + **Compuertas** de olvido, entrada, candidato y salida.
* Cada compuerta filtra y regula qué información se conserva o descarta.

#### **7. GRU – Gated Recurrent Unit**

* Alternativa más simple a LSTM, con **menos parámetros**.
* Tiene dos compuertas: **update** (actualización) y **reset** (reinicio).
* Produce resultados similares a LSTM para muchas tareas, aunque puede fallar en casos donde el orden es crucial.

#### **8. Skip Connections y estabilidad del entrenamiento**

* Tanto LSTM como GRU presentan **skip connections internas**, que facilitan el paso del gradiente en backpropagation.
* Esto permite manejar secuencias más largas de forma estable, superando las limitaciones de las RNN simples.

#### **9. Primer acercamiento a modelos de lenguaje**

* Se introdujo la idea de **modelo de lenguaje**: dada una secuencia de palabras, predecir la siguiente.
* Base de tecnologías como GPT y similares.
* El objetivo es construir secuencias de texto plausibles a partir de un texto semilla, aprendiendo la distribución de probabilidad de palabras.

### **📌 Organización del curso**

* Se trabajó con **Keras** para familiarizarse con su API.
* Se adelantó que el próximo desafío será **entrenar un modelo de lenguaje**.
* La próxima clase se retomará el notebook de modelo de lenguaje y se presentará formalmente el nuevo desafío.