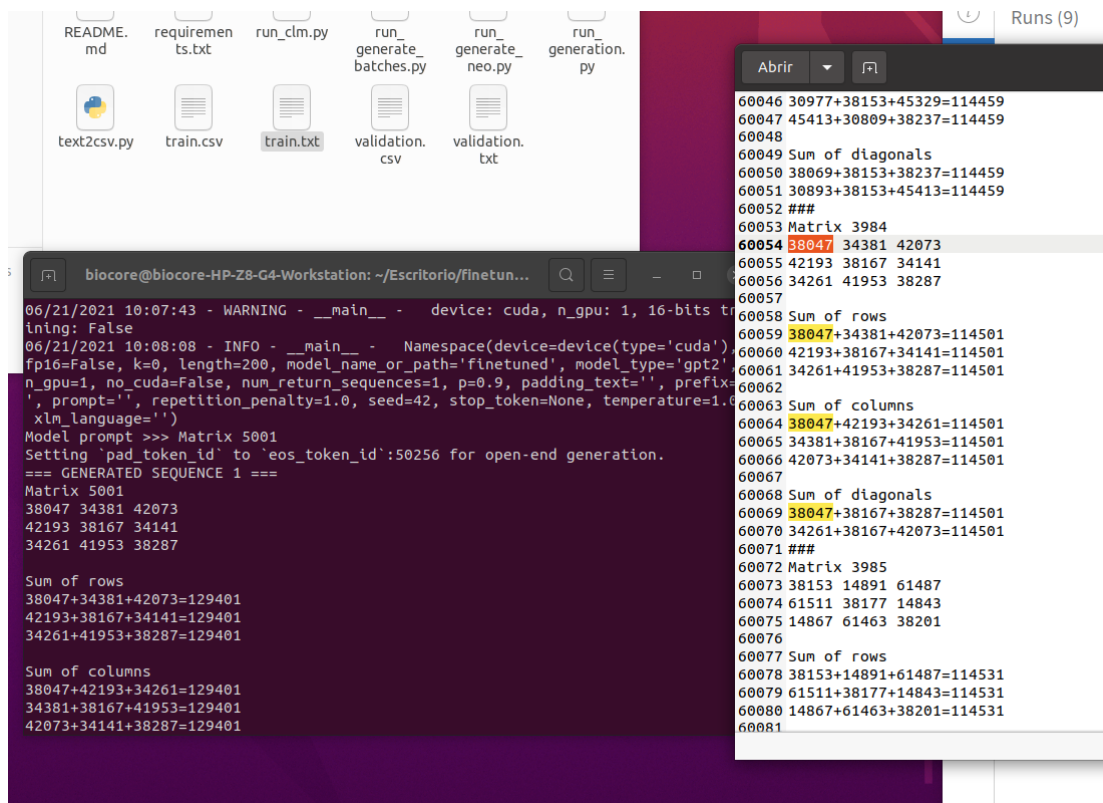


REENTRENAMIENTO O FINE-TUNING DE GPT2XL CON LAS MATRICES DE NÚMEROS PRIMOS.

Se entrenó un modelo gpt-2xl (<https://github.com/Xirider/finetune-gpt2xl>) alimentándolo con matrices anidadas de números primos

Los resultados no fueron los esperados ya que el modelo no ha aprendido a generar nuevas matrices de números primos con la suma de las columnas, filas y diagonales iguales. De hecho, nos dimos cuenta, que el modelo ni si quiera comprende lo que es un número primo ni tampoco sabe sumar correctamente. A continuación, mostramos algunos ejemplos de la inferencia realizada:

MATRICES COPIADAS DE LOS DATOS DE ENTRENAMIENTO



The image shows a file explorer window at the top with files: README.md, requirements.txt, run_clm.py, run_generate_batches.py, run_generate_neo.py, run_generation.py, text2csv.py, train.csv, train.txt, validation.csv, and validation.txt. Below it is a terminal window with the following output:

```
06/21/2021 10:07:43 - WARNING - __main__ - device: cuda, n_gpu: 1, 16-bits tr
ining: False
06/21/2021 10:08:08 - INFO - __main__ - Namespace(device=device(type='cuda'),
fp16=False, k=0, length=200, model_name_or_path='finetuned', model_type='gpt2',
n_gpu=1, no_cuda=False, num_return_sequences=1, p=0.9, padding_text='', prefix=
', prompt='', repetition_penalty=1.0, seed=42, stop_token=None, temperature=1.0
, xlm_language='')
Model prompt >>> Matrix 5001
Setting 'pad_token_id' to 'eos_token_id':50256 for open-end generation.
=== GENERATED SEQUENCE 1 ===
Matrix 5001
38047 34381 42073
42193 38167 34141
34261 41953 38287

Sum of rows
38047+34381+42073=129401
42193+38167+34141=129401
34261+41953+38287=129401

Sum of columns
38047+42193+34261=129401
34381+38167+41953=129401
42073+34141+38287=129401
```

