### **SEMANA 2 (RÚBRICA)**

| **Rúbrica de Evaluación** | |  |
| --- | --- | --- |
| **REA 1.** | |
| **Actividad 1: Implementación de Redes Neuronales Básicas en Google Colab** | |
| **Máxima calificación 5.0** | |
| **Primer criterio de Evaluación.**  En este criterio, los estudiantes deben implementar un **perceptrón** desde cero, sin utilizar librerías especializadas en deep learning como TensorFlow o PyTorch. El perceptrón debe estar programado utilizando operaciones matriciales y lógicas con **NumPy**. Se espera que los estudiantes demuestren cómo este modelo puede realizar una clasificación binaria, basándose en un conjunto de datos sencillo.  **Este criterio representa 1 unidades del total de 5 que tiene la actividad.** | **(4 - 5) : Excelente**: El perceptrón está completamente implementado y resuelve correctamente problemas de clasificación linealmente separables. El código es eficiente, claro y sigue buenas prácticas de programación.  **(3 - 4) : Bueno**: El perceptrón está implementado correctamente y resuelve la mayoría de los problemas de clasificación lineal. El código es funcional, aunque puede mejorar en claridad o eficiencia.  **(2 - 3) : Suficiente**: El perceptrón está parcialmente implementado, con algunos errores que afectan su funcionamiento. Resuelve algunos casos de clasificación, pero con limitaciones.  **(1 - 2) : Deficiente**: El perceptrón está incompleto o con fallos graves que impiden que funcione en la mayoría de los casos.  **(0 - 1) : Insuficiente**: El perceptrón no está implementado o la implementación es incorrecta y no clasifica correctamente ningún conjunto de datos. | |
| **Segundo criterio de Evaluación.**  Este criterio evalúa la implementación de una **red neuronal de una sola capa**. El modelo debe extender el concepto de perceptrón a múltiples neuronas en una capa de salida, utilizando operaciones matriciales. El objetivo es que los estudiantes implementen una red que pueda realizar tareas de clasificación binaria o multiclase simple.  **Este criterio representa 1 unidades del total de 5 que tiene la actividad.** | **(4 - 5) : Excelente**: La red neuronal de una capa está correctamente implementada. Los resultados son correctos y el código sigue buenas prácticas.  **(3 - 4) : Bueno**: La red neuronal de una capa está mayormente implementada, con pequeños detalles que pueden mejorarse en términos de eficiencia o claridad.  **(2 - 3) : Suficiente**: La red neuronal de una capa tiene algunos errores, pero la lógica básica está presente.  **(1 - 2) : Deficiente**: La red neuronal de una capa está incompleta o con errores graves que afectan su funcionamiento.  **(0 - 1) : Insuficiente**: La red neuronal de una capa no está implementada correctamente o no produce resultados válidos. | |
| **Tercer criterio de Evaluación.**  Este criterio evalúa la implementación de una **red neuronal multicapa**, que incluye al menos una capa oculta. Los estudiantes deben demostrar cómo las capas ocultas permiten a la red aprender características más complejas. Se espera que la red ajuste manualmente o básicamente los pesos y realizar la clasificación.  **Este criterio representa 3 unidades del total de 5 que tiene la actividad.** | **(4 - 5) : Excelente**: La red neuronal multicapa está completamente implementada con múltiples capas y funciones de activación, que funcionan correctamente.  **(3 - 4) : Bueno**: La red neuronal multicapa está implementada, pero podría mejorar en claridad o eficiencia del código. Los resultados son en su mayoría correctos.  **(2 - 3) : Suficiente**: La red neuronal multicapa tiene algunos errores en la implementación, pero funciona parcialmente.  **(1 - 2) : Deficiente**: La red neuronal multicapa está incompleta o con errores graves.  **(0 - 1) : Insuficiente**: La red neuronal multicapa no está implementada correctamente o no produce resultados válidos. | |
| **Cuarto criterio de Evaluación.**  Este criterio evalúa el uso adecuado de **NumPy** para realizar las operaciones matriciales necesarias en la implementación de las redes neuronales. Se espera que los estudiantes utilicen operaciones vectorizadas para mejorar la eficiencia de los cálculos y evitar el uso de bucles innecesarios.  **Este criterio representa 1 unidades del total de 5 que tiene la actividad.** | **(4 - 5) : Excelente**: Todas las operaciones matriciales (multiplicaciones, sumas, transposiciones, etc.) están correctamente implementadas usando **NumPy**. El código es eficiente y bien optimizado.  **(3 - 4) : Bueno**: Las operaciones con **NumPy** están bien implementadas, pero podrían mejorarse algunas áreas para mayor eficiencia.  **(2 - 3) : Suficiente**: Las operaciones con **NumPy** están parcialmente implementadas, pero tienen errores o ineficiencias en algunos puntos.  **(1 - 2) : Deficiente**: El uso de **NumPy** es limitado o tiene errores significativos en la implementación.  **(0 - 1) : Insuficiente**: El uso de **NumPy** es incorrecto o mínimo, y las operaciones no se realizan de manera correcta. | |
| **Quinto criterio de Evaluación.**  Este criterio evalúa la **documentación** y **organización del código**. Los estudiantes deben presentar un código bien estructurado, fácil de seguir, con comentarios que expliquen cada parte del proceso, desde la carga de los datos hasta la implementación de las redes neuronales y las funciones de optimización.  **Este criterio representa 1 unidades del total de 5 que tiene la actividad.** | **(4 - 5) : Excelente**: El código está bien documentado, con comentarios claros que explican cada paso. Está organizado, con funciones y variables bien definidas y estructuradas.  **(3 - 4) : Bueno**: El código está mayormente documentado y organizado, pero podría mejorarse en algunos comentarios o en la estructura del código.  **(2 - 3) : Suficiente**: La documentación y organización del código son mínimas, pero aún así es posible entender el código con algo de esfuerzo.  **(1 - 2) : Deficiente**: La documentación es escasa o inexistente y el código está desorganizado, dificultando la comprensión.  **(0 - 1) : Insuficiente**: No hay documentación ni organización en el código, lo que lo hace difícil de entender. | |
| **Sexto criterio de Evaluación.**  Este criterio evalúa el **funcionamiento global** del código y la **calidad de los resultados obtenidos**. El enfoque está en qué tan bien el modelo implementado funciona en términos de clasificación y si los resultados obtenidos son consistentes y de alta calidad.  **Este criterio representa 1 unidades del total de 5 que tiene la actividad.** | **(4 - 5) : Excelente**: Todos los modelos (perceptrón, red de una capa y multicapa) funcionan correctamente y los resultados son consistentes. La presentación de los resultados es clara y adecuada.  **(3 - 4) : Bueno**: Los modelos funcionan correctamente en su mayoría, aunque algunos resultados podrían estar mejor presentados o justificarse con más detalle.  **(2 - 3) : Suficiente**: Los resultados son parcialmente correctos, pero hay errores o inconsistencias en algunos casos.  **(1 - 2) : Deficiente**: Los resultados son mayormente incorrectos y hay errores importantes en el funcionamiento de los modelos.  **(0 - 1) : Insuficiente**: Los modelos no funcionan correctamente y no producen resultados válidos. | |