AI

FOR FILM CHARACTERIZATION

USING AEL

Bárbara Galindo

Supervisors:Raquel Nunes, Pamela Teubig, Tomás Sousa

What do we want to characterize?

Thickness of the Target

I started with this as it might be the simplest value for the AI to guess.

Homogeneity of the target

An human who saw the graphic of the spectrum would know something about the homogeneity just looking how neat are the peaks. We want the AI to do that too.

Material

Two targets with different materials and thickness can create the same spectrum, so one of them has to be a known variable. The material is to be given as a parameter.

How is the thickness being calculated?

It’s a very precise method, but takes a lot of time

Manual calibration with a reference spectrum

Representation of the spectrum

Searching for the start and end of the peaks

Interpretation of the results

It is necessary to know some values such as density or stopping power

Why using AI

01 We don’t need a real person to do all this process

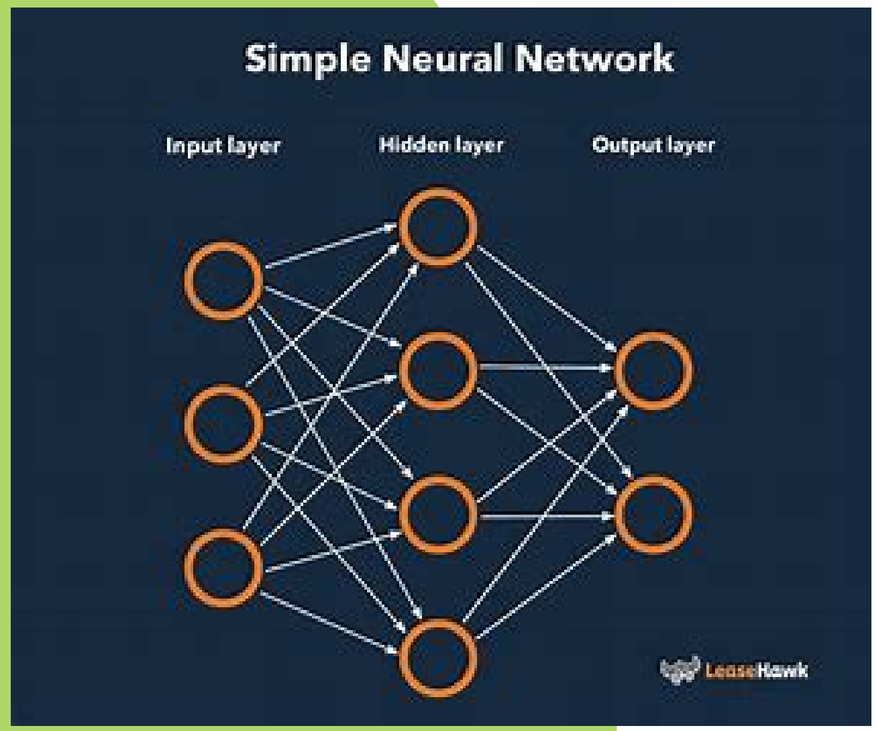
02 Fast results

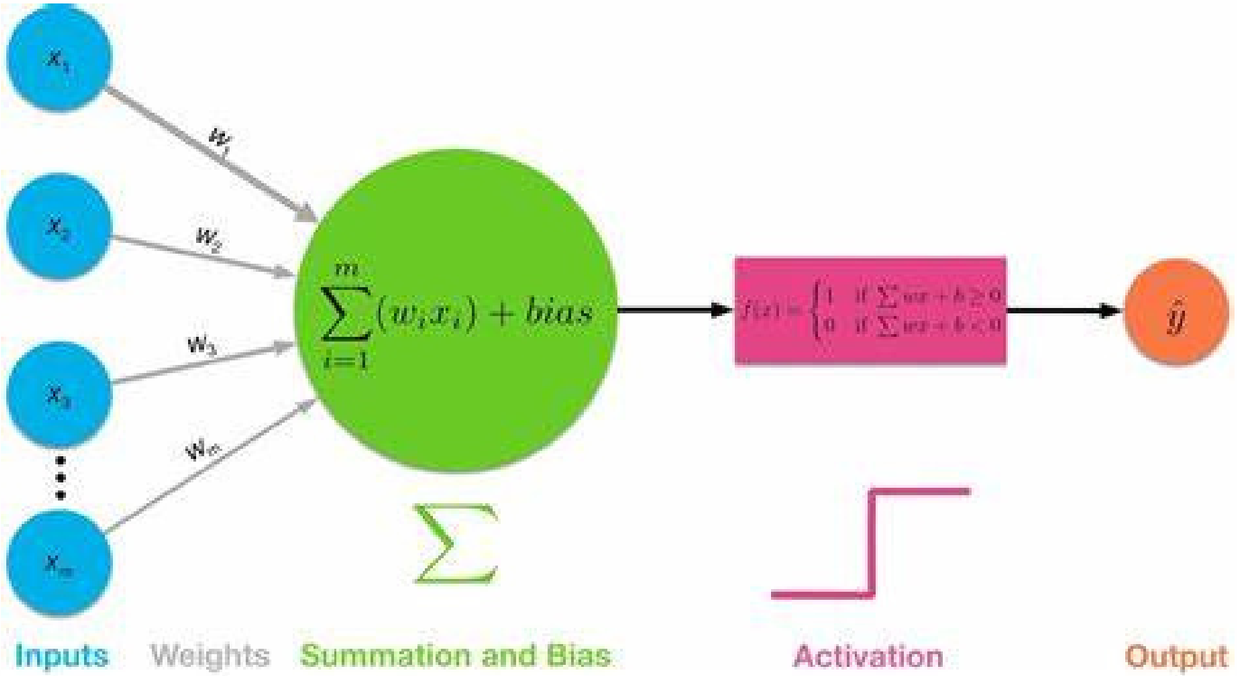
03 Very easy to use

04 Good way to get intuitive results

05 Eliminates middle steps, we get directy the results

How does it work





Keras

Hyperparameters

- Number of layers and neurons
- Epocs: how much the data is “seen”
- Learning rate
- Activation functions

Alpha MC

When training AI we need big amounts of known, controlled data.

