Instituto Politécnico Nacional

Escuela Superior de Cómputo

Bioinformatics

Lab Session 3. Force Field

González Bocio Erik Alexander 2020630163

Jorge Luis Rosas Trigueros

Fecha de realización: 28 de febrero de 2022

Fecha de entrega: 7 de marzo de 2022

Marco Teórico:

Google Colab:

Colab, también conocido como "Colaboratory", te permite programar y ejecutar Python en tu navegador con las siguientes ventajas:

- No requiere configuración
- Da acceso gratuito a GPUs
- Permite compartir contenido fácilmente

Colab puede facilitar tu trabajo, ya seas estudiante, científico de datos o investigador de IA. Este trabaja en un entorno interactivo denominado cuaderno de Colab que te permite escribir y ejecutar código. También trabaja con celdas de código como la mostrada en la siguiente ilustración

Fig. 1. Celdas de código de ejemplo, se ejecuta un código en python y se compila

Los cuadernos de Colab te permiten combinar código ejecutable y texto enriquecido en un mismo documento, además de imágenes, HTML, LaTeX y mucho más. Los cuadernos que creas en Colab se almacenan en tu cuenta de Google Drive. Puedes compartir tus cuadernos de Colab fácilmente con compañeros de trabajo o amigos, lo que les permite comentarlos o incluso editarlos.

Con Colab, puedes aprovechar toda la potencia de las bibliotecas más populares de Python para analizar y visualizar datos.

De igual manera puedes importar un conjunto de datos de imágenes, entrenar un clasificador de imágenes con dicho conjunto de datos y evaluar el modelo con tan solo usar unas pocas líneas de código. Los cuadernos de Colab ejecutan código en los servidores en la nube de Google, lo que te permite aprovechar la potencia del hardware de Google, incluidas las GPU y TPU, independientemente de la potencia de tu equipo. Lo único que necesitas es un navegador.

Colab es una herramienta muy utilizada en la comunidad de aprendizaje automático. Estos son algunos ejemplos de las aplicaciones que tiene Colab:

- Dar los primeros pasos con TensorFlow
- Desarrollar y entrenar redes neuronales
- Experimentar con TPUs
- Divulgar datos de investigación sobre IA
- Crear tutoriales

Matplotlib

Matplotlib es una biblioteca de visualización de datos multiplataforma basada en matrices NumPy y diseñada para funcionar con la pila SciPy más amplia.

Una de las características más importantes de Matplotlib es su capacidad para funcionar bien con muchos sistemas operativos y backends gráficos. Matplotlib admite docenas de backends y tipos de salida, lo que significa que puede contar con que funcionará independientemente del sistema operativo que esté utilizando o del formato de salida que desee. Este enfoque multiplataforma de todo para todos ha sido una de las grandes fortalezas de Matplotlib.

Importando Matplotlib

Así como usamos la abreviatura np para NumPy y la abreviatura pd para Pandas, usaremos algunas abreviaturas estándar para las importaciones de Matplotlib:

```
import matplotlib as mpl
import matplotlib.pyplot as plt
```

Usaremos la directiva plt.style para elegir estilos estéticos apropiados para nuestras figuras. Aquí estableceremos el estilo clásico, lo que garantiza que los gráficos que creamos utilicen el estilo clásico de Matplotlib:

```
plt.style.use('classic')
```

Material y equipo:

- Computadora
- Google Colab
- Red Wi-Fi

Desarrollo de la práctica:

En esta práctica se llevo a cabo la introducción a Google Colaboratory, programando en Python haciendo uso de la librería Matplotlib, la cual nos ayuda a graficar.
Con ayuda del archivo par_all36m_prot.prm llevamos a cabo las ecuaciones que nos ayudaron a llevar a cabo esta práctica, la cual, dependiendo de nuestro número de boleta nos representaba diferentes casos.
Así, las instrucciones fueron las siguientes:

Find the file par_all36m_prot.prm and plot the following equations for the case according to your student Id.

Bond
CE1 CE1

Angle
CT1 CD OH1

Dihedral
CPT CPT NY CA

Improper
NC2 X X C

Lennard-Jones
CD CE1

Así, primero se llevó a cabo la introducción a Google Colaborative que se nos proporciono con la primera liga mostrada anteriormente, posteriormente se hizo la introducción a la librería de Python que nos ayuda a graficar, esta corresponde a Matplotlib.