On the right, a 'Runs (9)' window shows a list of runs with their outputs. The output for run 60054 is highlighted in yellow:

```
60046 30977+38153+45329=114459
60047 45413+30809+38237=114459
60048
60049 Sum of diagonals
60050 38069+38153+38237=114459
60051 30893+38153+45413=114459
60052 ###
60053 Matrix 3984
60054 38047 34381 42073
60055 42193 38167 34141
60056 34261 41953 38287
60057
60058 Sum of rows
60059 38047+34381+42073=114501
60060 42193+38167+34141=114501
60061 34261+41953+38287=114501
60062
60063 Sum of columns
60064 38047+42193+34261=114501
60065 34381+38167+41953=114501
60066 42073+34141+38287=114501
60067
60068 Sum of diagonals
60069 38047+38167+38287=114501
60070 34261+38167+42073=114501
60071 ###
60072 Matrix 3985
60073 38153 14891 61487
60074 61511 38177 14843
60075 14867 61463 38201
60076
60077 Sum of rows
60078 38153+14891+61487=114531
60079 61511+38177+14843=114531
60080 14867+61463+38201=114531
60081
```

Como podemos ver en este ejemplo, el modelo genera una matriz 5001 (en la terminal, a la izda. de la imagen), pero que es una copia de una de las matrices introducidas como datos de entrenamiento (a la dcha. de la imagen). Además, ni siquiera copia bien las sumas de las filas y columnas, como se puede observar comparando la matriz generada con la de entrenamiento.

```
ca      text2csv.py  train.csv  train.txt  validation.  validation.
os      csv        txt
le
biocore@biocore-HP-Z8-G4-Workstation: ~/Escritorio/finetun...
', prompt='', repetition_penalty=1.0, seed=42, stop_token=None, temperature=1.0,
  xlm_language='')
Model prompt >>> Generate a prime magic matrix
Setting 'pad_token_id' to 'eos_token_id':50256 for open-end generation.
=== GENERATED SEQUENCE 1 ===
Generate a prime magic matrix
15881+17837+20123=47381
20177+15971+17011=47381
17021+20123+16087=47381

Sum of diagonals
15881+15971+16087=47381
20123+15971+17021=47381
###
Matrix 1817
15889 8887 23143
23227 15973 8719
8803 23059 16057

Sum of rows
15889+8887+23143=47389
23227+15973+8719=47389
8803+23059+16057=47389

27454 Sum of columns
27455 15881+21863+10169=47913
27456 10259+15971+21683=47913
27457 21773+10079+16061=47913
27458
27459 Sum of diagonals
27460 15881+15971+16061=47913
27461 10169+15971+21773=47913
27462 ###
27463 Matrix 1817
27464 15889 8887 23143
27465 23227 15973 8719
27466 8803 23059 16057
27467
27468 Sum of rows
27469 15889+8887+23143=47919
27470 23227+15973+8719=47919
27471 8803+23059+16057=47919
27472
27473 Sum of columns
27474 15889+23227+8803=47919
27475 8887+15973+23059=47919
27476 23143+8719+16057=47919
27477
27478 Sum of diagonals
27479 15889+15973+16057=47919
27480 8803+15973+23143=47919
27481 ###
27482 Matrix 1819
27483 15809 13883 18311
27484 18503 16001 13499
27485 13691 18119 16193
27486
27487 Sum of rows
27488 15809+13883+18311=48003
27489 18503+16001+13499=48003
```

En este ejemplo ocurre lo mismo, la matriz 1817 generada es una copia de la de entrenamiento, y las sumas tampoco las ha copiado bien.