We use simulations

Alpha MC is a Monte Carlo program for simulations of fast alpha particle transport in complex geometries

We can desing our target by giving it's geometry and material to AlphaMc Then we run the simulation of the alpha particles colision with the target

Automation of AlphaMC

```
v2.f90
program auto
    integer, parameter :: n_espectro = 1000
    real, parameter :: grosor_min = 100.d-7, grosor_max = 2000.d-7
        real*8:: grosor, incremento
    integer::i,j,k
    real, allocatable::label(:)
    allocate(label(n_espectro))
    incremento = (grosor_max - grosor_min)/n_espectro

    grosor=0
    do i=1,n_espectro
        grosor = grosor + incremento
        open(0,file="geom.f")
        write(0,*)"                module geometry_mod"
        write(0,*)"                    integer, parameter :: nmaterial=4"
        write(0,*)"                    real*8, parameter :: rcha=1. "
        write(0,*)"                    real*8, parameter :: hcha=10."
        write(0,*)"                    real*8, parameter :: zwcha=0.d0"
        write(0,*)"                    real*8, parameter :: xcha=0.d0"
        write(0,*)"                    real*8, parameter :: ycha=0.d0"
        write(0,*)"                    real*8, parameter :: zcha=zwcha+hcha/2.d0"
        write(0,*)"                    integer           :: idcha"
        write(0,*)"                    real*8, parameter :: rtar=1.d0"
        write(0,*)"                    real*8, parameter :: htar=",grosor
        write(0,*)"                    real*8, parameter :: zwtar=8.0"
        write(0,*)"                    real*8, parameter :: xtar=0. "
        write(0,*)"                    real*8, parameter :: ytar=0."
        write(0,*)"                    real*8, parameter :: ztar=zwtar+htar/2."
        write(0,*)"                    integer           :: idtar "
        write(0,*)"                    real*8, parameter :: sigmatar=0."
        write(0,*)"                    real*8, parameter :: rwin=0.57/2.d0"
        write(0,*)"                    real*8, parameter :: hwin=2.07d-6"
        write(0,*)"                    real*8, parameter :: zwwin=8.5"
        write(0,*)"                    real*8, parameter :: xwin=0.d0"
        write(0,*)"                    real*8, parameter :: ywin=0.d0"
        write(0,*)"                    real*8, parameter :: zwin=zwwin+hwin/2."
        write(0,*)"                    integer           :: idwin"
        write(0,*)"                    real*8, parameter :: rdet=0.57/2.d0"
```

```
        call system ("gfortran   geom.f input_main.f main.f ulgeo.f ulsource.f ulli
    call system ("chmod +x main.exe")
    call system ("time ./main.exe < material.in")
    call system ("rm -f main.exe")
    call system ("rm -f *.mod")

    close(0,status="delete")

    open(0,file="nombre.f90")
    write(0,*)"program nombre"
    write(0,*)"     open (1,file='Edet.csv')"
    write(0,*)"     call rename('Edet.csv','espectro",i,".csv')"
    write(0,*)"end program"



    call system("gfortran nombre.f90 -o nombre.exe")
    call system("./nombre.exe")

    close(0,status="delete")
    label(i)=grosor
end do

open(3,file="labels.csv")


do i=1,n_espectro
    write(3,*)label(i)
end do

end program
```

Automation of AlphaMC

Labels (thickness)

millabels.csv

1	1.89999994E-07
2	3.79999989E-07
3	5.69999997E-07
4	7.59999978E-07
5	9.49999958E-07
6	1.13999999E-06
7	1.32999992E-06
8	1.51999996E-06
9	1.70999999E-06
10	1.89999992E-06
11	2.08999995E-06
12	2.27999999E-06
13	2.47000003E-06
14	2.65999984E-06
15	2.84999987E-06
16	3.03999991E-06
17	3.22999995E-06
18	3.41999998E-06
19	3.60999979E-06
20	3.79999983E-06
21	3.99000010E-06
22	4.17999991E-06
23	4.36999971E-06
24	4.55999998E-06
25	4.74999979E-06
26	4.94000005E-06
27	5.12999986E-06
28	5.31999967E-06
29	5.50999994E-06
30	5.69999975E-06
31	5.89000001E-06
32	6.07999982E-06
33	6.26999963E-06
34	6.45999990E-06
35	6.64999970E-06
36	6.83999997E-06
37	7.02999978E-06

Spectrums

	A	B	C
4	0.315000E-01	0.200000E+01	0.141421E+01
5	0.405000E-01	0.100000E+01	0.100000E+01
6	0.495000E-01	0.100000E+01	0.100000E+01
7	0.585000E-01	0.100000E+01	0.100000E+01
8	0.675000E-01	0.300000E+01	0.173205E+01
9	0.765000E-01	0.200000E+01	0.141421E+01
10	0.855000E-01	0.100000E+01	0.100000E+01
11	0.945000E-01	0.400000E+01	0.200000E+01
12	0.103500E+00	0.100000E+01	0.100000E+01
13	0.112500E+00	0.000000E+00	0.000000E+00
14	0.121500E+00	0.000000E+00	0.000000E+00
15	0.130500E+00	0.100000E+01	0.100000E+01
16	0.139500E+00	0.000000E+00	0.000000E+00
17	0.148500E+00	0.000000E+00	0.000000E+00
18	0.157500E+00	0.100000E+01	0.100000E+01
19	0.166500E+00	0.000000E+00	0.000000E+00
20	0.175500E+00	0.000000E+00	0.000000E+00
21	0.184500E+00	0.000000E+00	0.000000E+00
22	0.193500E+00	0.100000E+01	0.100000E+01
23	0.202500E+00	0.000000E+00	0.000000E+00
24	0.211500E+00	0.100000E+01	0.100000E+01
25	0.220500E+00	0.100000E+01	0.100000E+01
26	0.229500E+00	0.000000E+00	0.000000E+00
27	0.238500E+00	0.000000E+00	0.000000E+00
28	0.247500E+00	0.000000E+00	0.000000E+00
29	0.256500E+00	0.000000E+00	0.000000E+00
30	0.265500E+00	0.000000E+00	0.000000E+00