Así una vez hecho esto y usando el archivo par_all36m_prot.prm se comenzó con esta práctica. Primero empezamos con bond.
El caso correspondiente al 3 es:

CE1 CE1

En este caso hacemos la ecuación correspondiente, la cuál es:

V(bond) = Kb(b - b0)**2

Ahora, tenemos en el código de Python:

```
import numpy as np
from matplotlib import pyplot as plt

kb=440
b0=1.34
b=np.arange(-30,30.0,0.01)
V=kb*(b-b0)**2
plt.plot(b,V)
plt.show()
```

Fig. 2. Código en Python para Bond

Aquí ingresamos el valor de kb y b0 dados en el documento, los cuales son:

atom type Kb b0
CE1 CE1 440.000 1.3400
Mientras que b es el rango que va variando, en este caso varia de -30 a 30 con paso de 0.01 y tenemos finalmente la siguiente gráfica.

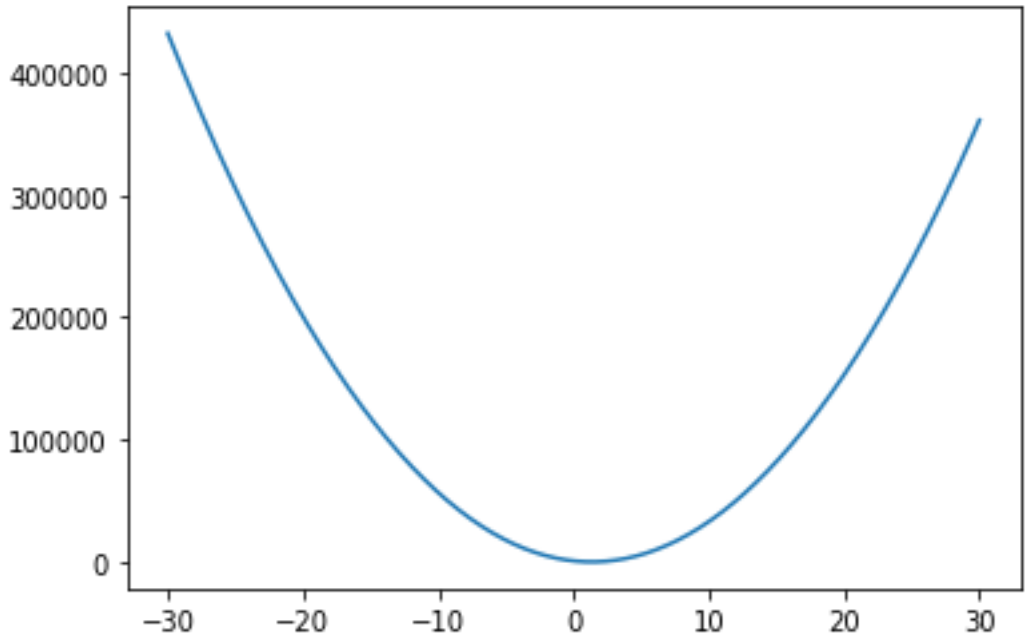

Fig. 3. Gráfica correspondiente a Bond de CE1 CE1

Luego el siguiente caso es el de Angle, en este caso el caso era
CT1 CD OH1

Recordando la formula tenemos que es:

V(angle) = Ktheta(Theta - Theta0)**2

Así, el código en Python queda de la siguiente manera:

```python
import numpy as np
from matplotlib import pyplot as plt

ktheta=55
theta0=110.50
theta=np.arange(-300.0,300.0,50.0)
V=ktheta*(theta-theta0)**2
plt.plot(theta,V)
plt.show()
```

Fig. 4. Código en Python correspondiente a Angle

Así podemos ver representada la formula, además de que se le asigno un valor numérico a los valores de ktheta y theta0, mientras que theta varia de -300 a 300 con paso de 50.

