train_LaengeNetMod

February 7, 2022

[26]:
```
from Neural_Nets.LaengeNetModified.Development.LaengeNetModified import␣
↪LaengeNetModified, LaengeNetModifiedLossFunc
from Neural_Nets.ThermoDatasetModified.Development.ThermoDatasetModified import␣
↪ThermoDatasetModified
from Utils.PlotHandler.Development.PlotHandler import PlotHandler
import torch
from torch.utils.data import DataLoader, Dataset
import torch.nn as nn
from torch.optim import *
from Data_Handling.SGTEHandler.Development.SGTEHandler import SGTEHandler
import numpy as np
import matplotlib.pyplot as plt
```

[2]:
```
def epoch(net: LaengeNetModified, dataloader, loss_func, optimizer):
    epoch_losses = np.zeros([len(dataloader), ])

    for i, (temp, target) in enumerate(dataloader):
        temp = temp.unsqueeze(-1)

        # Forward pass
        prediction = net(temp.float())

        # Get the loss
        loss = loss_func(prediction, target)

        # Backward pass
        net.zero_grad()
        loss.backward()
        optimizer.step()
        epoch_losses[i] = loss

    mean_epoch_loss = epoch_losses.mean(axis=0)
    #print('Mean epoch loss: ', mean_epoch_loss)
    return mean_epoch_loss
```

[73]:
```
def train(net, dataset):
    # Hyperparameters
```

```
    n_epochs = 5000
    lr = 0.001
    batch_size = 64

    # Data
    dataloader = DataLoader(dataset, batch_size=batch_size, shuffle=False)

    # Optimizer
    #optimizer = RMSprop(net.parameters(), lr=lr)
    optimizer = AdamW(net.parameters(), lr=lr)
    loss_func = LaengeNetModifiedLossFunc()

    losses = []

    best_loss = epoch(net, dataloader, loss_func, optimizer)
    best_net = net

    for i in range(n_epochs):
        #print('-----\nEpoch %i:\n' % i)
        loss = epoch(net, dataloader, loss_func, optimizer)
        losses.append(loss)

        if loss < best_loss:
            best_net = net

        if i % 10 == 0:
            print(loss)

    return losses, net
```

[74]:
```
net = LaengeNetModified(hidden_dim=4)

element = 'Fe'
phase = ['BCC_A2']
start_temp, end_temp = 200, 2000

dataset = ThermoDatasetModified(element, phase, step=2, start_temp=start_temp,␣
 ↪end_temp=end_temp)

losses, best_net = train(net, dataset)
```

```
Fe successfully selected!

1430.8434055873327
109.60085541861397
106.42783655439105
106.31332887922015
106.0709969656808
```

105.74783042498997
105.36922178949628
104.95056969778878
104.50204069273812
104.03074952534267
103.54200318881443
103.03946781158447
102.52631092071533
102.00490065983364
101.47707135336739
100.94459744862148
100.40828398295811
99.86929195267814
99.32852240971157
98.78642048154559
98.24366474151611
97.70058172089713
97.15747465406146
96.61482432910374
96.07252539907184
95.53094553947449
94.9901829447065
94.45025065967015
93.91128294808524
93.37320137023926
92.83632040023804
92.30027001244682
91.76536164964948
91.23149888856071
90.69859576225281
90.16660714149475
89.63558285576957
89.10525287900653
88.575674363545
88.0466957773481
87.51782471793038
86.98896091324943
86.4597304207938
85.92980715206691
85.3984786442348
84.86560344696045
84.3306359904153
83.79313049997602
83.25281081880841
82.70913757596698
82.1618595123291
81.61084754126412
81.05586893217904

80.49684953689575
79.93362018040249
79.36631338936942
78.79501499448504
78.21968177386692
77.64063385554722
77.05794402531215
76.47184467315674
75.88262040274483
75.29058769771031
74.6959947858538
74.09914602552142
73.50023109572274
72.89953998156956
72.29748221806118
71.69443781035287
71.09039623396737
70.48585963249207
69.88100494657245
69.27596950531006
68.67105071885246
68.06651289122445
67.46241075651986
66.85901117324829
66.25653811863491
65.65507102012634
65.05469509533474
64.45559787750244
63.857843092509675
63.26152385984148
62.66670172555106
62.073520047324045
61.48199653625488
60.89220012937273
60.30412939616612
59.71794523511614
59.13363415854318
58.55136316163199
57.97104338237217
57.39271477290562
56.8165545463562
56.242504835128784
55.670625822884695
55.1010411466871
54.53368520736694
53.96884788785662
53.406361034938264
52.84639699118478