espectro 6

Listo Accesibilidad: No disponible

Buscar

Clean Spectrum

A1

	A	B	C
1	0.000000E+00		
2	1.0		
3	1.0		
4	2.0		
5	1.0		
6	1.0		
7	1.0		
8	3.0		
9	2.0		
10	1.0		
11	4.0		
12	1.0		
13	0.0		
14	0.0		
15	1.0		
16	0.0		
17	0.0		
18	1.0		
19	0.0		
20	0.0		
21	0.0		
22	1.0		
23	0.0		
24	1.0		
25	1.0		
26	0.0		
27	0.0		

espectro 6

Listo Accesibilidad: No disponible

Buscar

First steps making the IA

One spectrum

Three spectrums

Ten spectrums

Very simple networks to decide the hyperparameters

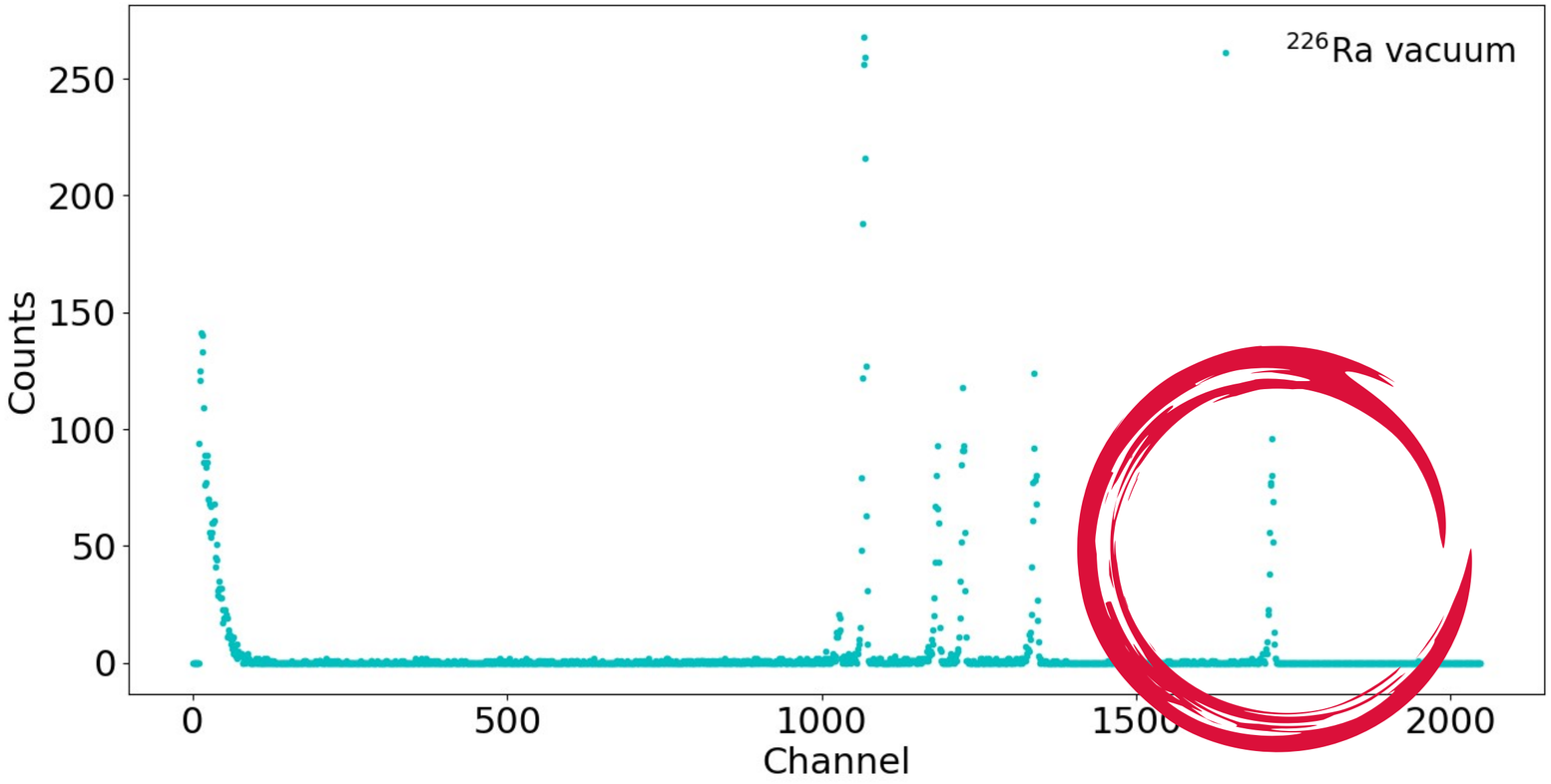
More data (didn't work)

Too complex

GIVING THE WHOLE SPECTRUM TO THE NETWORK WAS LEADING TO INCONCLUSSIVE RESULTS.