!atom types Ktheta Theta0 Kub S0

CT1 CD OH1 55.000 110.50 ! From ASPP CT2-CD-OH1

De esta forma la grafica queda de la siguiente manera:

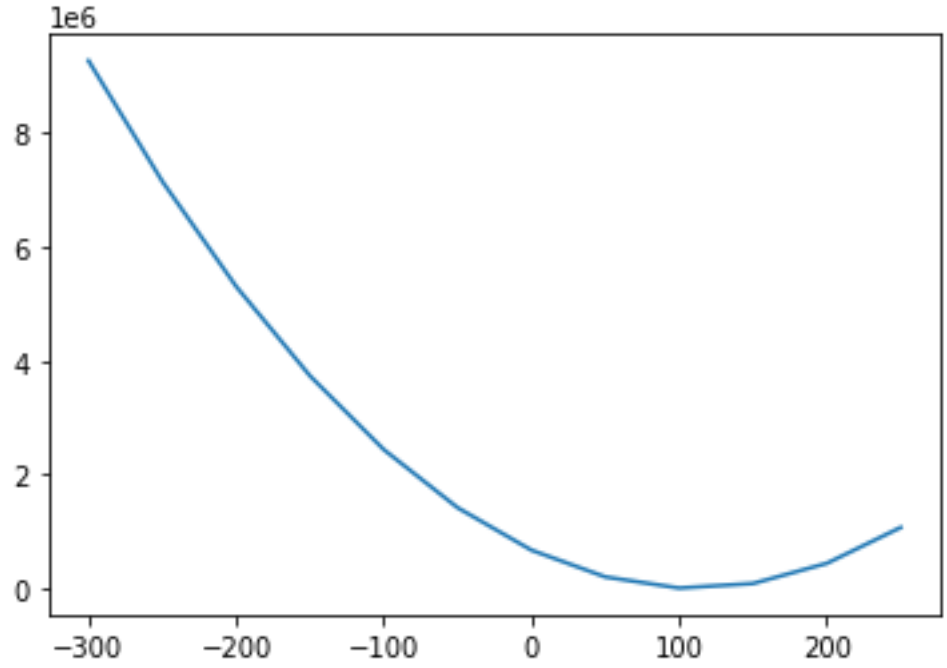

Fig. 5. Gráfica correspondiente a Angle de CT1 CD OH1

El tercer caso para representar es el de

CPT CPT NY CA

Calculando el Dihedral, el cual corresponde a la siguiente formula:
V(dihedral) = Kchi(1 + cos(n(chi) - delta))

Este caso tiene los siguientes valores:

atom types Kchi n delta

CPT CPT NY CA 6.5000 2 180.00 ! atm, methylindole, 1/17/04

Por lo tanto, el código en Python queda de la siguiente manera:

```
import numpy as np
from matplotlib import pyplot as plt

Kchi=6.50
n=2
chi=np.arange(3.5,8.0,0.01)
delta=180
V=Kchi*(1+np.cos(n*(chi)-delta))
plt.plot(chi,V)
plt.show()
```

Fig. 6. Código de Python para el Dihedral

Los valores representados en la tabla son inicializados en el código mientras que chi varia de 3.5 a 8 con un paso de 0.01, quedando la grafica de la siguiente manera:

