

Comentarios Tarea 1

☰ Tema	
🔗 Files	
☰ Autor	
🕒 Created	@October 5, 2022 9:07 PM
📅 Reminder	
▼ Status	Open
▼ Tipo	
🔗 URL	
🕒 Updated	@October 6, 2022 2:01 AM

La red neuronal con la que se trabajó tenía 784 datos de entrada, los cuales corresponden a los pixeles de las imagenes. Su capa de salida tiene 10 neuronas y una intermedia de 30 neuronas.

A esta red neuronal se le implementó el algoritmo de Stochastic Gradient Descent.

```

working.py > ...
1  import mnist_loader
2  import network
3  import pickle
4  training_data, validation_data, test_data = mnist_loader.load_data_wrapper()
5  training_data = list(training_data)
6  test_data = list(test_data)
7  net=network.Network([784,30,10])
8  net.SGD( training_data, 30, 10, 3.0, test_data=test_data)
9  archivo = open("red_prueba1.pkl","wb")
10 pickle.dump(net,archivo)
11 archivo.close()
12 exit()
13 #leer el archivo
14 archivo_lectura = open("red_prueba1.pkl","rb")
15 net = pickle.load(archivo_lectura)
16 archivo_lectura.close()
17 net.SGD( training_data, 10, 50, 0.5, test_data=test_data)
18 archivo = open("red_prueba1.pkl","wb")
19 pickle.dump(net,archivo)
20 archivo.close()
21 exit()

```

```

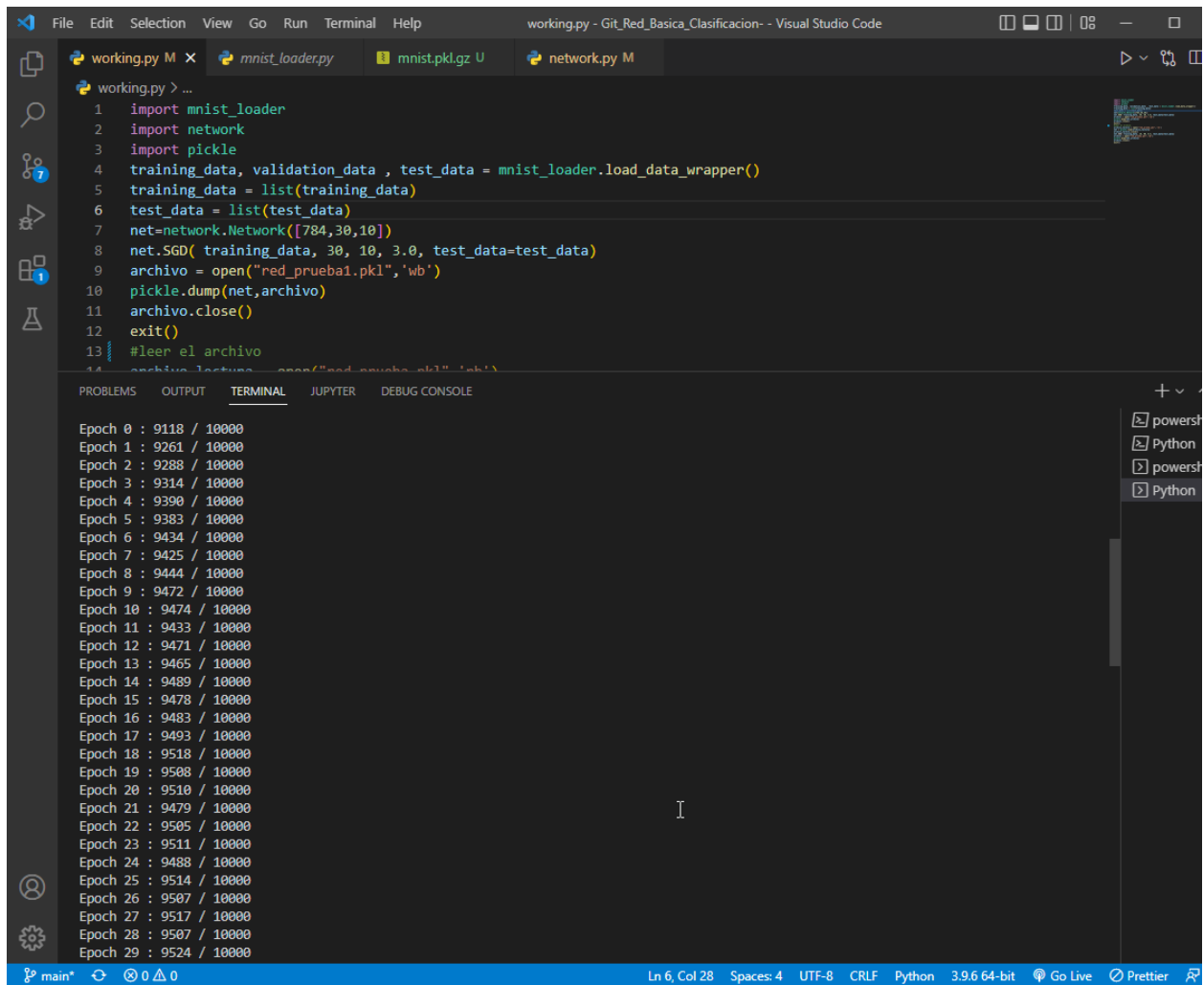
Epoch 5 : 9388 / 10000
Epoch 6 : 9424 / 10000
Epoch 7 : 9445 / 10000
Epoch 8 : 9414 / 10000
Epoch 9 : 9426 / 10000
Epoch 10 : 9436 / 10000
Epoch 11 : 9453 / 10000
Epoch 12 : 9494 / 10000
Epoch 13 : 9461 / 10000
Epoch 14 : 9466 / 10000
Epoch 15 : 9472 / 10000
Epoch 16 : 9484 / 10000
Epoch 17 : 9470 / 10000
Epoch 18 : 9492 / 10000
Epoch 19 : 9500 / 10000
Epoch 20 : 9487 / 10000
Epoch 21 : 9500 / 10000
Epoch 22 : 9512 / 10000
Epoch 23 : 9514 / 10000
Epoch 24 : 9499 / 10000
Epoch 25 : 9500 / 10000
Epoch 26 : 9483 / 10000
Epoch 27 : 9485 / 10000
Epoch 28 : 9508 / 10000
Epoch 29 : 9487 / 10000

