Aprendizaje Automático Profundo – Examen modelo **TEMA 1**

Dado que la resolución de este examen NO requiere ejecutar código Python, se autoriza el uso de la computadora únicamente para:

- **Consultar el material del curso disponible en Moodle.**
- **Utilizar la calculadora o Google Colab como herramienta para realizar los cálculos solicitados.**

1) **(1.5 ptos)** En el archivo **autos.csv**, el atributo **engine-size** contiene valores numéricos correspondientes al tamaño del motor de cada vehículo con las siguientes métricas:

	mean	**std**	**min**	**max**
engine-size	126,907	41,643	61	326

a) Dados cuatro valores nuevos del atributo **engine-size** complete la siguiente tabla con la información solicitada. Incluya todos los cálculos realizados

engine-size	**Valor normalizado linealmente entre 0 y 1**	**Valor normalizado usando media y desvío**
300	0,901887	4,156593
250	0,713208	2,955911
30	-0,116981	-2,327090
200	0,524528	1,755229

b) ¿Qué representan los valores normalizados linealmente?
Los valores normalizados linealmente son los datos originales ajustados para estar en un rango fijo, como entre 0 y 1. Esto significa que el valor más bajo se vuelve 0 y el más alto se vuelve 1, y los demás se ajustan en ese rango. Esto ayuda a que todos los datos estén en una escala similar, lo cual es útil para que los modelos los puedan comparar más fácilmente sin que unos valores sean mucho más grandes que otros.

c) ¿Qué representan los valores normalizados con media y desvío?
Cuando normalizamos con media y desvío, lo que hacemos es ajustar los datos para que estén centrados alrededor de 0 y medir qué tan lejos están de la media en términos de "pasos" llamados desviaciones estándar. Así, los datos que están más cerca de 0 están cerca de la media, mientras que los valores más alejados son los que están más lejos de la media. Esto permite ver qué datos son "normales" y cuáles se salen mucho del promedio.

d) ¿Cuál de las dos normalizaciones se relaciona más con la detección de valores atípicos y por qué?
La normalización con **media y desvío** ayuda más a detectar valores atípicos, porque se enfoca en lo lejos que está cada valor del promedio. Entonces, si un dato está muy lejos de la media, será fácil de notar, ya que tendrá un valor alto (positivo o negativo) en comparación con los que están cerca de la media.

2) **(1 pto)** Se dispone de información de deportistas seleccionados para cierta actividad.

Luego de numerizar el atributo HABILIDAD de la siguiente forma: BAJA->1, MEDIA->2 y ALTA->3 y sin normalizar los atributos, se entrenó un perceptrón para predecir el valor del atributo SELECCIONADO.

El modelo obtenido fue el siguiente:

EDAD	ALTURA	HABILIDAD	SELECCIONADO
16	202	Alta	Si
17	157	Baja	No
18	159	Media	No
16	148	Baja	No
11	187	Media	No
19	123	Baja	No
17	204	Alta	Si
19	154	Baja	No

W(EDAD)	W(ALTURA)	W(HABILIDAD)	Sesgo o bias
0.392	**0.001**	**-2.441**	**0.4022**

¿Cómo clasifica el perceptrón a un ejemplo donde **EDAD=10**, **Altura=150** y **Habilidad=Media**? Incluya todos los cálculos realizados.

$$y = (W(\text{EDAD}) \times \text{Edad}) + (W(\text{ALTURA}) \times \text{Altura}) + (W(\text{HABILIDAD}) \times \text{Habilidad}) + \text{Sesgo}$$

Y=(0,392*10)+(0,001*150)+(-2.441*2)+0,4022=-0,4098

Si uso como función de activación

- Si y >=0,el resultado es “si lo selecciono”
- Si y<=0, el resultado es “no lo selecciono”

La respuesta seria “no lo selecciono”

3) **(1.5 pto)** ¿Qué similitudes y diferencias tienen un perceptrón, un combinador lineal y una neurona no lineal?

Característica	Perceptrón	Combinador Lineal	Neurona No Lineal
Aprendizaje Supervisado	Sí	Sí	Sí
Usa Pesos y Sesgos	Sí	Sí	Sí
Suma Ponderada de Entradas	Sí, suma las entradas con pesos	Sí, solo hace una combinación lineal	Sí, empieza con una combinación lineal
Función de Activación	Sí, generalmente escalón o lineal	No tiene función de activación	Sí, usa funciones no lineales (ReLU, sigmoide)
Cantidad de Neuronas	Puede tener varias neuronas	No se organiza en neuronas	Puede tener varias neuronas, igual que el perceptrón
Modela No Linealidades	Limitado en versión básica	No puede modelar no linealidades	Sí, gracias a su función de activación
Aplicaciones Comunes	Clasificación binaria y problemas complejos	Usado en regresión y cálculos lineales	Redes profundas y modelos avanzados