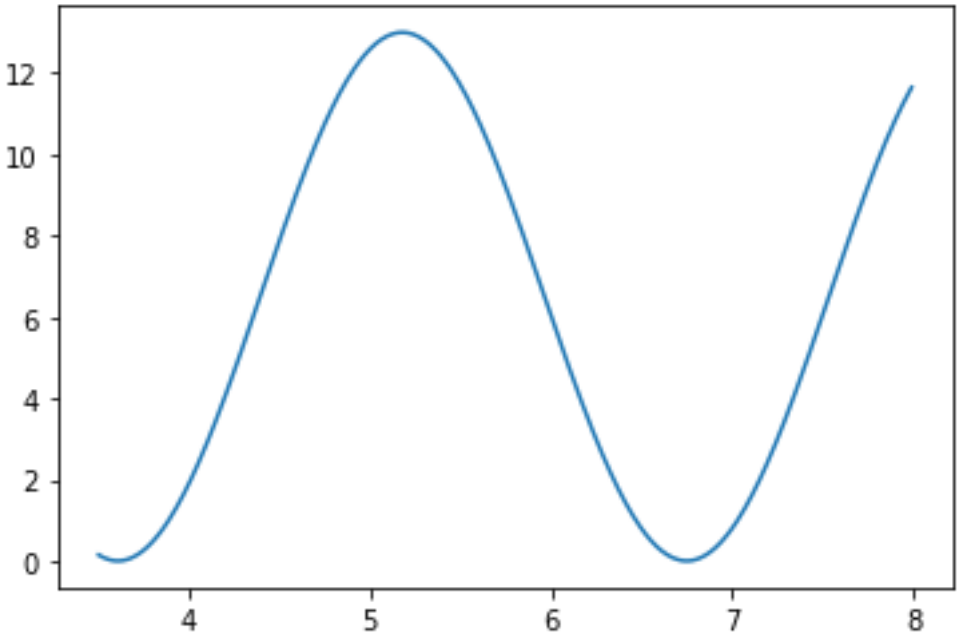

Fig. 7. Gráfica del Dihedral para NC2 X X C

Seguido de este, tenemos el Improper correspondiente a
NC2 X X C

El cual tiene los siguientes valores:

atom types Kpsi psi0
C2 X X C 45.0000 0 0.0000

Para calcular este caso esta la siguiente formula:

V(improper) = Kpsi(psi - psi0)**2

La cual esta representada en el siguiente código de Python

```
import numpy as np
from matplotlib import pyplot as plt

kpsi=45
psi0=0.00
psi=np.arange(-30,30,0.01)
V=kpsi*(psi-psi0)**2
plt.plot(psi,V)
plt.show()
```

Fig. 8. Código de Phyton para Improper

Y nos muestra la siguiente gráfica:

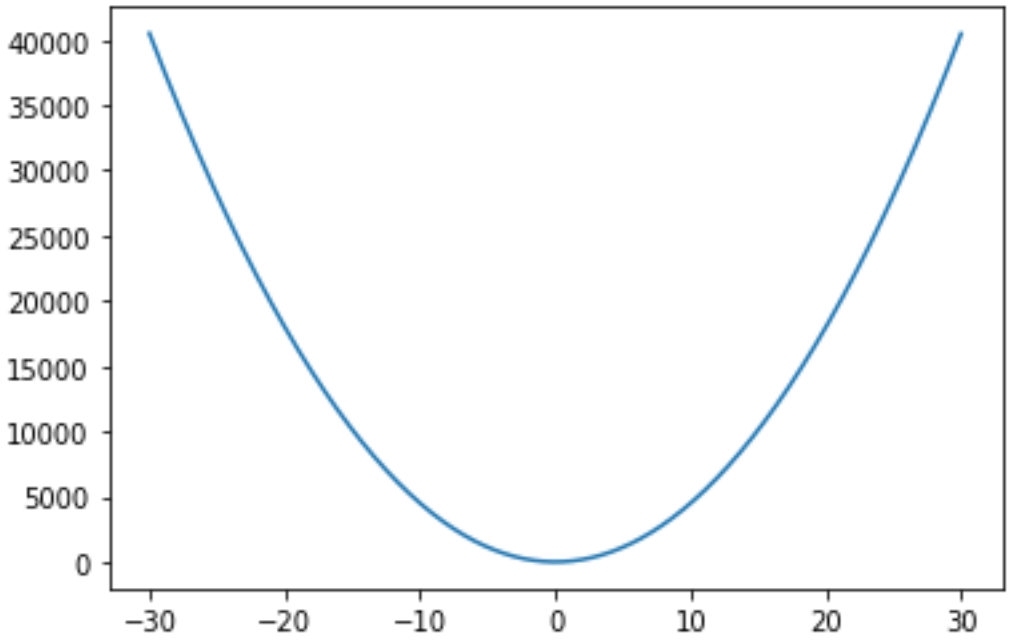

Fig. 9. Gráfica para Improper de C2 X X C

Podemos observar que el valor que varia en este caso es el de psi, ya que los demás se encuentra en la tabla.

Por último calculamos el Lennard-Jones de

CD CE1

Este con la formula:

V(Lennard-Jones) = Eps,i,j[(Rmin,i,j/ri,j)**12 - 2(Rmin,i,j/ri,j)**6]

Y los datos son los siguientes:

atom ignored epsilon Rmin/2 ignored eps,1-4 Rmin/2,1-4

CD 0.000000 -0.070000 2.000000 ! ALLOW POL
! adm jr. 3/19/92, acetate a.i. and dH of solvation