MATRICES 'INVENTADAS' POR EL MODELO

En estos ejemplos, podemos observar que el modelo genera matrices nuevas, no copiadas como en el caso de los ejemplos anteriores. Sin embargo, las matrices generadas no son las esperadas. Algunos de los números que forman la matriz no son primos (aunque la mayoría si lo son, de hecho, de los 4 ejemplos superiores, un 80,55% de los números de las matrices son primos, y el resto no lo son), y la suma de las diferentes filas, columnas y diagonales no dan como resultado un mismo valor, como se puede ver en los cálculos a la derecha de las imágenes.

terminal window showing file explorer and terminal output:

```
biocore@biocore-HP-Z8-G4-Workstation: ~/Escritorio/finetun...
19009+16253+16253=52827
15971+20051+17989=52827
16223+17989+20100=52827
Sum of diagonals
19009+20051+16253=52827
16253+20051+16223=52827
###
Matrix 6001
19973 12979 28433
28463 20063 12919
12949 28403 20083
Sum of rows
19973+12979+28433=52869
28463+20063+12919=52869
12949+28403+20083=52869
Sum of columns
19973+28463+12949=52869
12979+20063+28403=52869
284
(python) biocore@biocore-HP-Z8-G4-Workstation:~/Escritorio/finetune-gpt2x1
$ python run_generation.py --model_type=gpt2 --model_name_or_path=finetu
```

calculator window showing calculations:

19973+12979+28...	=	61385
28463+20063+12...	=	61445
12949+28403+20...	=	61435
19973+28463+12...	=	61385
12979+20063+28...	=	61445
28433+12919+20...	=	61435
19973+20063+20...	=	60119
12949+20063+28...	=	61445
61445		

terminal window showing file explorer and terminal output:

```
biocore@biocore-HP-Z8-G4-Workstation: ~/Escritorio/finetun...
163 7 139 Setting 'pad_token_id' to 'eos_token_id':50256 for open-end generation
=== GENERATED SEQUENCE 1 ===
Matrix 1
15881 16463 17971
18061 15731 16223
16363 17881 15859
Sum of rows
15881+16463+17971=47913
18061+15731+16223=47913
16363+17881+15859=47913
Sum of columns
15881+18061+16363=47913
16463+15731+17881=47913
17971+16223+15859=47913
Sum of diagonals
15881+15731+15859=47913
16363+15731+17971=47913
###
Matrix 2
15727 14591 16619
```

calculator window showing calculations:

19973+20063+20...	=	60119
12949+20063+28...	=	61445
15881+16463+17...	=	50315
18061+15731+16...	=	50015
16363+17881+15...	=	50103
15881+18061+16...	=	50305
16463+15731+17...	=	50075
17971+16223+15...	=	50053
15881+15731+15...	=	47471
16363+15731+17...	=	50065
50065		

File explorer: text2csv.py, train.csv, train.txt, validation.csv, validation.txt

Terminal window: biocore@biocore-HP-Z8-G4-Workstation: ~/Escritorio/finetun...

```
15881+16463+17971=47913
18061+15731+16223=47913
16363+17881+15859=47913

Sum of columns
15881+18061+16363=47913
16463+15731+17881=47913
17971+16223+15859=47913

Sum of diagonals
15881+15731+15859=47913
16363+15731+17971=47913
###
Matrix 2
15727 14591 16619
16691 15739 14447
14519 16547 15791

Sum of rows
15727+14591+16619=47931
16691+15739
(pytorch) biocore@biocore-HP-Z8-G4-Workstation:~/Escritorio/finetune-gpt2xl_matr
ices$ 67 199 43
ls.p:67: orden no encontrada
```

Calculator window: Modo básico

15881+16463+17...	=	50315
18061+15731+16...	=	50015
16363+17881+15...	=	50103
15881+18061+16...	=	50305
16463+15731+17...	=	50075
17971+16223+15...	=	50053
15881+15731+15...	=	47471
16363+15731+17...	=	50065
15727+14591+16...	=	46937
16691+15739+14...	=	46877

46877

Calculator keypad: 7, 8, 9, ÷, ↶, C, 4, 5, 6, ×, (,), 1, 2, 3, −, x², √, 0, ., %, +, =

18149	18169	18181	18191	18199	18211
18251	18253	18257	18269	18287	18289
18329	18341	18353	18367	18371	18379
18433	18439	18443	18451	18457	18461
18521	18523	18539	18541	18553	18583
18661	18671	18679	18691	18701	18713
18757	18773	18787	18793	18797	18803
18911	18913	18917	18919	18947	18959

