UNIVERSIDAD ADOLFO IBÁÑEZ

Tarea 4-TICS579

Deep Learning 2024-2

Profesor: Alfonso Tobar-Arancibia **Ayudante**: María Alejandra Bravo
Fecha de Entrega: 01/12/24 - 23:59 hrs. **Puntos Totales**: 29

INSTRUCCIONES:

- Entregue la tarea en una carpeta comprimida con el siguiente formato: **Tarea_4_AT_MB**. Donde en este caso, **AT** y **MB** corresponden a las iniciales de cada integrante del grupo. Se deben incluir todos los archivos entregados más los generados por usted.
- El notebook debe entregarse ejecutado con las celdas **en orden** y **sin errores de ejecución**.
- Recuerde que el código debe ser defendido en una sesión de defensa. **NO COPIE Y PEGUE CÓDIGO** que no entiende.
- No cumplir con las instrucciones implica nota 1.0.

Parte 1: Diseño de un Modelo de Machine Translation

Contexto

En una reciente misión de exploración científica en una remota región de Africa, un equipo de lingüistas y antropólogos descubrió una comunidad que utiliza un idioma completamente desconocido, denominado provisionalmente como **Xanith**. Este idioma, caracterizado por una gramática única representa un desafío y una oportunidad para el campo de la lingüística computacional. Debido a su reciente descubrimiento, no existen herramientas tecnológicas para traducirlo ni recursos digitales para su aprendizaje. Afortunadamente se encontraron algunos escritos traducidos por la misma comunidad de libros que existen en el español.
Es por eso que se le solicita a usted diseñar un modelo de Traducción automática que permita tomar frases del español y traducirlas al idioma **Xanith** utilizando una arquitectura Encoder-Decoder utilizando Redes Recurrentes.

Considere los siguientes supuestos:

- Ingresan 3 secuencias de largo 6 al Encoder.
- Cada token se separa por espacios, tanto en español como en **Xanith**. Por simplicidad, no tome en consideración los signos de puntuación. En Xanith, el guión (−) es parte de la palabra y no debe ser eliminado.
- Se considerará el número de palabras más comunes del español, es decir, **10.000**. En el caso del **Xanith**, al ser un idioma mucho más simple, cuenta con cerca de **1.000** palabras.

(a) (2 puntos) Dibuje el esquema de la arquitectura utilizando GRUs. Especifique a su criterio, **todos los hiperparámetros** necesarios para esta tarea.

(b) (1 punto) Se entrega una muestra de cómo sería el set de datos. ¿Qué tokens especiales agregaría para llevar esta tarea cumpliendo todos los supuestos dados? ¿Cómo quedaría la muestra del set de entrenamiento dada en la Tabla 1 luego de agregar dichos tokens?

Español	**Xanith**
Hola, ¿cómo estás?	Ni sa?
Vengo en son de paz	Lo ke da-paz.
Me gustaría aprender Xanith	Lo si Xanith larn.

Tabla 1: Frases traducidas del Español al Xanith.

(c) (2 puntos) Muestre un ejemplo de Forward Pass con una de las frases al momento de entrenar. Explique claramente qué es lo que entra en el Encoder y el Decoder y qué es lo que sale del Decoder.

(d) (2 puntos) Muestre un ejemplo de Forward Pass pero al momento de predecir una nueva frase. Explique claramente qué es lo que entra en el Encoder y el Decoder y el proceso de Predicción Autoregresiva.

(e) (2 puntos) Calcule explicitamente cuáles serían las dimensiones de los tensores de salida de las siguientes partes del modelo: Justifique su respuesta

- Input Embedding
- Output Embedding
- La salida del Encoder
- El último hidden state del Encoder
- La salida del Decoder
- Linear Projection
- Output de la Red (justo antes de entrar al Loss Function)

(f) (3 puntos) Implemente el diseño propuesto en Pytorch creando una clase que herede de `nn.Module` y muestre en pantalla el número de parámetros del modelo. Guarde el código en el archivo **GRU.py**. Con dichos resultados, responda la siguiente pregunta: ***¿Cuál sería el número de Parámetros del Encoder, del Decoder y del Linear Projection?*** **OJO**: No hay que entrenar el modelo.

Parte 2: Bitcoin Forecast

El Bitcoin, como principal representante de las criptomonedas, se caracteriza por su alta volatilidad y dinámica impredecible en los mercados financieros. Esta naturaleza fluctuante presenta tanto oportunidades como riesgos para inversores, empresas y reguladores. Desarrollar un modelo de forecast para predecir su comportamiento es fundamental para mitigar riesgos asociados con inversiones e identificar si efectivamente es posible predecir comportamientos de tal nivel de volatilidad.

Dado su conocimiento en datos secuenciales, se le solicita la creación de una red Neuronal Recurrente que permita estimar el valor del Bitcoin en tiempo cercano a tiempo real (Near Real Time) utilizando el comportamiento de las 2 horas previas.

