

## Datos del estudiante

Nombre y apellidos

Fecha de entrega

## Actividad grupal: Trabajando con redes neuronales y *Deep Learning*

### Objetivos de la actividad

Esta actividad permitirá al equipo de trabajo profundizar en la aplicación de técnicas de aprendizaje supervisado (regresión y clasificación) basadas en redes neuronales utilizando las librerías Keras y TensorFlow sobre Python. Para ello, el grupo habrá de elegir dos *dataset* en plataformas *online* como Kaggle, OpenML o Google Dataset Search en base a unos requisitos mínimos especificados y aplicar, en un primer caso, una red neuronal para la regresión comparada con otra técnica de regresión no basada en redes neuronales y, en un segundo caso, otra red neuronal diferente para la clasificación, de nuevo comparada con otra técnica de clasificación no basada en redes neuronales.

### Pautas de elaboración

- ▶ En primer lugar, el grupo deberá repasar los siguientes contenidos teórico-prácticos de la asignatura:
  - Tema: Python para la implementación de técnicas de inteligencia artificial, haciendo especial hincapié en las secciones: Librerías útiles para el análisis de datos (apartados Keras, TensorFlow y Pandas); Importación de datos; Introducción a *Machine Learning* con librerías en Python.
  - Tema: Redes neuronales artificiales, especialmente: Aplicaciones y ejemplos de implementación.
  - Tema: *Deep Learning*, especialmente: Ejemplos de implementación.

- ▶ El anterior repaso y la ejecución de los ejemplos incluidos en los diferentes apartados permitirá al grupo asegurar que cuenta con el entorno de ejecución de Python y con todas las librerías necesarias para llevar a cabo esta actividad, además de haber realizado el ejemplo básico con TensorFlow 2.0, así como el tutorial oficial de dicha librería (secciones 2.6 y 6.12). Del mismo modo, permitirá asegurar que se está familiarizado con los conceptos de regresión lineal y regresión logística (recuerda, que, en este último caso, se trata de un clasificador).
- ▶ Se recomienda encarecidamente, además, al menos seguir los dos siguientes tutoriales oficiales de TensorFlow 2.0:
  - TensorFlow 2.0: Conceptos básicos de AA con Keras: Regresión (<https://www.tensorflow.org/tutorials/keras/regression?hl=es-419>).
  - TensorFlow 2.0: Conceptos básicos de AA con Keras: Clasificación de imágenes básica.
- ▶ **En ambos** apartados el grupo ha de utilizar de utilizar, **al menos**, las siguientes librerías:
  - **Pandas** para la importación de datos.
  - **Keras** sobre **TensorFlow 2.0** ([tensorflow.keras](https://www.tensorflow.org/tutorials/keras)).
  - El grupo tendrá total libertad para usar otras librerías para trabajar con otros modelos además de con los de redes neuronales (p.ej., scikit-learn), para la muestra de gráficas (p.ej., matplotlib o seaborn) o para otras operaciones auxiliares, si así lo necesitaras.

**Apartado A (Regresión)**

- ▶ Comenzamos con el ejercicio de regresión. En primer lugar, el grupo habrá de elegir un *dataset* de acceso público y disponible *online*. El *dataset* ha de ser apropiado para aplicar diferentes técnicas de regresión y compararlas entre sí.
- ▶ Se puede elegir un *dataset* en Kaggle, OpenML o Google Dataset Search (es un metabuscador, de modo que puede llevar igualmente a los anteriores). Otras fuentes de datos son posibles, siempre que sean de acceso público y estén disponibles *online* sin necesidad de autenticarse, pues se habrá de indicar dicha fuente y se comprobará su accesibilidad para evaluar la actividad.
- ▶ **El *dataset* escogido ha de contar, como mínimo, con 1000 instancias y el grupo deberá trabajar con todas ellas. Se debe elegir la fracción que se emplea para entrenar los modelos y la fracción que se utiliza para el test de los mismos, siempre que se haga uso en total de al menos 1000 instancias en todos los modelos.**
- ▶ El problema de regresión a resolver contará con, al menos, una variable/atributo (obviamente numérico) a la salida, y **que dependa de, al menos, 6 variables/ atributos de entrada.**
- ▶ Se deberá describir el conjunto de datos utilizado y explicar el problema de regresión a resolver (no son necesarios contenidos teóricos, sino explicar qué relaciones tratamos de comprobar y con qué métodos).
- ▶ Utilizar la librería **pandas** para la importación del *dataset*. **No es posible adjuntar ningún archivo adicional.** El propio código Python del archivo aportado ha de ingerir/descargar automáticamente los datos de entrenamiento y test correspondientes y trabajar sobre ellos. Es importante **preprocesar los datos si fuera necesario.**
- ▶ Mediante Python y las librerías que el grupo considere, analizar el *dataset* proporcionando una caracterización del mismo, mostrando **al menos** algunas de

sus características en modo texto (mediante tablas o prosa) y **al menos** algunas de ellas en modo gráfico (p.ej., histogramas, diagramas de dispersión, diagramas de cajas y bigotes, etc.). **Las características y las gráficas incluidas han de provenir de la ejecución del código en Python que se aporte como respuesta.**