File explorer: text2csv.py, train.csv, train.txt, validation.csv, validation.txt

Terminal window: biocore@biocore-HP-Z8-G4-Workstation: ~/Escritorio/finetun...

```
Model prompt >>> Prime magic matrices\n\nThe following matrices are built accord
ing to the rules:\n1. All the numbers are prime numbers\n2. The sum of each row,
column and diagonal is always the same for a certain matrix\n
Setting 'pad_token_id' to 'eos_token_id':50256 for open-end generation.
=== GENERATED SEQUENCE 1 ===
Prime magic matrices\n\nThe following matrices are built according to the rules:
\n1. All the numbers are prime numbers\n2. The sum of each row, column and diago
nal is always the same for a certain matrix\n

Sum of diagonals
1+3+4=7
39+3+7=7
###
Matrix 5
4519 743 109
143 3779 727
769 107 2789

Sum of rows
4519+743+109=1431
143+3779+727=1431
769+107+2789=1431
```

Calculator window: Modo básico

16363+17881+15...	=	50103
15881+18061+16...	=	50305
16463+15731+17...	=	50075
17971+16223+15...	=	50053
15881+15731+15...	=	47471
16363+15731+17...	=	50065
15727+14591+16...	=	46937
16691+15739+14...	=	46877
4519+743+109	=	5371
143+3779+727	=	4649

4649

Calculator keypad: 7, 8, 9, ÷, ↶, C, 4, 5, 6, ×, (,), 1, 2, 3, −, x², √, 0, ., %, +, =

3733	3739	3761	3767	3769	3779
3823	3833	3847	3851	3853	3863
3911	3917	3919	3923	3929	3931
4001	4003	4007	4013	4019	4021
4073	4079	4091	4093	4099	4111
4153	4157	4159	4177	4201	4211
4241	4243	4253	4259	4261	4271

Problemas de descomposición de los números

Nos dimos cuenta de que el motivo por el que al modelo le cuesta tanto hacer sumas puede estar en la tokenización que hace de los números: haciendo pruebas con el modelo GPT-J (que usa la misma tokenización que el nuestro) vimos que le costaba descomponer los números en unidades, decenas, etc...

Ejemplo (lo que está en negrita es la entrada y el resto es la salida del modelo):

###

57

ones->7

tens->5

###

50

ones->0

tens->5

###

28

ones->8

tens->2

###

52

ones->2

tens->5

###

36

ones->6

tens->3

###

26

ones->6

tens->2

###

89

ones->9

tens->8

###

98

ones->8

tens->9

###

71

ones->1

tens->7

###

43

ones->4

tens->3

###

33

ones->3

tens->3

###

64

ones->4

tens->6

###

28

ones->8

tens->2

###

87

ones->7

tens->7

###

58

ones->7

tens->7

###

60

ones->8

tens->4

###

60

ones->4

tens->4

###

73

ones->9

tens->6

###

55

ones->7

tens->5

###

55

ones->5

tens->5

###

34

ones->4

tens->4

###

27

ones->4

tens->2

###

61

ones->7

tens->5

###

23

ones->6

tens->3

###

49

ones->4

tens->4

###

65

ones->5

tens->5

###

46

ones->5

Esto nos llevó a la conclusión de que el modelo no tiene del todo bien asociado, por ejemplo, el número 65 con los números 6 y 5. Esto puede explicar que le cueste tanto sumar ya que no es capaz de descomponer bien los números y tendría que aprender a sumar cada número por separado. Es decir, en lugar de hacer:

$$12 + 79$$

1. $2 + 9 = 11$, me llevo una

2. $1 + 7 + 1 = 9$

Resultado: $12 + 79 = 91$

Tendría que saber de alguna manera que el resultado es 91 sin pasos intermedios. Es decir, tendría que memorizar las sumas.

Comprobamos, de hecho, que si repetimos el problema de la descomposición de los números pero introduciendo un espacio entre los números de dos cifras (es decir, en lugar de 65, escribir 6 5, para que no lo tokenize como un número solo sino como un 6 y después un 5) es capaz de descomponer sin problema:

Ejemplo (lo que está en **negrita** es la entrada y el resto es la salida del modelo):