52.288917200905935
51.734227657318115
51.18220373562404
50.63298429761614
50.086620807647705
49.54320573806763
49.00289811406817
48.46568390301296
47.93162727355957
47.40097277505057
46.87359840529306
46.34988250051226
45.82954260281154
45.31293191228594
44.80002123968942
44.29099491664341
43.78591002736773
43.28472948074341
42.78765835080828
42.29478406906128
41.80604376111712
41.32166460582188
40.84176427977426
40.366380895887104
39.89541588510786
39.429103817258564
38.96752854755947
38.510598591395784
38.058513811656404
37.611304828098845
37.16900822094509
36.73159074783325
36.29926235335214
35.87192610331944
35.44971122060503
35.03259130886623
34.620652607509065
34.21390158789499
33.81226961953299
33.41576279912676
33.02462175914219
32.63860147339957
32.257865122386384
31.882268735340663
31.511834689549037
31.146666731153214
30.786675930023193
30.431684562138148

30.081894533974783
29.737102729933603
29.397361636161804
29.062606522015162
28.732659544263566
28.407724039895193
28.087428314345225
27.771987455231802
27.46110495499202
27.15489946092878
26.85306215286255
26.55571118422917
26.26260733604431
25.97370912347521
25.688979421343124
25.40824749640056
25.131486688341415
24.858397841453552
24.589173010417394
24.323373513562338
24.061195654528483
23.802285177367075
23.546747497149877
23.294254081589834
23.044891229697637
22.798432094710215
22.554857679775782
22.314002513885498
22.075814425945282
21.840201641832078
21.6071309873036
21.37635281256267
21.147968215601786
20.921875987734115
20.697902739048004
20.47614209992545
20.25642968927111
20.03883332014084
19.82316553592682
19.609508097171783
19.397769502231053
19.18798984800066
18.980006882122584
18.773900764329092
18.569709062576294
18.36722879750388
18.166664753641403
17.967805343014852

17.77071737391608
17.57552763393947
17.381995499134064
17.19036078453064
17.00045508146286
16.812334256512777
16.62600679056985
16.44143953493663
16.25872435740062
16.07782472882952
15.898669864450182
15.721412556512016
15.545942987714495
15.37234332731792
15.200569033622742
15.030681022575923
14.862630537578038
14.696513380323138
14.532196419579643
14.36988034418651
14.209417121750969
14.05084582737514
13.894327368055071
13.73969052519117
13.586969801357814
13.436267180102211
13.287496881825584
13.140666246414185
12.99582404749734
12.852995301995959
12.712069775377001
12.57318844965526
12.43631672007697
12.301428828920637
12.16855492762157
12.037687940256935
11.908750687326703
11.781767393861498
11.656943653311048
11.533988731248039
11.413008059774127
11.294087520667485
11.177132155214037
11.062083235808782
10.94898670060294
10.83794183390481
10.728801258972712
10.621584568704877

10.516386883599418
10.413132207734245
10.311656628336225
10.212118753365107
10.114533943789345
10.018763550690242
9.924814547811236
9.832704876150403
9.742494591644832
9.654002904891968
9.567231816904885
9.482277342251368
9.399130276271276
9.317677685192653
9.237907869475228
9.159958175250463
9.083978406020574
9.010497450828552
8.94043790442603
8.876079371997289
8.822942129203252
8.79479627949851
8.827016651630402
9.01863729102271
9.68739983013698
12.0819580831698
22.10442921093532
21.749824319567
9.69434071438653
8.315440016133445
8.11368333867618
8.106046344552722
8.206770615918296
8.509455225297383
9.421480108584676
12.367272291864667
18.468969404697418
14.790341453892845
9.766744847808566
8.584238671830722
8.395864597388677
8.61689424089023
9.401257500052452
11.425324989216668
14.626625452722822
13.784236771719796
10.461879866463798
9.075154436486107