- ▶ **Normalizar los datos si fuera necesario.**
- ▶ **El grupo deberá elegir al menos un método de regresión no basado en redes neuronales** (por ejemplo: regresión lineal, regresión polinómica, regresión logarítmica, SVR, *random forest regression*, etc.).
- ▶ **Se deberá incluir en el informe una explicación de los parámetros que se consideren relevantes en cada ejecución.**
- ▶ Mediante Python y las librerías utilizadas, entrenar el modelo o modelos escogidos, dividiendo el *dataset* en datos de entrenamiento y datos de test previamente en base al criterio grupal y probar los mismos frente a dichos datos de test.
- ▶ **Elegir al menos una arquitectura de red neuronal (describe en el informe las neuronas en la capa de entrada, las capas intermedias —al menos dos capas intermedias— y capa de salida, funciones de activación en cada caso) que permita realizar una regresión.**
- ▶ Mediante Python y utilizando **al menos Keras sobre TensorFlow 2.0 (tensorflow.keras)**, se deberá entrenar el modelo o modelos de red neuronal escogidos, probándolo con los datos de test.
- ▶ Mediante Python y las librerías que el grupo considere, se deberán **mostrar los resultados** obtenidos por los diferentes algoritmos escogidos de forma gráfica y comparada/superpuesta.

- ▶ El grupo deberá consensuar en el informe una **discusión sobre los resultados obtenidos**, argumentando con qué técnica se obtienen mejores resultados en base a las diferentes métricas escogidas y **explicando cómo se podrían mejorar los resultados obtenidos por las redes neuronales**, independientemente de que mejoren o no a los algoritmos no basados en redes neuronales.
  
- ▶ **El grupo deberá incluir en el informe al menos lo siguiente:**
  - Descripción de la fuente de datos empleada.
  - Caracterización del *dataset* utilizado en modo texto y gráfico.
  - Parámetros relevantes utilizados en los diferentes algoritmos (en el caso de las redes neuronales, su arquitectura, al menos).
  - Resultados obtenidos por los diferentes algoritmos escogidos de forma gráfica y comparada/superpuesta.
  - Discusión de los resultados obtenidos y argumentos sobre cómo mejorar de dichos resultados.

### Apartado B (Clasificación)

- ▶ Seguimos, de forma análoga, con el ejercicio de clasificación. En primer lugar, el grupo elegirá un *dataset* de acceso público y disponible online. El *dataset* ha de ser apropiado para aplicar diferentes técnicas de clasificación y compararlas entre sí.
  
- ▶ Se puede elegir un *dataset* en Kaggle, OpenML o Google Dataset Search. De nuevo, otras fuentes de datos son posibles, siempre que sean de acceso público y estén disponibles online sin necesidad de autenticarse, pues se habrá de indicar dicha fuente y se comprobará su accesibilidad para evaluar la actividad.
  
- ▶ **El *dataset* escogido ha de contar, como mínimo, con 1000 instancias y el grupo deberá trabajar con todas ellas. Se deberá consensuar qué fracción se emplea para entrenar los modelos y la fracción que se usa para el test de los mismos, siempre que se haga uso en total de al menos 1000 instancias en todos los modelos.**

- ▶ El problema de clasificación a resolver contará con, al menos, una variable categórica de **al menos 5 clases posibles a la salida, y que dependa de, al menos, 6 variables/atributos de entrada.**
- ▶ El grupo deberá describir el conjunto de datos utilizado y explicar el problema de clasificación a resolver (no son necesarios contenidos teóricos, sino explicar qué relaciones tratamos de comprobar y con qué métodos).
- ▶ Utilizar la librería **pandas** para la importación del *dataset*. **No es posible adjuntar ningún archivo adicional.** El propio código Python del archivo aportado ha de ingerir/descargar automáticamente los datos de entrenamiento y test correspondientes y trabajar sobre ellos. **Se deberán preprocesar los datos si fuera necesario.**
- ▶ Mediante Python y las librerías que el grupo considere, se deberá analizar el *dataset* proporcionando una caracterización del mismo, mostrando **al menos** algunas de sus características en modo texto (mediante tablas o prosa) y **al menos** algunas de ellas en modo gráfico (por ejemplo: histogramas, diagramas de dispersión, diagramas de cajas y bigotes, etc.). **Las características y las gráficas incluidas han de provenir de la ejecución del código en Python que se aporte como respuesta.**
- ▶ **Se deberán normalizar los datos si fuera necesario.**
- ▶ **El grupo deberá consensuar al menos un método de clasificación no basado en redes neuronales** (por ejemplo: regresión logística, árboles de decisión, reglas de clasificación, *random forest*, SVM, etc.).
- ▶ **Incluir en el informe una explicación de los parámetros que el grupo considere relevantes en cada ejecución.**
- ▶ Mediante Python y las librerías que se consideren, se deberá entrenar el modelo o modelos escogidos, dividiendo el *dataset* en datos de entrenamiento y datos

de test previamente en base al criterio consensuado por el grupo, y probar el modelo frente a dichos datos de test.