###

5 7
ones->7
tens->5

###

5 0
ones->0
tens->5

###

2 8
ones->8
tens->2

###

5 2
ones->2
tens->5

###

3 6
ones->6
tens->3

###

2 6
ones->6
tens->2

###

8 9
ones->9
tens->8

###

9 8
ones->8
tens->9

###

7 1
ones->1
tens->7

###

4 3
ones->3
tens->4

###

4 7
ones->7
tens->4

###

4 9
ones->9
tens->4

###

6 0
ones->0
tens->6

###

8 0
ones->0
tens->8

###

8 2
ones->2
tens->8

###

4 1
ones->1
tens->4

###

6 5
ones->5
tens->6

###

3 5
ones->5
tens->3

###

2 5
ones->5
tens->2

###

6 2
ones->2
tens->6

###

5 3
ones->3
tens->5

###

7 8
ones->8
tens->7

###

7 9
ones->9
tens->7

###

6 1
ones->1
tens->6

###

5 0
ones->0
tens->5

###

8 4
ones->

ENTRENAMIENTO DE LA SUMA

Entrenamos el modelo durante un fin de semana con datos de entrenamiento de sumas de dos números y una breve explicación de lo que es la operación suma, pero los resultados no fueron buenos, el modelo seguía fallando en casi todas las sumas. Sólo realizaba bien las que ella tenía en los datos de entrenamiento ya que simplemente las copiaba.

Después decidimos probar a reentrenar del modelo con datos de entrenamiento que contenían tutoriales de cómo realizar sumas paso a paso (con miles de ejemplos). En este caso sí que notamos una cierta mejora en los resultados de las sumas que introducíamos, pero seguía fallando en algunas, sobre todo las sumas de un número muy pequeño con uno grande. A continuación, algunos ejemplos (en negrita el texto de input). Ninguna de estas sumas está en los datos de entrenamiento:

Prueba 1:
8292+29039\n\nSTEP 1

Number: 8292
ones->2
tens->9
hundreds->2
thousands->8

Number: 29039
ones->9
tens->3
hundreds->0
thousands->9
ten thousands->2

STEP 2
ones->2+9=11

Number: 11
ones->1
tens->1

STEP 3
tens->9+3+1=13

Number: 13
ones->3
tens->1

STEP 4
hundreds->2+0+1=3

Number: 3
ones->3

STEP 5
thousands-> $8+9=17$

Number: 17
ones->7
tens->1

STEP 6
ten thousands-> $2+1=3$

Number: 3
ones->3

STEP 7
ones->1
tens->3
hundreds->3
thousands->7
ten thousands->3
Number: 37331

RESULT
 $8292+29039=37331$

Prueba 2:
67677+92471\n\nSTEP 1

Number: 67677
ones->7
tens->7
hundreds->6
thousands->7
ten thousands->6

Number: 92471
ones->1
tens->7
hundreds->4
thousands->2
ten thousands->9

STEP 2
ones-> $7+1=8$

Number: 8
ones->8

STEP 3
tens-> $7+7=14$

Number: 14
ones->4
tens->1

STEP 4
hundreds-> $6+4+1=11$

Number: 11
ones->1
tens->1

STEP 5

thousands-> $7+2+1=10$

Number: 10

ones->0

tens->1

STEP 6

ten thousands-> $6+9+1=16$

Number: 16

ones->6

tens->1

STEP 7

hundred thousands-> $1=1$

Number: 1

ones->1

STEP 8

ones->8

tens->4

hundreds->1

thousands->0

ten thousands->6

hundred thousands->1

Number: 160148

RESULT

$67677+92471=160148$

Prueba 3:(suma 80959 en lugar de 9)

9+71421\n\nSTEP 1

Number: 80959

ones->9

tens->5

hundreds->9

thousands->0

ten thousands->8

Number: 71421

ones->1

tens->2

hundreds->4

thousands->1

ten thousands->7

STEP 2

ones-> $9+1=10$

Number: 10

ones->0

tens->1

STEP 3

tens-> $5+2+1=8$

Number: 8

ones->8

STEP 4

hundreds-> $9+4=13$

Number: 13

ones->3
tens->1

STEP 5
thousands-> $0+1+1=2$

Number: 2
ones->2

STEP 6
ten thousands-> $8+7=15$

Number: 15
ones->5
tens->1

STEP 7
hundred thousands-> $1=1$

Number: 1
ones->1

STEP 8
ones->0
tens->8
hundreds->3
thousands->2
ten thousands->5
hundred thousands->1
Number: 152380

RESULT
 $80959+71421=152380$

Hicimos varias pruebas más siempre con el mismo resultado: realiza bien las sumas de números de cifras parecidas, pero falla en sumas de números de cifras muy diferentes.

Pruebas con GPT-J-GB

Este modelo no lo pudimos reentrenar, ya que no disponíamos de medios para ello, pero sí realizamos algunas pruebas con él. Los resultados fueron poco satisfactorios.