CE1 0.000000 -0.068000 2.090000 !
! for propene, yin/adm jr., 12/95

De esta forma el Código en python nos queda:

```
import numpy as np
import math
from matplotlib import pyplot as plt

epsi=-0.070000
epsj=-0.068000
eps=math.sqrt(epsi*epsj)
rmini=2.000000
rminj=2.090000
Rmin=rmini+rminj
r=np.arange(3.5,8,0.01)
V=eps*((Rmin/r)**12-2*(Rmin/r)**6)
plt.plot(r,V)
plt.show()
```

Fig.Ilustración 10. Gráfica Lennard-Jones para 11. Código Lennard-Jones en Python

En este caso vemos que se cambia un poco el código con respecto a los demás códigos, esto es porque en este caso se usan mas formulas, como lo son:

$$Eps,i,j = sqrt(eps,i * eps,j)$$

$$Rmin,i,j = Rmin/2,i + Rmin/2,j$$

Para esto se usan los valores de epsi y eps j, además de Rmini y Rminj, también tenemos el valor de r, el cual es el que variara en este caso, variando de 3.5 a 8 con paso de 0.01.

De esta forma tenemos la siguiente gráfica:

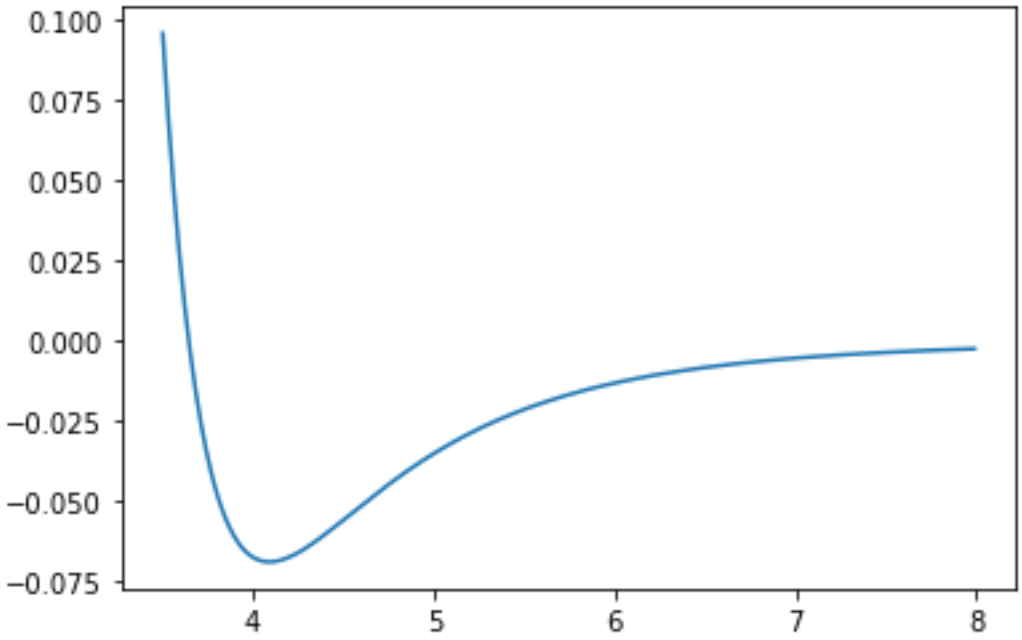

Fig. 12. Código Lennard-Jones para CD CE1

Conclusiones y recomendaciones:

Con esta práctica se dio un breve comienzo en el uso de Google Colab, además de usar Python y una librería la cual no conocía que es Matplotlib, esto añadiendo numpy. Me pareció una práctica entretenida porque jamás había trabajado en Google Colab, al principio no entendía un poco que es lo que se tenia que hacer, pero una vez que el profesor dio un pequeño ejemplo de como es que se utilizaba fue rápido encontrar una manera de llevarlas a cabo.

De igual manera hay algunos conceptos que aun no entiendo del todo pero me parece divertido y muy importante las prácticas que se están llevando a cabo.

Bibliografía:

- Google Colaboratory. Colab.research.google.com. (2022). Retrieved 2 March 2022, from https://colab.research.google.com/?hl=es.
- Google Colaboratory. Colab.research.google.com. (2022). Retrieved 2 March 2022, from https://colab.research.google.com/github/jakevdp/PythonDataScienceHandbook/blob/master/notebooks/04.00-Introduction-To-Matplotlib.ipynb#scrollTo=PS3gupXweUPY.