

Instrucciones:

- Esta es una actividad en grupos de no más de 3 integrantes.
 - 0 Recuerden **unirse al grupo de canvas**
- No se permitirá ni se aceptará cualquier indicio de copia. De presentarse, se procederá según el reglamento correspondiente.
- Tendrán hasta el día indicado en Canvas.
 - 0 No se confíen, aprovechen el tiempo en clase para entender todos los ejercicios y avanzar lo más posible.

REPOSITORIO DE TASKS 2: <https://github.com/Javilejoo/Lab-3-IA>

Task 1 - Preguntas Teóricas

Responda a cada de las siguientes preguntas de forma clara y lo más completamente posible.

1. **Explique la diferencia entre descenso de gradiente, descenso de gradiente por mini batches y descenso de gradiente estocástico. Asegúrese de mencionar las ventajas y desventajas de cada enfoque.**

El método de **Descenso de Gradiente** utiliza el conjunto completo de datos de entrenamiento para calcular el gradiente de la función de costo. Luego, actualiza los parámetros del modelo en una sola iteración.

Ventajas:

- Convergencia más estable, ya que utiliza información precisa del conjunto de datos completo.
- Puede beneficiarse de optimizaciones numéricas al procesar grandes cantidades de datos simultáneamente.

Desventajas:

- Puede ser computacionalmente costoso, especialmente para conjuntos de datos grandes.
- No es práctico para conjuntos de datos que no caben en la memoria.

El **Descenso de Gradiente por Mini Lotes** divide el conjunto de datos de entrenamiento en pequeños lotes y realiza actualizaciones de parámetros en cada uno de estos lotes. Combina características del descenso de gradiente y el descenso de gradiente estocástico.

Ventajas:

- Mejora la eficiencia computacional al procesar varios ejemplos simultáneamente.
- Aprovecha la capacidad de cómputo de hardware moderno al paralelizar cálculos.
- Menos propenso a quedarse atascado en mínimos locales que el descenso de gradiente por lotes.

Desventajas:

- Requiere ajuste de hiperparámetros, como el tamaño del lote.
- No es tan eficiente como el descenso de gradiente estocástico para datos muy grandes.

El **Descenso de Gradiente Estocástico** actualiza los parámetros del modelo después de cada ejemplo de entrenamiento. Selecciona aleatoriamente ejemplos del conjunto de datos para cada actualización de parámetros.

Ventajas:

- Eficiente para conjuntos de datos grandes, ya que no requiere cargar todo el conjunto de datos en memoria.
- Puede escapar de mínimos locales y converger hacia el mínimo global debido a la aleatoriedad en la selección de ejemplos.

Desventajas:

- Variabilidad en las actualizaciones puede llevar a oscilaciones en el proceso de convergencia.
- Menos eficiente en términos computacionales que el descenso de gradiente por lotes debido a las actualizaciones más frecuentes.

(What is gradient descent? | IBM, s. f.)

2. Compare y contraste técnicas de extracción de features (feature extraction) y selección de features (feature selection) en machine learning. De ejemplos de escenarios donde cada técnica sería más apropiada.

El **feature selection** consiste en elegir un subconjunto relevante de características existentes para evaluar y seleccionar features según la importancia del modelo.

Las técnicas de feature selection son:

- **Filtros:** Selecciona características basadas en medidas estadísticas (p. ej., correlación).
- **Wrappers:** Utiliza el rendimiento del modelo para evaluar combinaciones de características.
- **Incrustados (Embedded):** Realiza la selección de características durante el proceso de entrenamiento del modelo.

(Brownlee, 2020)

Los **features de extracción** implican la creación de nuevos features en base a las existentes para buscar representaciones más compactas.

Algunos ejemplos de feature extraction son:

- **PCA (Análisis de Componentes Principales):** Transforma variables originales en componentes no correlacionados.
- **Extracción Manual:** Identificación y creación manual de características relevantes.
- **Extracción Automatizada:** Utiliza algoritmos o redes profundas para aprender nuevas representaciones.

(Feature extraction explained, s. f.)

Ejemplos Prácticos:

Feature Selection: Para clasificar textos en categorías poder usar este método para eliminar palabras que no son de interés

Feature Extraction: Reconocer rostros en imágenes se puede usar la extracción para crear características que represente una combinación de píxeles para la detección facial.

3. Describa la arquitectura y el funcionamiento de un perceptrón de una sola capa (un tipo de red neuronal sin backpropagation). Explique cómo aprende y la forma en la que actualiza sus parámetros.

Un perceptrón es una neurona artificial, indispensable para las redes numerales del Deep Learning. Es un modelo matemático que tiene la estructura y funcionalidad de una neurona biológica.

Un perceptrón con capa única puede enseñar únicamente funciones lineales separables.

Arquitectura:

Entrada: X_1, X_2, X_N : denotan los valores del perceptrón de la característica.

Pesos: W_1, W_2, W_N : Ofrecen un valor preliminar en el inicio de aprendizaje.

