UTN FACULTAD REGIONAL MENDOZA
UNIVERSIDAD TECNOLÓGICA NACIONAL

Trabajo Práctico Integrador

Red Neuronal con Aplicación Web

Materia: Redes Neuronales Profundas - Ingeniería en Sistemas de Información

Objetivo del Trabajo Práctico Integrador:

El objetivo de este trabajo es que integren los conocimientos adquiridos sobre redes neuronales desarrollando una aplicación web que incluya un componente inteligente que hayan entrenado ustedes mismos. Con este trabajo, experimentarán con cada una de las etapas del ciclo de vida de un modelo de red neuronal, desde la selección del dataset y el preprocesamiento de los datos hasta la implementación y despliegue de una aplicación web funcional.

Consigna:

Cada grupo (máximo de 4 estudiantes) deberá desarrollar una aplicación web que cumpla con las siguientes características:

1. **Componente Inteligente:** Desarrollen y entrenen una red neuronal que resuelva un problema específico usando un dataset de su elección. El problema puede ser de clasificación, regresión, o cualquier otra aplicación relevante de redes neuronales. Algunos ejemplos podrían ser: clasificación de imágenes, predicción de precios, o reconocimiento de patrones. Como mínimo, deberán realizar un fine-tuning de un modelo preentrenado. No se permite el uso directo de un modelo preentrenado sin modificaciones.
2. **Dataset:** Seleccionen un dataset público que consideren relevante para el problema. Pueden explorar sitios como Kaggle, UCI Machine Learning Repository, o cualquier fuente de datos abierta. El dataset debe ser lo suficientemente interesante como para justificar el uso de una red neuronal, pero no tan complejo como para exceder las capacidades computacionales disponibles.
3. **Entrenamiento de la Red Neuronal:** Desarrollen y entrenen la red neuronal utilizando PyTorch. Documenten su proceso de selección del modelo, ajuste de hiperparámetros y evaluación del rendimiento. Se aceptan fine tuning de modelos preentrenados, pero no la utilización directa de los mismo.
4. **Aplicación Web:** Implementen una aplicación web simple que permita interactuar con el modelo entrenado. La interfaz debe ser desarrollada usando Streamlit y debe ser funcional, sin necesidad de un diseño visual avanzado o de UI/UX complejo.
5. **Hosting:** La aplicación debe ser desplegada en un servicio gratuito de hosting. Recomendamos usar Streamlit Cloud para el despliegue. Estas plataformas permiten el hosting gratuito de aplicaciones pequeñas y son adecuadas para principiantes.

Entregables:

1. **Primera Semana:**
    - **Elección del dataset y definición de la aplicación a desarrollar.**
    - **Entregable:** Documento con la descripción del problema, el dataset seleccionado y la aplicación propuesta. Presentar tres opciones, de las cuales el profesor elegirá la más adecuada teniendo en cuenta los tiempos del proyecto.
    - **Fecha de entrega: miércoles 28 y jueves 29 de mayo de 2025**.
2. **Segunda Semana:**
    - **Creación del dataset en PyTorch (incluyendo preprocesamiento y preparación de los datos).**
    - **Entregable:** Dataset preprocesado y archivo de código de preparación de datos, documentado en un notebook.
    - **Fecha de entrega: miércoles 4 y jueves 5 de junio de 2025**.
3. **Tercera Semana:**
    - **Entrenamiento de la red neuronal y selección del mejor modelo.**
    - **Entregable:** Modelo entrenado junto con los resultados de evaluación y justificación de la selección del mejor modelo, documentado en un notebook.
    - **Fecha de entrega: miércoles 11 y jueves 12 de junio de 2025**.
4. **Cuarta Semana:**
    - **Desarrollo de la interfaz y despliegue de la aplicación en Streamlit Cloud.**
    - **Entregable:** Proyecto completo en un repositorio de GitHub, incluyendo la aplicación desplegada y la URL de acceso. Para más detalles, consulta la sección **Conformación del Repositorio**.
    - **Fecha de entrega: miércoles 18 y jueves 19 de junio de 2025**.
5. **Presentación Final:**
    - Cada grupo deberá preparar una presentación en PowerPoint de 15 minutos explicando los pasos realizados durante las cuatro semanas del proyecto. La presentación debe incluir:
        - Una explicación detallada de cada etapa del desarrollo (dataset, creación, entrenamiento, interfaz y despliegue).
        - Una demostración en vivo de la aplicación desplegada en Streamlit Cloud, mostrando la funcionalidad principal.
    - **Fecha de presentación: miércoles 25 de junio de 2025**.

Conformación del Repositorio:

El repositorio de GitHub del proyecto debe seguir la siguiente estructura:

1. **data/**: Contendrá los datasets usados para entrenar y evaluar la red neuronal. Podría incluir archivos de datos preprocesados (e.g., `training_data.csv`, `test_data.csv`) y scripts relacionados con la preparación de los datos.
2. **dev/**: Esta carpeta se usará para los notebooks y scripts de desarrollo experimental del modelo. Aquí se incluirán los experimentos en Jupyter notebooks

(`model_dev.ipynb`) y cualquier script que documente el proceso de exploración y pruebas.

3. **prod/**:
    - **app.py**: Archivo principal para la aplicación web. Este script contendrá el código para crear la interfaz gráfica con Streamlit y usar la aplicación.
    - **modelo.pth**: Archivo del modelo entrenado que contiene los pesos de la red neuronal en formato PyTorch.
    - **README.md**: Documento de descripción del proyecto que incluye instrucciones claras sobre cómo clonar el repositorio, instalar las dependencias y ejecutar la aplicación.
    - **requirements.txt**: Archivo con las dependencias necesarias para ejecutar el proyecto (e.g., PyTorch, Streamlit).
    - **utils.py**: Archivo con funciones auxiliares utilizadas por `app.py`, tales como preprocesamiento de datos y carga del modelo.

Ejemplo guía:

Se adjuntan enlaces al repositorio en github y a la aplicación desplegada para un ejemplo de traductor automático de inglés a español.

También hay ejemplos de años anteriores como este generador de gatitos y este detector de letras en lenguaje de señas.