Tarea 2 - Taller de Deep Learning

Fecha de entrega: 14 / 11 / 2023 **Puntaje máximo: 20**

Introducción:
El objetivo de esta tarea es evaluar su conocimiento sobre Deep Learning aplicado a un caso de uso real. En particular, vamos a evaluar la performance de sus modelos en una tarea de **clasificación de preguntas**.

Dataset:
El dataset a ser utilizado consiste de más de **250.000** preguntas de entrenamiento agrupadas en **2** posibles categorías: **sinceras** o **insinceras**.
Pueden descargarlo en el siguiente **link**.
El mismo contiene dos archivos (**train_set.csv** y **test_set.csv**) donde se encuentran las preguntas y su clase correspondiente (1 para preguntas deshonestas, 0 para preguntas honestas).

Tarea:
Tienen total libertad sobre cómo implementar y resolver el problema, así como las técnicas y herramientas que quieran usar, recomendamos usar colab por simplicidad pero pueden implementarlo en sus máquinas si así lo prefieren. La única limitante es que esperamos que la entrega sea en formato **.ipynb** (Jupyter Notebook).

Se espera que implementen los **modelos** utilizados a mano y los entrenen específicamente en este dataset. Pueden implementar arquitecturas preexistentes como las que vimos en clase pero no están permitidos los modelos pre-entrenados. En cambio, tienen total libertad para usar **embeddings** pre entrenados si lo consideran necesario.

Esperamos que las decisiones tomadas sobre el preprocesamiento del texto (transforms, augmentation, etc.) sean resultado de la exploración del dataset y estén propiamente justificadas (una sección de exploración en el notebook con comentarios es suficiente).

Como se mencionó anteriormente, cuentan con un set de datos de test, el cual **no puede ser utilizado para entrenar**, su finalidad es reportar métricas de su solución.
En particular les pedimos que reporten: **accuracy, precision, recall y f1**, todas pueden ser obtenidas usando **sklearn**. Como el dataset es desbalanceado (approx 4x más preguntas sinceras que insinceras) vamos a usar **f1** como métrica principal para comparar.
También se espera poder observar la evolución del modelo (en los datos de train) a medida que se entrena (logs, gráficas, etc).

Guía para Colab:

Descargar el dataset y subirlo a google drive dentro de una carpeta “data”.
Ejecutar el siguiente código en la primera celda del notebook:

```
from google.colab import drive
drive.mount("/content/drive")

! cp "/content/drive/My Drive/data/insincere_questions.zip" .
! unzip -q insincere_questions.zip
! rm insincere_questions.zip
! ls
```

Hacer click en el link, autorizar y pegar el link en el cuadro de texto. Esto debe hacerse cada vez que inicializamos el notebook, pero solo una vez por ejecución.
Deberían quedar dos archivos: **train_set.csv** y **test_set.csv** que pueden ser leídos directamente con pandas.
Cargando el dataset

```
import time
import torch
import itertools
import numpy as np
import pandas as pd
import torch.nn as nn
import matplotlib.pyplot as plt
import torch.nn.functional as F
from torch.utils.data.dataloader import DataLoader
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix

# Global device config
device = torch.device('cuda:0' if torch.cuda.is_available() else 'cpu')

all_train_data = pd.read_csv("train_set.csv")

train, validation = train_test_split(all_train_data,
stratify=all_train_data.target, test_size=0.2)

print(f"{len(train)} Training questions, {len(validation)} Validation questions")
```