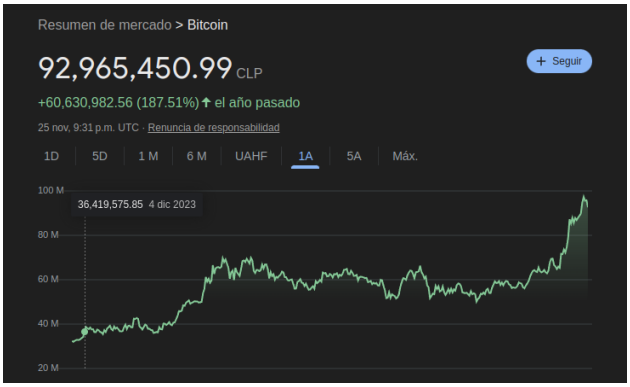

Figura 1: Ejemplo tendencia del Bitcoin

Para ello se le entrega un set de datos real que puede descargar **acá** y un archivo **generate_data.py** que incluye una clase que permite importar datos, crear variables, separar los datos y crear datos secuenciales correspondientes al comportamiento del Bitcoin durante todo el año 2024.

(a) (1 punto) Complete el método `.create_variables()` e implemente cómo obtener variables que permitan a la red determinar estacionalidad a partir de la fecha.

(b) (1 punto) Implemente el método `.split_data()` que permita separar en train y test de manera correcta los datos secuenciales del precio del Bitcoin. El dataset de train son los datos hasta **month** inclusive. El resto va a validación.

(c) (1 punto) Implemente el método `.scale()` aplicando la clase `MinMaxScaler` de Scikit-Learn. El método debe devolver los datos escalados de train y test y la instancia del `MinMaxScaler`.

(d) (1 punto) Implemente el método `.create_sequences()` el cuál debe generar secuencias utilizando una *sliding_window*. Debe recibir como parámetros un Dataframe, una columna que será el target a predecir, y el largo de la secuencia a generar. El objetivo es que tome *sequence_length* registros de todas las variables y se asocie el valor del *target_column* inmediatamente siguiente.

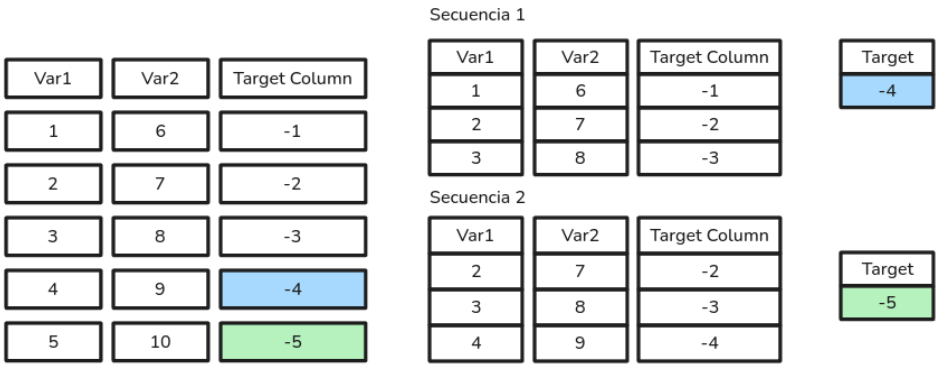

Figura 2: Ejemplo de Cálculo de Secuencias de Largo 3 en un dataset de 5 datos.

Consejo: Entienda bien lo que hace el método `.run()` y qué transformaciones se terminan aplicando a los datos.

(e) (2 puntos) Instancie la clase DataCreator y ejecute el método `.run()`. Entrene con los datos hasta Septiembre (inclusive) y genere secuencias de **2 horas**.

(f) (1 punto) Genere un sólo gráfico (similar al mostrado en la Figura 1) que muestre el comportamiento del Bitcoin. Use colores distintos para el set de entrenamiento y el de validación. **Grafique los datos sin escalar**.

Hint: Los datos sin escalar también pueden ser extraídos de la clase **DataCreator**.

(g) (2 puntos) Cree una red neuronal recurrente de 2 capas que permita recibir una secuencia y predecir el siguiente valor. Para ello utilice una LSTM y defina su `input_size` y `hidden_state`. Utilice el último `hidden_state` como base para la predicción final. Guarde una arquitectura en un archivo **LSTM.py**.

(h) (2 puntos) En el notebook **train.ipynb**, entrene el modelo utilizando una Loss Function apropiada para el problema. Importe los archivos **generate_data.py** y **lstm.py** según corresponda y agregue Datasets, Dataloaders y Training Loop según necesite.

Hint: Utilice el método `.descale()` para regresar las predicciones a la escala original.

(i) (1 punto) Reporte los resultados de su entrenamiento utilizando una métrica apropiada. Genere otra versión del gráfico solicitado en el punto (d). En esta ocasión incluya el set de entrenamiento, el de validación y la predicción generada por el modelo.

Parte 3: Discusión

De acuerdo a sus resultados obtenidos en la sección anterior, responda las siguientes preguntas:

(a) (2 puntos) ¿Qué tan aceptable son sus resultados? Justifique su respuesta haciendo referencia a los resultados reportados en la Parte 2 (g).

(b) (2 puntos) ¿Qué tanto recomendaría el uso de su modelo a una persona que quiere comenzar a invertir en Bitcoin?

(c) (1 punto) ¿Qué estrategias que no haya utilizado en su implementación agregaría a una futura implementación?

OJO: Asocie sus respuestas con sus resultados y con la materia vista en clases. No son preguntas de opinión. **Se esperan respuestas al nivel de un estudiante de Magíster.**