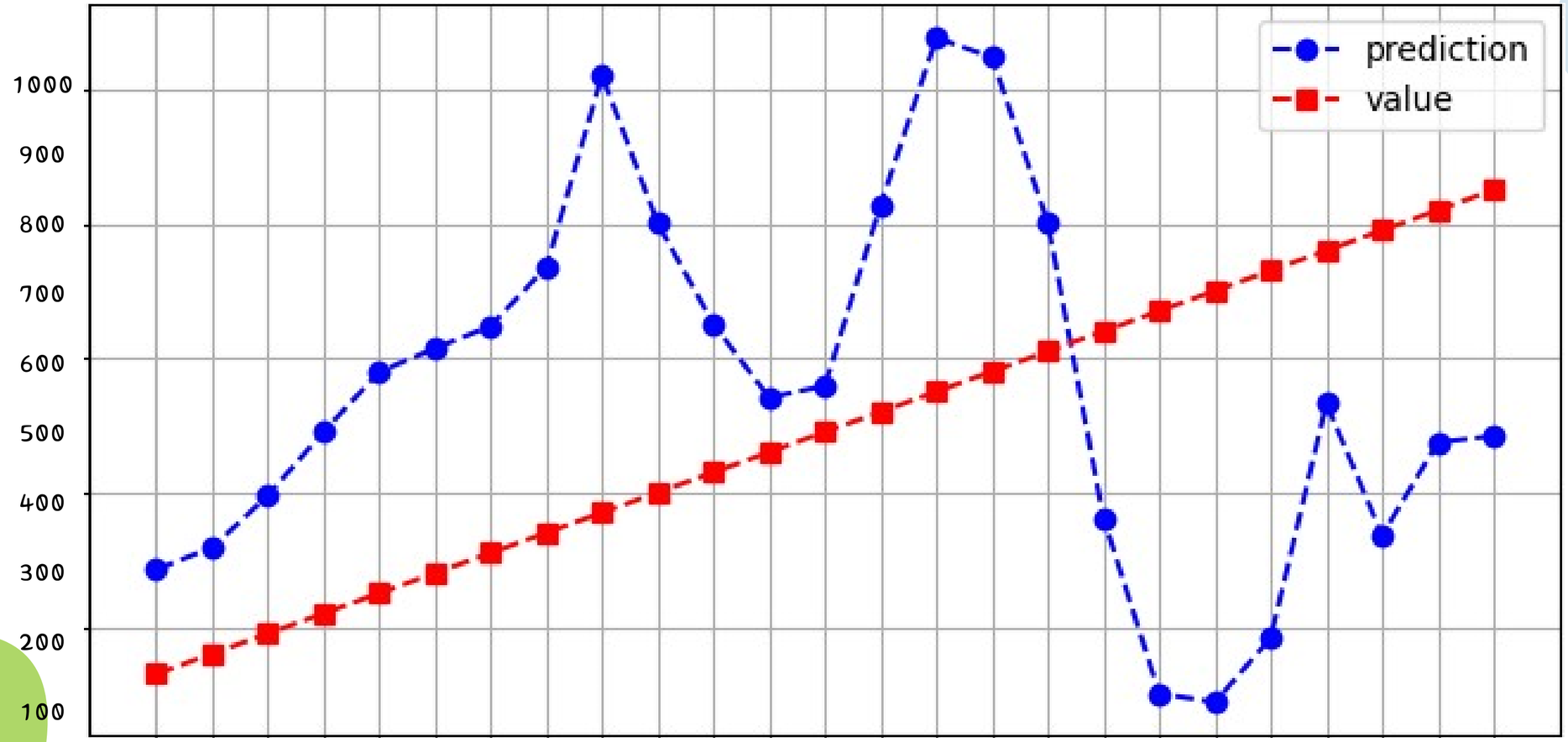
SIMPLIFICATION

Analyze the spectrums partially


^{226}Ra vacuum
Counts
Channel
0
50
100
150
200
250
0
500
1000
1500
2000

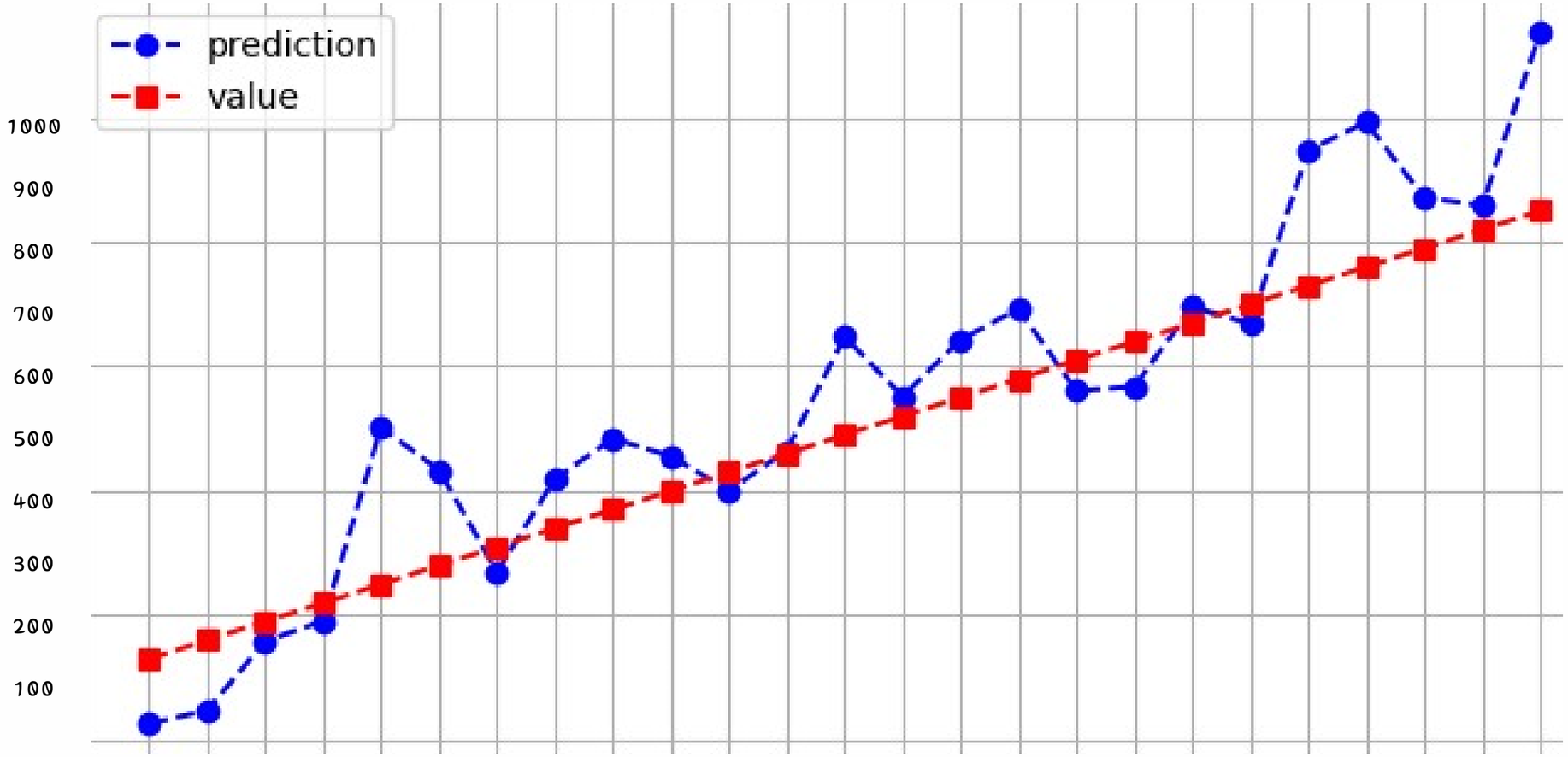
100 spectrums

Thickness (nm)