8.789399457829338
9.090984016656876
10.069300104464803
11.869081701551165
13.027732261589595
11.577515474387578
9.861571882452283
9.14968493793692
9.139403421963964
9.685861898320061
10.76191440650395
11.741557163851601
11.375928248677935
10.184025364262718
9.406744484390531
9.157281271048955
9.393520793744496
10.014968671969005
10.735169768333435
10.78697497504098
10.112716266087123
9.394604380641665
9.085059664079122
9.166704603603907
9.566902590649468
10.062198383467537
10.221749484539032
9.854541514601026
9.331513183457512
9.021044675792966
9.014105426413673
9.261022559234075
9.60302495104926
9.760006283010755
9.56354693855558
9.188489547797612
8.910533509084157
8.850240784032005
8.991321052823748
9.22223950283868
9.353576200348991
9.25016405752727
8.988291280610222
8.757109931537084
8.674067016158785
8.747712450368065
8.902140302317482
9.004404681069511

8.949913271835872
8.766668443168912
8.581114964825767
8.49368274637631
8.525277721030372
8.626433857849666
8.701967656612396
8.672998679535729
8.542609938553401
8.39581686258316
8.313476170812335
8.32109786782946
8.386767617293767
8.441484885556358
8.423341623374395
8.328267429556165
8.212704002857208
8.139429117952075
8.133737249033791
8.175613573619298
8.21405553817749
8.200604030064174
8.128593291555132
8.036529447351183
7.972736835479736
7.960487834044865
7.9863607713154385
8.011591651609965
7.998874953814915
7.940902956894466
7.865470894745418
7.810404905251095
7.7947660599436075
7.80895570559161
7.823200672864914
7.8084215223789215
7.758621824639184
7.695095820086343
7.6471923142671585
7.630128517746925
7.63637261731284
7.642252709184374
7.625278081212725
7.581541580813272
7.528105635728155
7.487931526132992
7.472682816641671
7.476009560482843

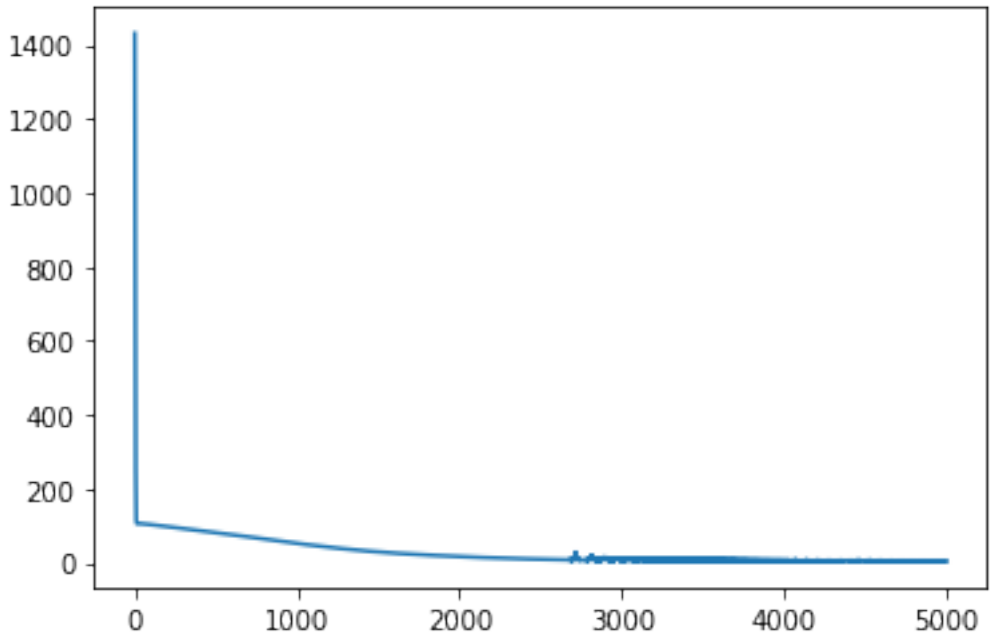
7.478782336626734
7.464448756405285
7.430832747902189
7.391660526394844
7.364754689591272
7.358226290770939
7.365758753248623
7.37160373372691
7.361908925431115
7.335529733981405
7.294745626194136
7.277942966137614
7.272807487419674
7.264731453997748
7.248646619064467
6.804222015397889
7.112335394535746
7.535396082060678
7.704179661614554
7.40931037919862
6.942216479352543
6.6935208056654245
6.739460089376995
7.032658892018454
7.406927181141717
7.528233287589891
7.2431730436427255
6.850572343383517
6.643951433045523
6.690056294202805
6.942168923360961
7.241874605417252
7.317650371364185
7.073816231318882
6.752956501075199
6.58949929688658
6.642876999718802
6.866304455058915
7.10823381798608
7.145587516682489
6.93097077522959
6.6742376940591
6.563027852347919
6.643259161284992
6.854645154305866
7.0417750626802444
7.025308023606028
6.823666006326675