- ▶ **El grupo deberá elegir al menos una arquitectura de red neuronal (Se deberá describir en el informe:** las neuronas en la capa de entrada, las capas intermedias – **al menos dos capas intermedias** – y capa de salida, funciones de activación en cada caso – **al menos utilizar *relu* en algunas de las capas intermedias y utilizar *softmax* en la capa de salida**) de forma que se permita realizar una **clasificación**.
- ▶ Mediante Python y utilizando **al menos Keras sobre TensorFlow 2.0 (`tensorflow.keras`)**, se deberá entrenar el modelo o modelos de red neuronal escogidos, dividiendo el *dataset* en datos de entrenamiento y datos de test previamente en base al criterio consensuado por el grupo y probar los modelos frente a dichos datos de test.
- ▶ Mediante Python y las librerías que el grupo considere, **mostrar los resultados** obtenidos por los diferentes algoritmos escogidos de forma gráfica y comparada/superpuesta.
- ▶ El informe incluirá una **discusión sobre los resultados obtenidos**, donde el grupo deberá argumentar con qué técnica se obtienen mejores resultados en base a las diferentes métricas seleccionadas y **explicar cómo se podrían mejorar los resultados obtenidos por las redes neuronales**, independientemente de que mejoren o no a los algoritmos no basados en redes neuronales.
- ▶ **El informe a desarrollar por el grupo deberá incluir al menos lo siguiente:**
  - Descripción de la fuente de datos empleada.
  - Caracterización del *dataset* utilizado en modo texto y gráfico.
  - Resultados obtenidos por los diferentes algoritmos escogidos de forma gráfica y comparada/superpuesta.
  - Discusión de los resultados obtenidos y argumentos sobre cómo mejorar de dichos resultados.

- **NOTA:** No está permitido entregar ninguno de los ejemplos ya resueltos en clase (ya sean de teoría o de refuerzo). Tampoco está permitido el plagio entre trabajos del mismo curso o cursos anteriores. Tampoco está permitido el plagio de fuentes de Internet, libros, manuales o de cualquier otro tipo que no se citen apropiadamente. **Entregar un ejercicio que incurra en los anteriores escenarios implicará automáticamente un 0 como calificación de la actividad, sin perjuicio de que se tomen medidas disciplinarias de acuerdo con la correspondiente normativa de la UNIR.** Está permitido utilizar código, fracciones de código o ejemplos existentes en Internet u otras fuentes **siempre que se citen apropiadamente y su uso se limite a partes minoritarias de la actividad.**

### Extensión y formato

Extensión máxima de la actividad: un único informe de 8 páginas como máximo de extensión (Arial 11, interlineado 1,5) y dos archivos Python, uno para el problema de regresión y otro para el problema de clasificación. Ambos archivos han de poder ejecutarse de forma independiente el uno del otro. El informe no debe incluir anexos, portada ni repetir el enunciado o la rúbrica de la actividad. Cualquier página que el grupo incluya será computada dentro de la extensión entregada. No se evaluará ninguna página a partir más allá de la octava. No es posible adjuntar ningún archivo adicional. El propio código Python de los dos archivos ha de ingerir/ descargar automáticamente los datos de entrenamiento y test correspondientes a cada uno y trabajar sobre ellos.

### Organización y gestión de equipos

En el foro correspondiente de la asignatura, el alumno encontrará un nuevo tema específico para la organización de equipos donde el profesor explicará todos los detalles.

Una vez cerrado el equipo de trabajo el grupo se puede comenzar a poner en contacto a través de las cuentas @comunidadunir.net e iniciar los trabajos. Se



puede ampliar la información sobre el trabajo en equipo, consultando los [Tutoriales de trabajo en grupo](#).

**IMPORTANTE:** Aquellos **estudiantes que no comiencen su trabajo dentro de los 7 primeros días**, contados a partir del día de inicio de la actividad, **quedarán excluidos** de la actividad, no pudiendo tomar parte en ella. Se trata de una actividad colaborativa, por lo que unos estudiantes no pueden beneficiarse del trabajo que hayan realizado sus compañeros. **No se permiten realizar el trabajo individualmente.**

### Entrega de la actividad grupal

Al finalizar la actividad grupal, todos los miembros del equipo entregarán la misma actividad a través del apartado «Envío de actividades» del aula virtual. El documento a entregar debe ir nombrado así:

APELLIDO1\_APELLIDO2\_NOMBRE\_Titulo\_actividad (sin tildes ni apóstrofes ni ningún otro carácter que pudiera resultar conflictivo).

Todos los miembros del equipo deben hacer la entrega en el aula virtual y deben adjuntar el mismo documento.

Cada alumno del grupo deberá indicar en la actividad el nombre de todos los componentes del mismo, y cumplimentar la siguiente tabla de valoración individual:

	Sí	No	A veces
Todos los miembros se han integrado al trabajo del grupo			
Todos los miembros participan activamente			
Todos los miembros respetan otras ideas aportadas			
Todos los miembros participan en la elaboración del informe			
Me he preocupado por realizar un trabajo cooperativo con mis compañeros			
Señala si consideras que algún aspecto del trabajo en grupo no ha sido adecuado			