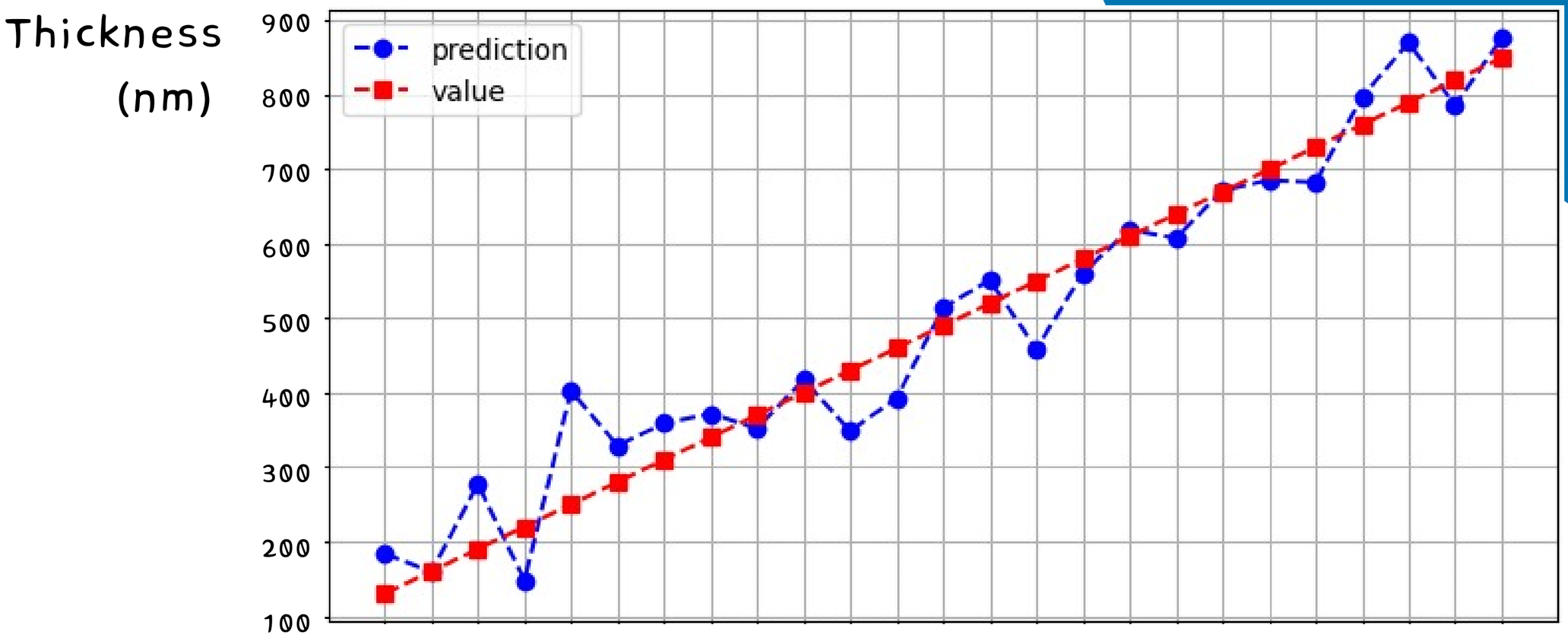


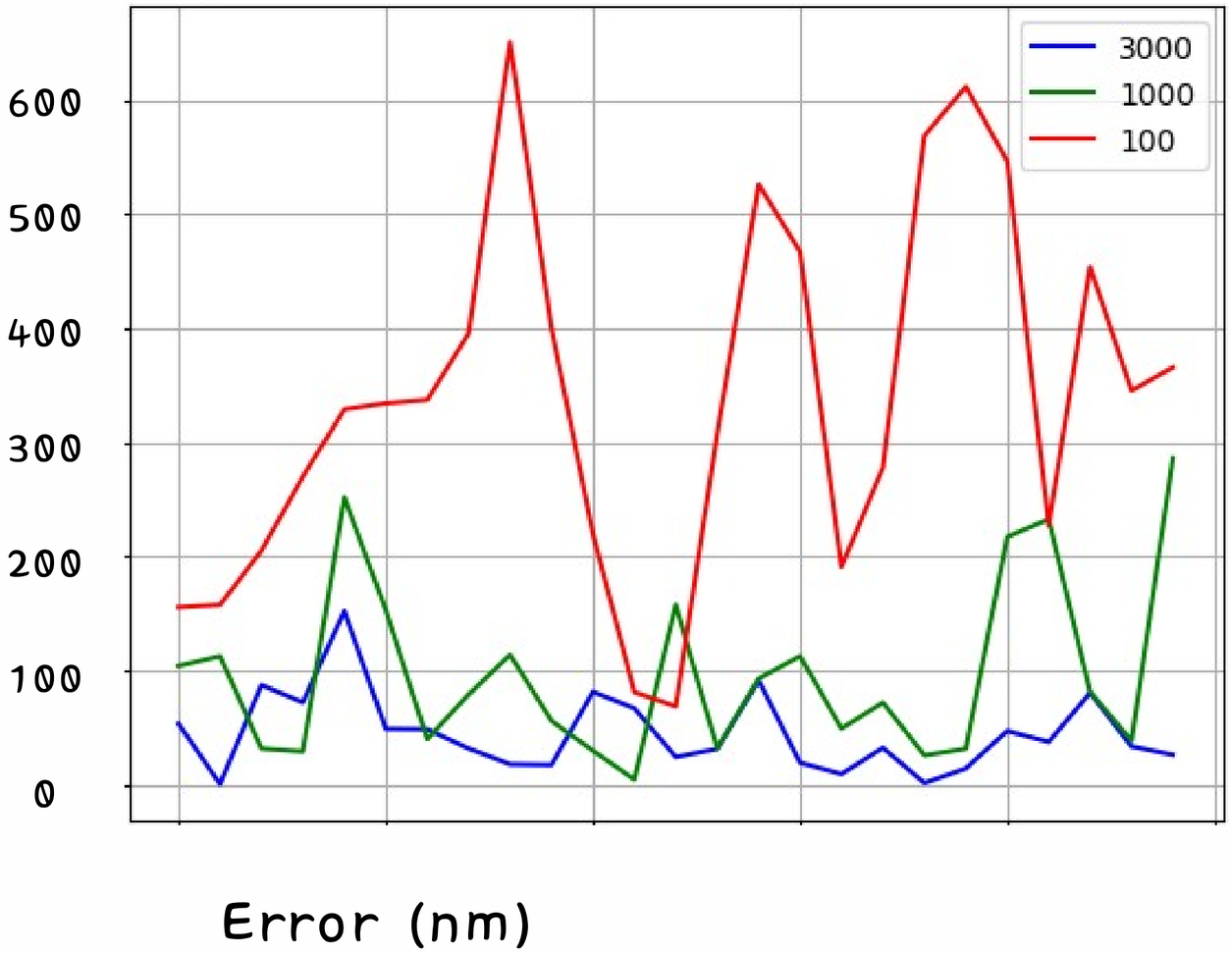
1000 spectrums

Thickness (nm)