```

Se observa que

- para la primera época inició con 90%
- en la época 10 arrojó un valor de 94.22%
- en la época 20 dió un valor de 94.88%
- Para la época 30 dió un valor de 95.04%

Cross Entropy



```
1 import mnist_loader
2 import network
3 import pickle
4 training_data, validation_data, test_data = mnist_loader.load_data_wrapper()
5 training_data = list(training_data)
6 test_data = list(test_data)
7 net=network.Network([784,30,10])
8 net.SGD( training_data, 30, 10, 3.0, test_data=test_data)
9 archivo = open("red_prueba1.pkl","wb")
10 pickle.dump(net,archivo)
11 archivo.close()
12 exit()
13 #leer el archivo
14 archivo = open("red_prueba1.pkl","rb")
```

Epoch 0 : 9118 / 10000
Epoch 1 : 9261 / 10000
Epoch 2 : 9288 / 10000
Epoch 3 : 9314 / 10000
Epoch 4 : 9390 / 10000
Epoch 5 : 9383 / 10000
Epoch 6 : 9434 / 10000
Epoch 7 : 9425 / 10000
Epoch 8 : 9444 / 10000
Epoch 9 : 9472 / 10000
Epoch 10 : 9474 / 10000
Epoch 11 : 9433 / 10000
Epoch 12 : 9471 / 10000
Epoch 13 : 9465 / 10000
Epoch 14 : 9489 / 10000
Epoch 15 : 9478 / 10000
Epoch 16 : 9483 / 10000
Epoch 17 : 9493 / 10000
Epoch 18 : 9518 / 10000
Epoch 19 : 9508 / 10000
Epoch 20 : 9510 / 10000
Epoch 21 : 9479 / 10000
Epoch 22 : 9505 / 10000
Epoch 23 : 9511 / 10000
Epoch 24 : 9488 / 10000
Epoch 25 : 9514 / 10000
Epoch 26 : 9507 / 10000
Epoch 27 : 9517 / 10000
Epoch 28 : 9507 / 10000
Epoch 29 : 9524 / 10000

Se observa que

- para la primera época inició con 90%
- en la época 10 arrojó un valor de 94.22%
- en la época 20 dió un valor de 94.88%
- Para la época 30 dió un valor de 95.24%

Se nota que el rendimiento es menor con respecto al intento sin cross entropy

Stochastic Gradient Descent con momento

```
working.py > ...
1 import mnist_loader
2 import network
3 import pickle
4 training_data, validation_data, test_data = mnist_loader.load_data_wrapper()
5 training_data = list(training_data)
6 test_data = list(test_data)
7 net=network.Network([784,30,10])
8 net.SGD( training_data, 30, 10, 4.0,0.3, test_data=test_data)
9 archivo = open("red_prueba1.pkl", 'wb')
10 pickle.dump(net,archivo)
11 archivo.close()
12 exit()
13 #leer el archivo
14 archivo_lectura = open("red_prueba.pkl", 'rb')
15 net = pickle.load(archivo_lectura)
16 archivo_lectura.close()
17 net.SGD( training_data, 10, 50, 0.5,0.1, test_data=test_data)
18 archivo = open("red_prueba.pkl", 'wb')
19 pickle.dump(net,archivo)
20 archivo.close()
21 exit()
```

