DESY
summer student project:

ML for ASTRA simulations

Sergei Kladov
Novosibirsk, 29.07.21

HELMHOLTZ RESEARCH FOR GRAND CHALLENGES

About me

Idk, maybe you are interested?

Education + Job experience + research

- Finished 1st masters degree year (NSU Physics)
- Have shifts on the VEPP-2000 as an operator-physicist
- Maintain a simple Java program on the VEPP-2000

- Researches:
- Nonlinear dynamics on the main linear resonance
- Two-stream effects on the main linear resonance
- Both are published as posters at the IPAC21

Interests

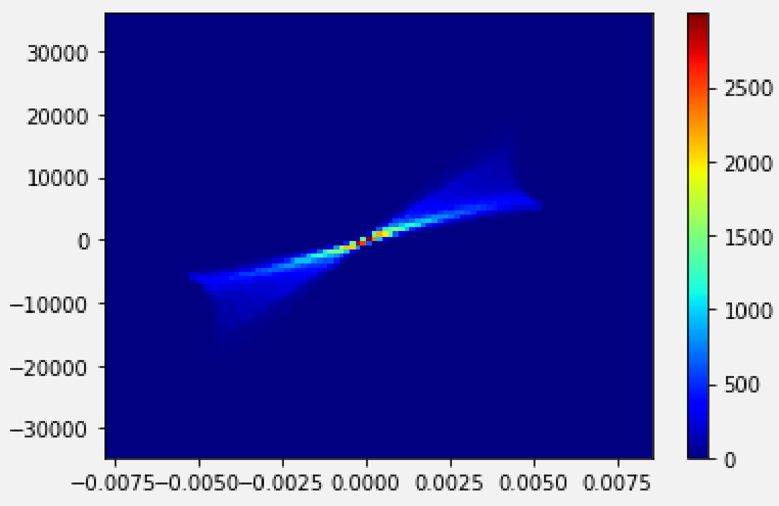
- Started programming in 11.20. What have I done:
- Wrote a simple simulation program for the beam-beam in circular colliders on Java
- Finished the NN course in python, fulfilled the 6 complex NNs tasks
- Rewrote the “alarmer” Java program for the VEPP-2000
- Made a basics of a 2d game in the plain Java, JavaFX, LiteEngine, LibGDX; and a core of a 3d game with multiplayer in Unity (c#)

The problem and the goal

My task

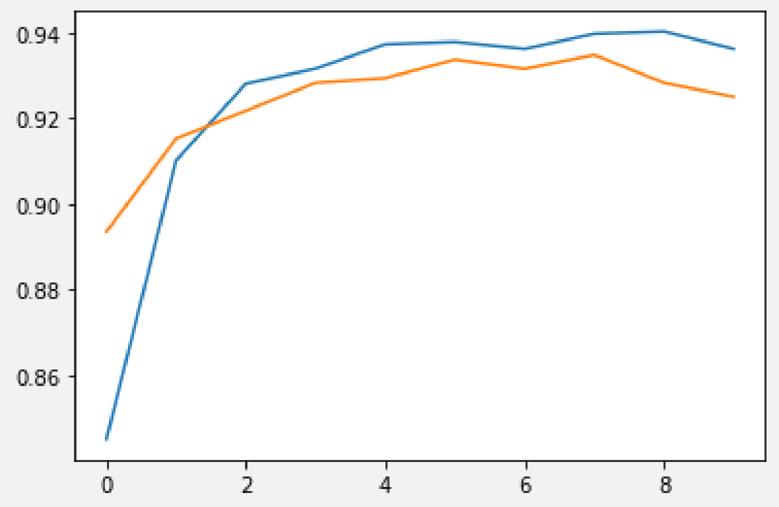
Problem

- Simulations take a lot of time
- Precize results can be obtained only by these time-consumung simulations



Goal

- Make a decent estimation of the output parameters based on the input parameters



Methods. ML

Nonlinear regression

Neural network

- An ultimate weapon
- Was used by other groups
- Has a great variety of parameters
- I am familiar with them
- Easy to make an overkill

Others

- Variety of methods: decision trees, ridge and others
- Simple data flow (trees)
- Tend to be overfitted or underfitted
- A bit faster
- Discrete (trees)

Realization

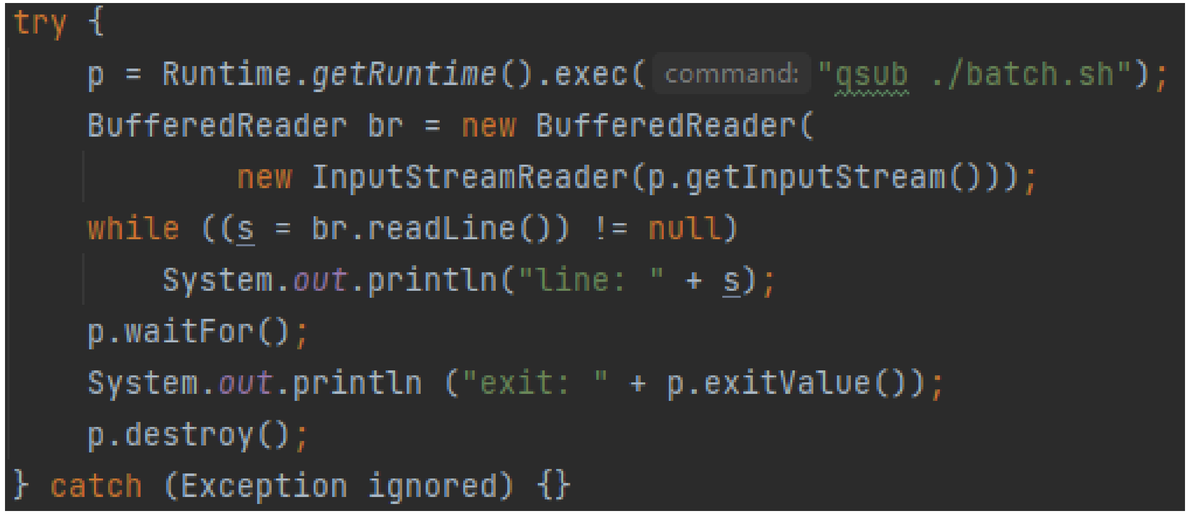
Step 1

Beginning

- Do not have a dataset
- Prepare all the necessary files in the DESY working directory
- Progress for now: not automated