Suma Ponderada: Es la proliferación de cada valor de entrada o característica

Función de activación: Toma un único número y realiza una determinada operación matemática fija sobre él

Salida: La suma ponderada se pasa a la función de activación y cualquier valor que obtengamos después del cálculo es nuestra salida predicha.

(Team, 2023)

Referencias:

- *What is gradient descent?* | IBM. (s. f.). <https://www.ibm.com/mx-es/topics/gradient-descent>
- *Feature extraction explained.* (s. f.). MATLAB & Simulink. <https://www.mathworks.com/discovery/feature-extraction.html#:~:text=What%20is%20Feature%20Extraction%3F,directly%20to%20the%20raw%20data.>
- Brownlee, J. (2020, 20 agosto). *How to choose a feature selection method for machine learning.* MachineLearningMastery.com. <https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/>
- Team, D. (2023, 30 octubre). *Perceptron : Qu'est-ce que c'est et à quoi ça sert ?* Formación en ciencia de datos | DataScientest.com. <https://datascientest.com/es/perceptron-que-es-y-para-que-sirve>

Task 2 - Ejercicios Prácticos

Use Python en Jupyter Notebook para resolver los siguientes problemas relacionados a los temas que se vieron en clase. Asegúrese de citar todas las fuentes que utilice.

Task 2.1 - Gradiente Descendiente Estocástico

Implemente el descenso de gradiente estocástico, el descenso de gradiente y el descenso de gradiente por mini batches para una función polinómica de grado 3. Luego, grafique la función aproximada por cada uno de los métodos solicitados y la distribución real de puntos. Para esto considere que:

- El polinomio que debe usar es $2 * x^{**3} - 3 * x^{**2} + 5 * x + 3$
- Considere el uso de `np.polyval`
- Considere el uso de `np.polyder`
- Compare el tiempo de ejecución de cada uno de los métodos solicitados. De esto, mencione cuál método fue más rápido y por qué.
- Compare el fitness de cada método solicitado (recuerde hacer una evaluación numérica y no sólo visual). De esto mencione cuál método lo hizo mejor y por qué

Task 2.2 - Feature Selection

Tome de nuevo el dataset del laboratorio pasado referente a la clasificación de Partidas de League of Legends. Recuerde que lo puede descargar de este [enlace](#). Sobre este aplique **al menos 3 técnicas de feature selection** distintas, y con el dataset resultante vuelva a ajustar el modelo de Support Vector Machine (de la librería Sklearn). Recuerde que:

- Debe definir una métrica de desempeño y justificar el por qué de esta métrica
- Mida el performance de su modelo con las tres técnicas de feature selection y compare cada una de ellas. Es decir, debe tener al menos 4 valores a comparar (el obtenido en el laboratorio pasado, y al menos 3 de las técnicas de feature selection)
- De las medidas obtenidas, diga qué versión lo hizo mejor y por qué.
- Recuerde que para hacer parameter tuning, deben usar las mismas columnas seleccionadas después del feature selection
- Recuerde que su variable objetivo es "blueWins"

Task 2.3 - Perceptrón

Implemente un perceptrón de una sola capa para realizar una clasificación binaria del dataset de Iris (incluido en `sklearn.datasets.load_iris`). Visualice la frontera de decisión que fue aprendida por el perceptrón. Recuerde que:

- Debe cargar el dataset de Iris desde la librería mencionada
- Use solamente **dos** features, siendo estos *sepal length* y *sepal width*

Inteligencia Artificial - Laboratorio 3 -

- Asegúrese de que su clase *Perceptron* tenga los métodos de *fit* y *predict*
- Visualice la frontera de decisión
 - El eje Y debería ser sepal width y el eje X debería ser sepal length
- Debe definir una métrica de desempeño y justificar el por qué de esta métrica. Con dicha métrica, evalúe el rendimiento de su modelo. Es decir, diga si lo hizo bien o no y el por qué.

Para todos los subtasks de la parte 2 **evite el uso de herramientas de AI generativas (ChatGPT)**.

Nota: Para las partes donde se solicita que usen librerías, consideren usar [Sckit-learn](#). Por favor, **asegúrese de leer la documentación** de estas librerías para entender mejor los hiperparametros. Siempre recuerde, si tiene alguna duda o consulta, por favor comuníquese con el catedrático.

Entregas en Canvas

1. Jupyter Notebook respondiendo a cada task.
 - a. Incluyendo las preguntas del task 1
 - b. Dentro del Jupyter Notebook deben colocar el link a su repositorio en GitHub. Este puede permanecer como privado hasta la fecha de entrega.

Evaluación

1. [1 pt] Task 1
 - a. [0.2 pts] Pregunta 1
 - b. [0.4 pts] Pregunta 2 y 3
2. [4 pts.] Task 2
 - a. [1 pts] Subtask 2.1 y 2.3
 - b. [2 pts] Subtask 2.2