Aqui se usó una tasa de aprendizaje de 4 , con fricción de 0.3

```
TypeError: SGD() got multiple values for argument 'test_data'
PS C:\Users\SELENE\Downloads\escuela\RedesNeuronales\Git_Red_Basica_Clasificacion-> & C:/Users/SELENE/AppData/Local/Programs/Python/Python39/python.exe c:/Users/SELENE/Downloads/escuela/RedesNeuronales/Git_Red_Basica_Clasificacion-/working.py
Traceback (most recent call last):
  File "c:\Users\SELENE\Downloads\escuela\RedesNeuronales\Git_Red_Basica_Clasificacion-\working.py", line 8, in <module>
    net.SGD( training_data, 30, 10, 4.0, 0.1, test_data=test_data)
TypeError: SGD() got multiple values for argument 'test_data'
PS C:\Users\SELENE\Downloads\escuela\RedesNeuronales\Git_Red_Basica_Clasificacion-> & C:/Users/SELENE/AppData/Local/Programs/Python/Python39/python.exe c:/Users/SELENE/Downloads/escuela/RedesNeuronales/Git_Red_Basica_Clasificacion-/working.py
Traceback (most recent call last):
  File "c:\Users\SELENE\Downloads\escuela\RedesNeuronales\Git_Red_Basica_Clasificacion-\working.py", line 8, in <module>
    net.SGD( training_data, 30, 10, 4.0, 0.2, test_data=test_data)
TypeError: SGD() got multiple values for argument 'test_data'
PS C:\Users\SELENE\Downloads\escuela\RedesNeuronales\Git_Red_Basica_Clasificacion->
```

Pero no pude resolver los errores

errores

1. De los primeros errores que tuve fue instalar la librería numpy, ya que no la tenía instalada.

```
Traceback (most recent call last):
  File "c:\Users\SELENE\Downloads\escuela\RedesNeuronales\Git_Red_Basica_Cla
on-\network.py", line 10, in <module>
    import numpy as np
ModuleNotFoundError: No module named 'numpy'
PS C:\Users\SELENE\Downloads\escuela\RedesNeuronales\Git_Red_Basica_Clasific
& C:/Users/SELENE/AppData/Local/Programs/Python/Python39/python.exe c:/Users
Downloads/escuela/RedesNeuronales/Git_Red_Basica_Clasificacion-/network.py
Traceback (most recent call last):
  File "c:\Users\SELENE\Downloads\escuela\RedesNeuronales\Git_Red_Basica_Cla
on-\network.py", line 10, in <module>
    import numpy as np
ModuleNotFoundError: No module named 'numpy'
PS C:\Users\SELENE\Downloads\escuela\RedesNeuronales\Git_Red_Basica_Clasific
```

```
working.py > ...
1  import mnist_loader
2  import network
3  import pickle
4  training_data, validation_data , test_data = mnist_loader.load_data_wrapper()
5  training_data = list(training_data)
6  test_data = list(test_data)
7  net=network.Network([784,30,10])
8  net.SGD( training_data, 30, 10, 5.0,0.3, test_data=test_data)
9  archivo = open("red_prueba1.pkl",'wb')
10 pickle.dump(net,archivo)
11 archivo.close()
12 exit()
13 #leer el archivo
14 archivo_lectura = open("red_prueba.pkl",'rb')
15 net = pickle.load(archivo_lectura)
16 archivo_lectura.close()
17 net.SGD( training_data, 10, 50, 0.5,0.1, test_data=test_data)
18 archivo = open("red_prueba.pkl",'wb')
19 pickle.dump(net,archivo)
20 archivo.close()
21 exit()
```

PROBLEMS 17 OUTPUT TERMINAL JUPYTER DEBUG CONSOLE

Epoch 26 : 9501 / 10000
Epoch 27 : 9501 / 10000
Epoch 28 : 9498 / 10000
Epoch 29 : 9477 / 10000
PS C:\Users\SELENE\Downloads\escuela\RedesNeuronales\Git_Red_Basica_Clasificacion->
PS C:\Users\SELENE\Downloads\escuela\RedesNeuronales\Git_Red_Basica_Clasificacion-> & C:/Users/SELENE/AppData/Local/Programs/Python/Python39/python.exe c:/Users/SELENE/Downloads/escuela/RedesNeuronales/Git_Red_Basica_Clasificacion-/working.py
Traceback (most recent call last):
File "c:\Users\SELENE\Downloads\escuela\RedesNeuronales\Git_Red_Basica_Clasificacion-\working.py",
line 2, in <module>
import network
File "c:\Users\SELENE\Downloads\escuela\RedesNeuronales\Git_Red_Basica_Clasificacion-\network.py",
line 73
"""Se usa SGD con momento, de acuerdo a la definici3n"""
^
IndentationError: unindent does not match any outer indentation level

2. Error de escritura