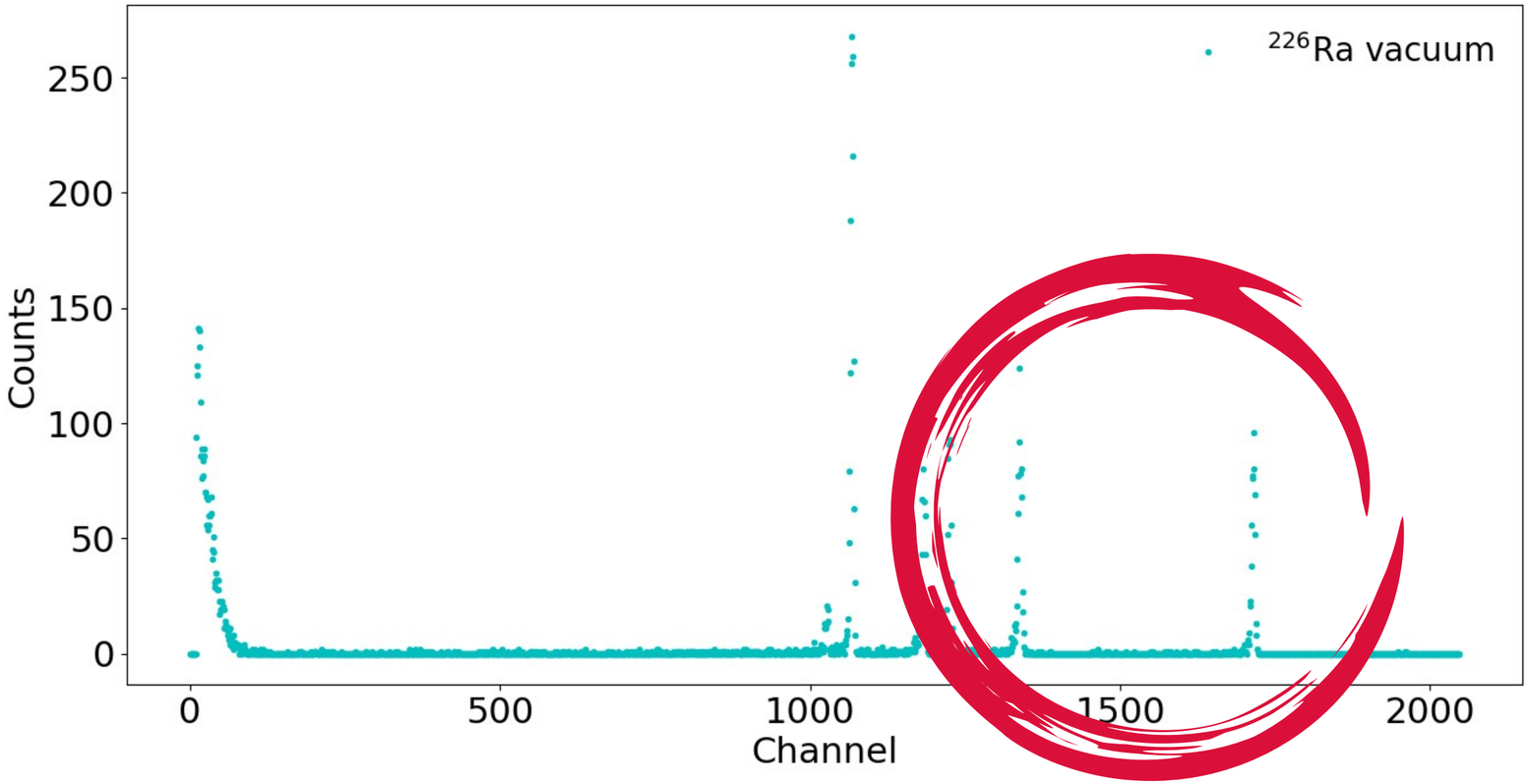


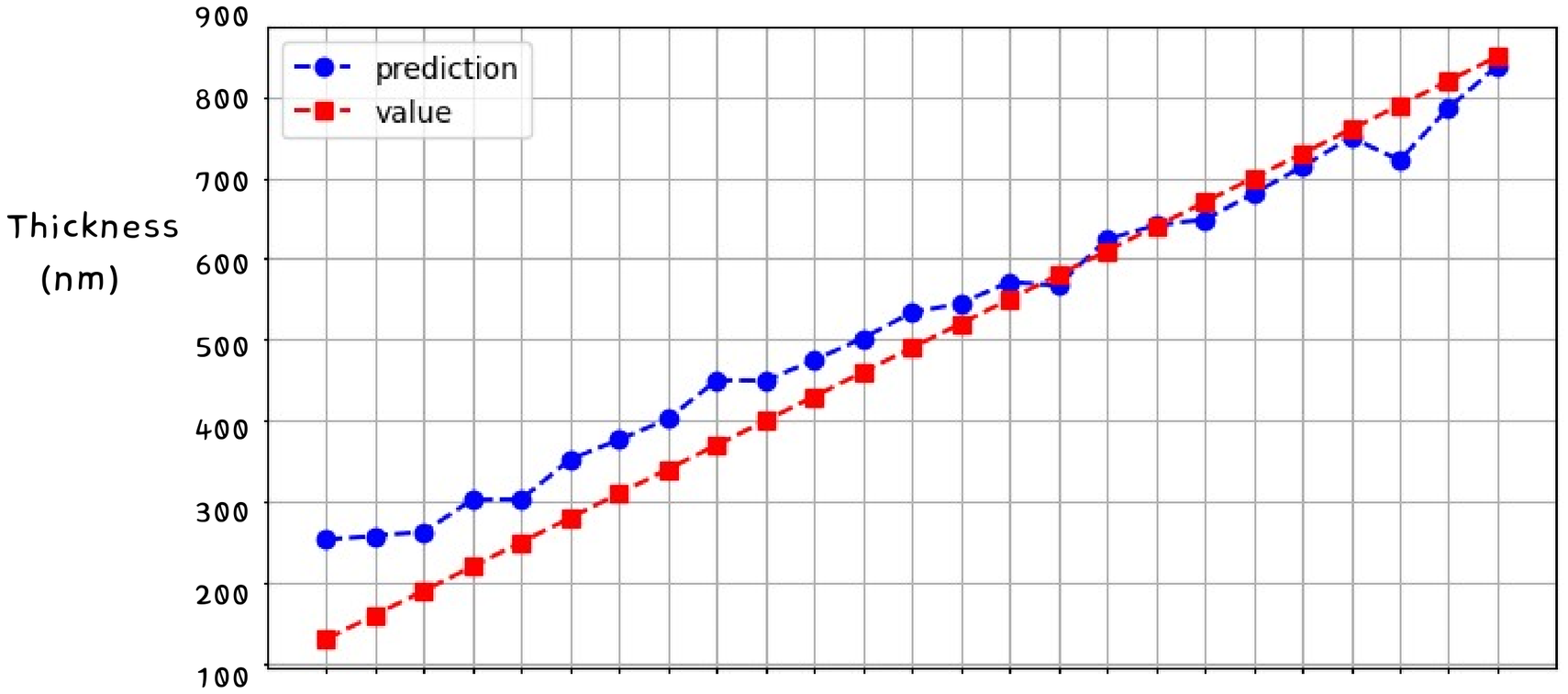
3000 spectrums

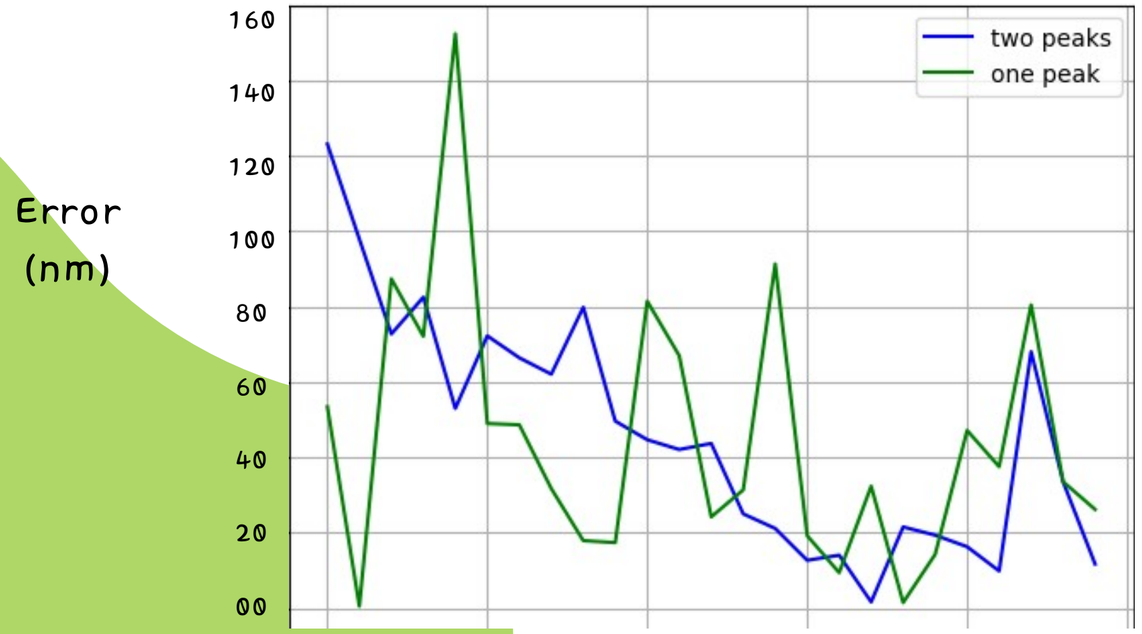




It seems it would eventually converge with more data


^{226}Ra vacuum
Counts
250
200
150
100
50
0
0
500
1000
1500
2000
Channel