4) **(1.5 ptos)** El archivo **vinos.csv** contiene 178 muestras de vinos de 3 clases distribuidos de la siguiente forma: 59 de Tipo1, 68 de Tipo 2 y 51 de Tipo 3. Se construyó un clasificador capaz de predecir con una precisión del 100% las muestras de Tipo 1 y del 50% las del Tipo 2. Además, el recall del clasificador para las muestras de Tipo 2 fue del 50%.

   a) Indique cuál es la matriz de confusión correspondiente a este clasificador.

   b) Indique los valores de precisión y recall faltantes junto con la precisión (accuracy) del clasificador.

Recordar

- **Matriz de confusión:** herramienta que permite evaluar el desempeño de un modelo de clasificación.
- **Precisión:** indica la porción de predicciones correctas para cada clase.
- **Recall:** indica la porción de verdaderos positivos sobre el total de muestras de la clase.

Datos:

Cantidad de muestras

- Tipo 1=68
- Tipo 2=51
- Tipo 3=51

Valores de las metricas

- Tipo 1 =precisión 100%
- Tipo 2= precisión 50%
- Tipo 2= recall 50%

a)

Matriz de confusión

	Predicho tipo 1	Predicho tipo 2	Predicho tipo 3
Real tipo 1	59	0	0
Real tipo 2	0	34	34
Real tipo 3	0	0	0

Como existen 3 clases(tipo 1,tipo 2 y tipo 3) la matriz de confusión va a ser de 3x3

b)

Métricas faltantes

	Tipo 1	Tipo 2	Tipo 3
Recall	0%	50%	0%
Precision	100%	50%	0%

Calculo del acurracy total

$$\text{Accuracy} = \frac{\text{TP Tipo 1} + \text{TP Tipo 2} + \text{TP Tipo 3}}{\text{Total de muestras}} = \frac{59 + 34 + 0}{178} = \frac{93}{178} \approx 52.25\%$$

5) **(1 pto)** Se entrenó una red capaz de identificar correctamente a cuál de los 3 tipos de vino pertenece una muestra del archivo vinos.csv formada sólo por dos capas: la capa de entrada y la capa de salida. Dicha capa de salida está formada por 3 neuronas.
   Se ingresa una muestra a la red y se calcula la entrada neta de cada neurona de la capa de salida. Los valores obtenidos para la 1ra., 2da y 3ra. neurona fueron: **-1.5**, **4** y **6** respectivamente.
   a) Indique cuál sería la respuesta de la red si la función de activación de la capa de salida es la función sigmoide entre 0 y 1.
   b) Indique cuál sería la respuesta de la red si la función de activación de la capa de salida fuera la función softmax.

Datos

Capa de salida:

- Neurona 1=-1,5
- Neurona 2=4
- Neurona 3=6

Función de la sigmoide

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Cálculos:

Sigmoide (-1,5) =0,1824 → Neurona 1
Sigmoide (4) =0,9820 →Neurona 2
Sigmoide (6) =0,9975→Neurona 3

Función de la softmax:

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

Calculos

Softmax (-1,5) =0,0005→ Neurona 1

Softmax (4) =0,1191→Neurona 2

Softmax (6) =0,8804→Neurona 3

6) **(1 pto)** Función de costo de una neurona. Entropía cruzada binaria:

a) **¿Qué características debe tener una función para considerarse una función de costo?**

Una función de costo mide el error entre las predicciones de un modelo y los valores reales deseados. Las características principales de una función de costo son:

1. **Continuidad**: Debe ser continua en el dominio de los parámetros del modelo, permitiendo así un descenso suave en el error durante el entrenamiento.
2. **No-negatividad**: Los valores de la función deben ser no negativos y generalmente toman un mínimo de cero cuando el modelo hace predicciones perfectas.
3. **Minimización del error**: Al optimizar (minimizar) la función de costo, se espera que el modelo mejore su precisión en las predicciones.
4. **Diferenciabilidad**: Idealmente, la función de costo debe ser diferenciable para calcular gradientes, permitiendo el uso de técnicas como el descenso de gradiente.

b) **¿Qué ventaja tiene utilizar la función Entropía Cruzada Binaria en lugar del Error Cuadrático Medio para entrenar una neurona no lineal con función sigmoide entre 0 y 1? Explique.**

La Entropía Cruzada Binaria tiene ventajas sobre el Error Cuadrático Medio (MSE) cuando se trata de funciones de activación como la sigmoide entre 0 y 1, principalmente porque:

