

Profesor: Dr. Oldemar Rodríguez Rojas

Análisis de Datos 2

Fecha de Entrega: Domingo 13 de noviembre a las 12 media noche

Instrucciones:

- Las tareas son estrictamente individuales.
- Tareas idénticas se les asignará cero puntos.
- Todas las tareas tienen el mismo valor en la nota final del curso.
- Cada día de entrega tardía tendrá un rebajo de 20 puntos.

TAREA NÚMERO 11

1. [10 puntos] Cree en Python usando `numpy` un tensor de $0D$, $1D$, $2D$ y uno $3D$.
2. [10 puntos] Imprima las dimensiones (`shape`) de la tabla de entrenamiento de MNIST. Esta es una tabla de dígitos escritos a mano y se usan para problemas de clasificación (problemas predictivos). Esta tabla viene con el paquete *tensorflow* y se obtiene como sigue:

```
from tensorflow.keras.datasets import mnist

(x_train, y_train), (x_test, y_test) = mnist.load_data()
```

3. [10 puntos] ¿Qué tipo de datos contiene MNIST?
4. [10 puntos] La operación `relu()` es una operación que se aplica entrada por entrada de un vector, esta devuelve el máximo entre cada entrada del vector y 0 ($\text{relu}(\mathbf{x}) = \max(\mathbf{x}, 0)$). Reprograme esta función en Python y después pruébela con el siguiente vector \mathbf{x} .

```
x = np.array[1, -9, -0.9, 45]
```

5. [10 puntos] Calcule “a mano” el gradiente de la siguiente función:

$$f(x, y, z) = 3x^4 - 6x \sin(yz) + 3y \cos(z).$$

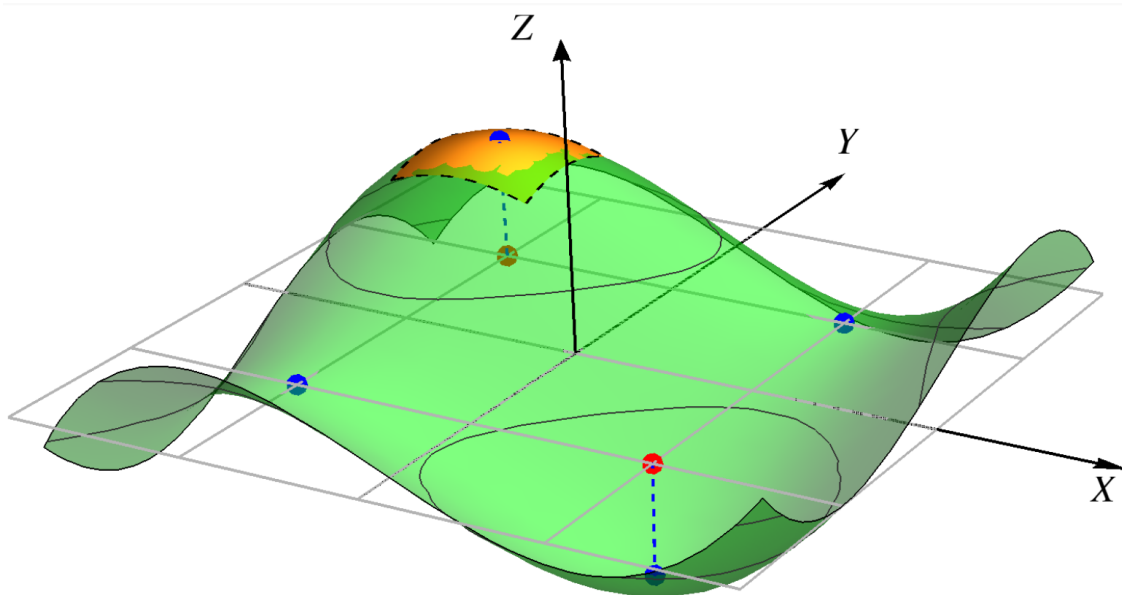
6. [10 puntos] Usando el algoritmo del Descenso de Gradiente encuentre “a mano” (puede utilizar excel) un mínimo local de la siguiente función en el intervalo $[-1, 4]$, use el punto de partida que considere adecuados. Luego grafique y verifique los resultados con código Python.

$$f(x) = 3x^4 - 16x^3 + 18x^2.$$

7. [15 puntos] Usando el algoritmo del Descenso de Gradiente encuentre “a mano” (puede utilizar excel) el mínimo global de la siguiente función, use un punto de partida que considere adecuado. Luego grafique y verifique los resultados con código Python.

$$f(x, y) = x^4 + y^4 - 2x^2 + 4xy - 2y^2.$$

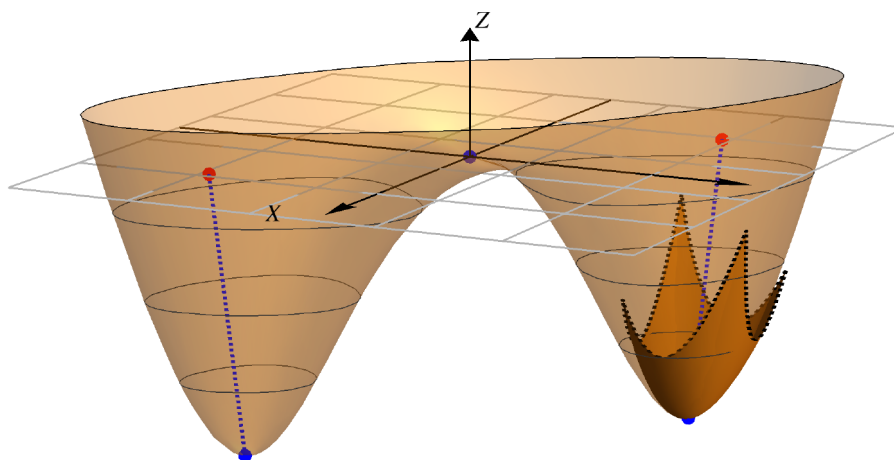
Gráficamente:



8. [15 puntos] Usando el algoritmo del Descenso de Gradiente encuentre “a mano” (puede utilizar excel) un mínimo local de la siguiente función, use el punto de partida que considere adecuados. Luego grafique y verifique los resultados con código **Python**.

$$f(x, y) = x^3 + 3y - y^3 - 3x$$

Gráficamente:



9. [10 puntos] ¿Qué pasa si en el código visto en clase para optimizar una Función de Costo (`gradient_descent`) se usa un momentum (impulso) más pequeño a 0.7? ¿Qué pasa si en el código visto en clase para optimizar una Función de Costo (`gradient_descent`) se usa un η (`learning_rate`) de 10^{-9} ?

NOTA: Para generar gráficos en 3D puede investigar sobre el paquete plotly y utilizar de guía el siguiente ejemplo.

```
import plotly.graph_objects as go
import pandas as pd

def f(x):
    return(x**2 - y**2)

x,y = np.meshgrid(np.arange(-2, 2, 0.1), np.arange(-2, 2, 0.1))
z = f(x, y)

fig = go.Figure()
fig.add_trace(go.Surface(x = x, y = y, z = z, opacity = .7, showscale = False,
                        colorscale = 'Viridis'))
fig.show()
```