Performing the simulations. Java program

- Changes the input parameters
- sends the job to the cluster (Astra simulations)
- Progress for now: 90%



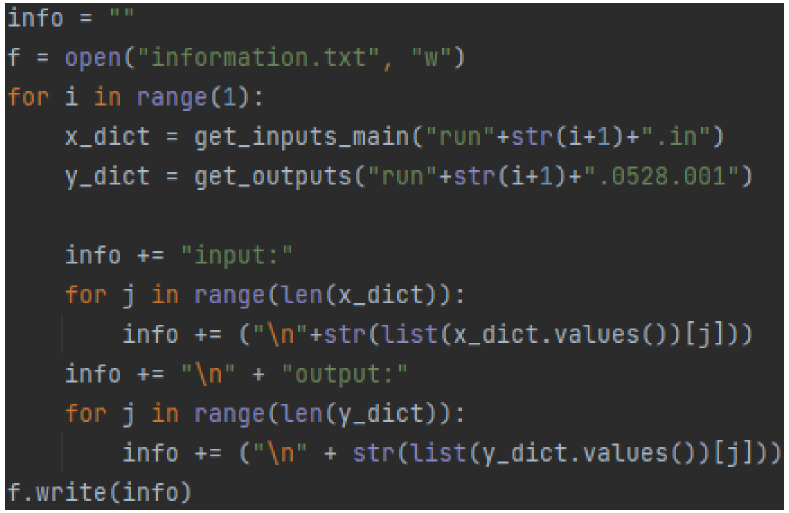
```
try {
    p = Runtime.getRuntime().exec( command: "qsub ./batch.sh");
    BufferedReader br = new BufferedReader(
            new InputStreamReader(p.getInputStream()));
    while ((s = br.readLine()) != null)
        System.out.println("line: " + s);
    p.waitFor();
    System.out.println ("exit: " + p.exitValue());
    p.destroy();
} catch (Exception ignored) {}
```

Realization

Step 2

Preparing a dataset. Python script

- Should be executed when when the cluster finishes his job
- collects all the necessary information about input and output beam properties from all simulations
- Saves obtained information to the single lightweight file
- Progress for now: 100%

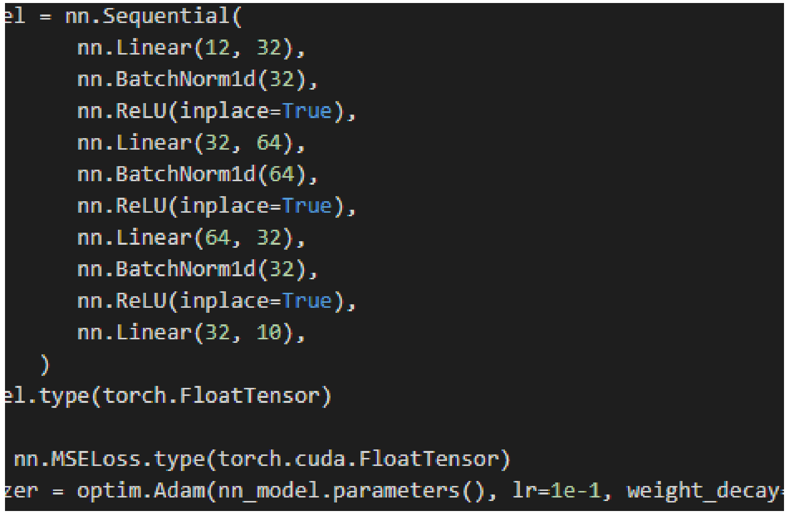


```
info = ""
f = open("information.txt", "w")
for i in range(1):
    x_dict = get_inputs_main("run"+str(i+1)+".in")
    y_dict = get_outputs("run"+str(i+1)+".0528.001")

    info += "input:"
    for j in range(len(x_dict)):
        info += ("\n"+str(list(x_dict.values())[j]))
    info += "\n" + "output:"
    for j in range(len(y_dict)):
        info += ("\n" + str(list(y_dict.values())[j]))
f.write(info)
```

Training the ML! Google colab

- Copy the lightweight file to my Google Drive
- I am playing with ML parameters as I wish, obtain results
- Progress for now: 10% (I wrote the cores of NN and deceison trees, but without a dataset it is pointless)



```
el = nn.Sequential(
        nn.Linear(12, 32),
        nn.BatchNorm1d(32),
        nn.ReLU(inplace=True),
        nn.Linear(32, 64),
        nn.BatchNorm1d(64),
        nn.ReLU(inplace=True),
        nn.Linear(64, 32),
        nn.BatchNorm1d(32),
        nn.ReLU(inplace=True),
        nn.Linear(32, 10),
    )
el.type(torch.FloatTensor)

 nn.MSELoss.type(torch.cuda.FloatTensor)
zer = optim.Adam(nn_model.parameters(), lr=1e-1, weight_decay
```

Thank you

Contact

DESY. Deutsches
Elektronen-Synchrotron

www.desy.de

Sergei Kladov
PITZ
s.kladov@g.nsu.ru
+7 923 239 70 24