1. **Mejor ajuste para clasificación**: La entropía cruzada está diseñada para problemas de clasificación, especialmente cuando se busca maximizar la probabilidad de clasificación correcta. Por otro lado, el MSE es más adecuado para problemas de regresión.
2. **Gradientes más efectivos**: La entropía cruzada genera gradientes más grandes que el MSE cuando las predicciones son incorrectas. Esto acelera el aprendizaje, ya que se penalizan más las predicciones incorrectas y el modelo puede ajustarse mejor.
3. **Evita el problema de saturación de la sigmoide**: Al usar MSE con una función sigmoide, es más probable que el modelo se quede en áreas de gradiente muy pequeño (problema de saturación). La entropía cruzada, al penalizar más fuerte las predicciones erróneas, mitiga este problema y facilita el aprendizaje.

c) **¿Podría utilizarse la función Entropía Cruzada Binaria como función de costo para un perceptrón? Explique**

Sí, la entropía cruzada binaria puede usarse como función de costo para un perceptrón si estamos resolviendo un problema de clasificación binaria. El perceptrón, aunque suele emplearse con una función de activación escalonada (output de 0 o 1), también puede implementarse con una función sigmoide, lo que lo convierte en un modelo adecuado para la entropía cruzada binaria.

7) **(1 pto)** Se dispone de 3000 ejemplos para entrenar un multiperceptrón y se debe decidir entre utilizar o no utilizar lotes de tamaño 150.

a) Explique qué diferencia hay entre estas dos opciones.

Opción 1: Sin lotes (entrenamiento en batch completo)
Cuando el modelo se entrena **sin dividir en lotes**, se utiliza el conjunto completo de datos (3000 ejemplos)

en cada iteración. Esto significa que el modelo realiza una sola pasada completa por todos los ejemplos, calcula el gradiente de error total en función de todos los datos, y luego ajusta los pesos de acuerdo a este gradiente.

- **Ventajas**: Al utilizar toda la información en una sola actualización, el modelo suele tener una **convergencia más estable**.
- **Desventajas**: Este enfoque **consume más memoria** y **puede ser más lento**, especialmente en sistemas con recursos limitados, ya que necesita cargar todos los datos de entrenamiento en cada iteración.

Opción 2: Con lotes (mini-batch)
En este caso, el entrenamiento se divide en **mini-lotes de 150 ejemplos** cada uno. Para cada lote, el modelo calcula el gradiente de error y ajusta los pesos antes de pasar al siguiente lote.

- **Ventajas**:
  - Es **más eficiente en términos de memoria**, ya que solo requiere procesar un pequeño lote de datos cada vez.
  - **Acelera la convergencia** al realizar varias actualizaciones de pesos en una sola época, lo que permite avanzar más rápidamente en el aprendizaje del modelo.
  - Ofrece un balance entre la estabilidad del batch completo y la variabilidad del descenso de gradiente estocástico.
- **Desventajas**: Los gradientes calculados en cada mini-lote pueden ser **menos representativos del conjunto total de datos**, lo que puede hacer que el proceso sea un poco más ruidoso y menos estable en cada actualización, aunque sigue siendo efectivo en términos generales.

b) Si se utilizan lotes de tamaño 150 y se sabe que el conjunto completo de ejemplos fue ingresado al multiperceptrón 200 veces hasta alcanzar la cota de error esperada,

i. ¿cuántas iteraciones y cuántas épocas se realizaron?

- **cantidad de lotes x época**= cantidad de ejemplos/tamaño del lote=3000/150=20 lotes por epoca
- **iteracione**s=épocas x lotes por época=200x20=4000 iteraciones

ii. ¿cuántas veces se actualizaron los pesos de la red?

Como en cada iteración se actualizan los pesos, se actualizaron 4000 veces

iii. ¿cuántas veces se habrían actualizado los pesos de la red si no se hubieran utilizado lotes?

200 veces ya que se hubieran actualizado al final de cada epoca

8) **(1.5 pto)** Si ingresa una imagen representada en RGB (3 canales) de 32x32 pixels a una capa Conv2D formada por 8 filtros de 4x4 con bias, sin padding y con un stride de 2

a) ¿Qué tamaño tendrá la salida de esta capa Conv2d? Detalle los cálculos realizados

Salida=(n+2*p-k)/s +1=(32+2*0-4)/2 +1=15

- n =32: tamaño de la entrada (32x32),
- p=0p = 0p=0: padding (sin padding),
- k=4k = 4k=4: tamaño del filtro (4x4),
- s=2s = 2s=2: stride.

b) ¿Cuántos parámetros tiene la capa Conv2D? Incluya los cálculos realizados.

Cant de parámetros= cantde filtros*tamaño del filtros*cant de canales + bias=8*4*4*3+8=392