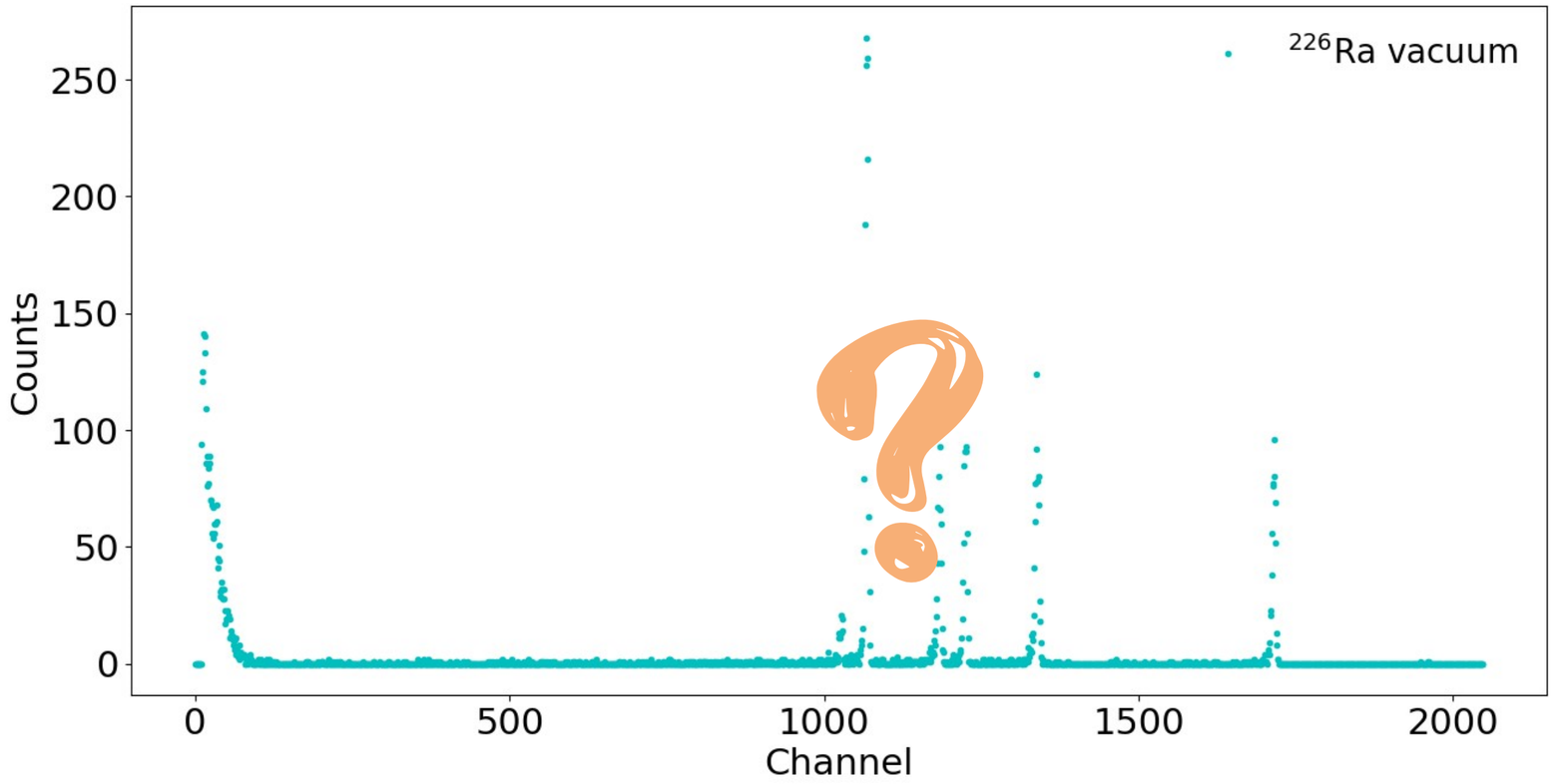
Mean error with one peak:

45,84 nm

Mean error with two peaks:

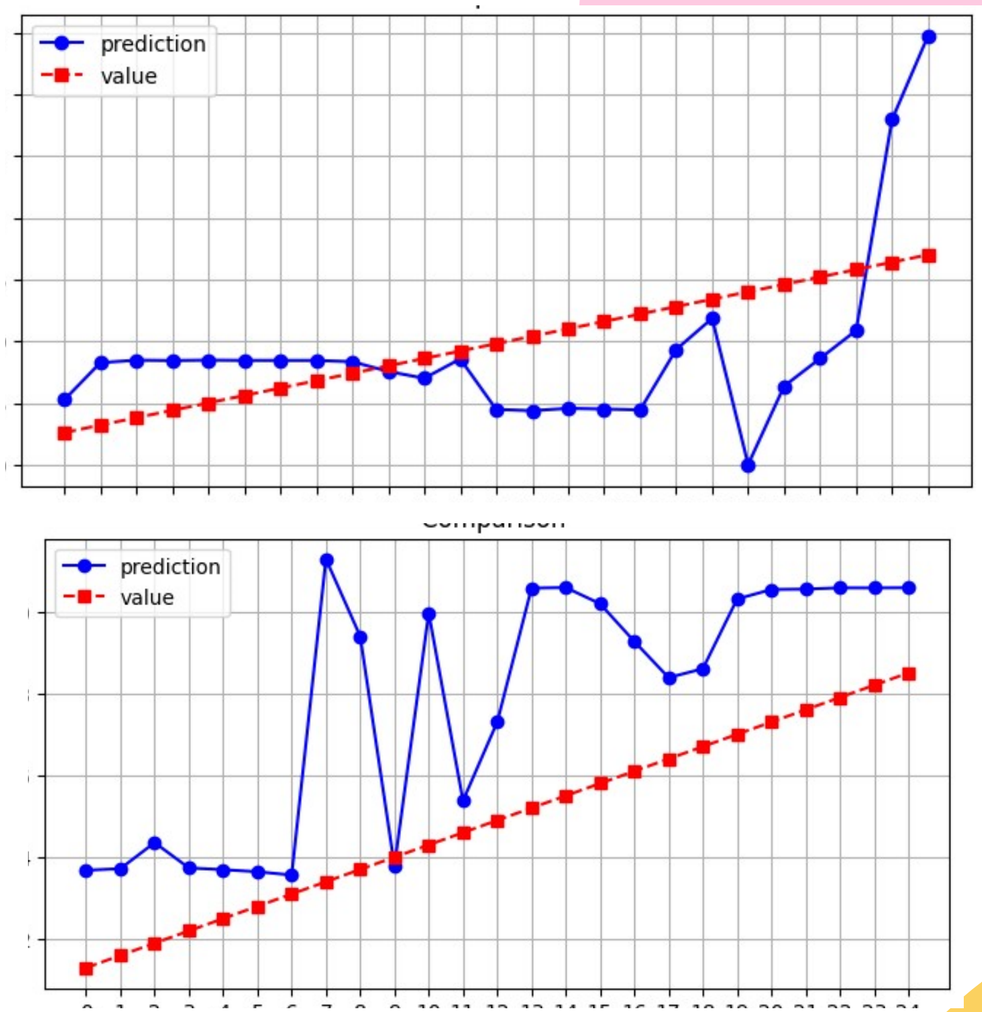
45,14 nm

When does it stop working?



Each train would get very different results, even using the same data and network

For each set of data I would use the best adjusted one after a few trials



The network I'm using

Epochs: 1000,100,30

Batch size: 5

	layers	
1.	100	relu
2.	90	sigmoid
3.	80	sigmoid
4.	70	sigmoid
5.	60	relu
6.	50	relu
7.	40	relu
8.	30	relu
9.	20	relu
10.	10	relu
11.	1	relu

Thanks

Questions?