6.636726275086403
6.60761604138783
6.735746628471783
6.899400696158409
6.921996593475342
6.983169802597591
6.2863796438489645
6.465467189039503
7.492357747895377
7.852759540081024
6.945178866386414
6.1544038610798975
5.923118216650827
6.128898407731738
6.740388397659574
7.494572188173022
7.501876550061362
6.772152677178383
6.241359336035592
6.160821495311601
6.462943858333996
6.92650651718889
7.167598894664219
6.9948757120541165
6.59765117083277
6.375545574086053
6.384400874376297
6.634329395634787
6.8708400428295135
6.888458247695651
6.693346655794552
6.4924042565482
6.450632101723126
6.583213925361633
6.645957154887063
6.816508853009769
6.672829934528896
6.610545437250819
6.461066686681339
6.570668446166175
6.545036613941193
6.688946536609104
6.596441830907549
6.637412003108433
6.4917678620134085
6.574570419532912
6.499507731625012
6.6350924713271

```
6.536262067300933
6.611916965671948
6.487551850931985
6.573473359857287
6.483573130198887
6.5885471148150305
6.499231070280075
6.591306920562472
6.461956558482988
6.554969180907522
6.452341430953571
6.567328738314765
6.4822738745382855
6.556933662721089
6.427076986857823
```

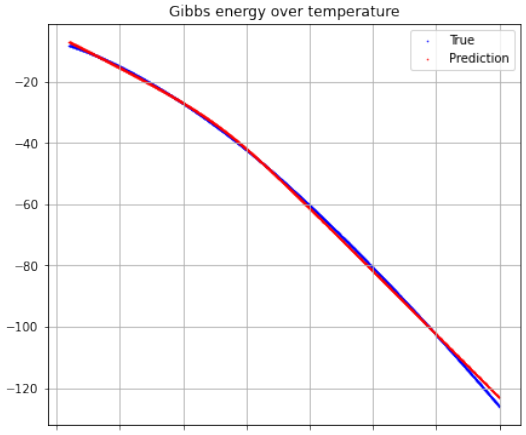
[75]:
```
plt.plot(range(len(losses)), losses)
```

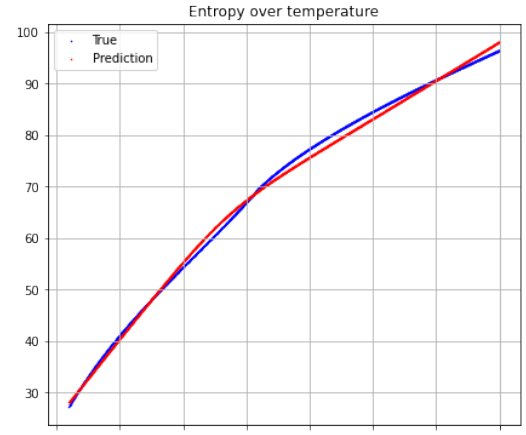
[75]:
```
[<matplotlib.lines.Line2D at 0x1aa6249b5b0>]
```

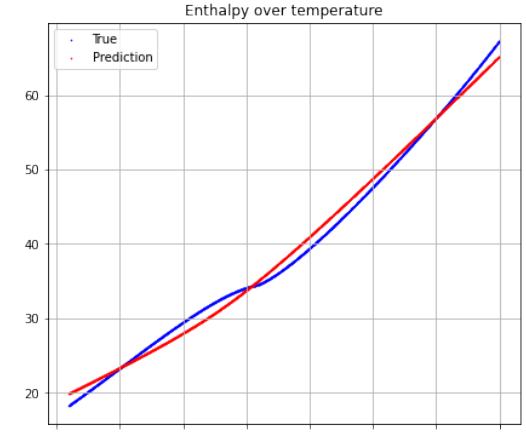


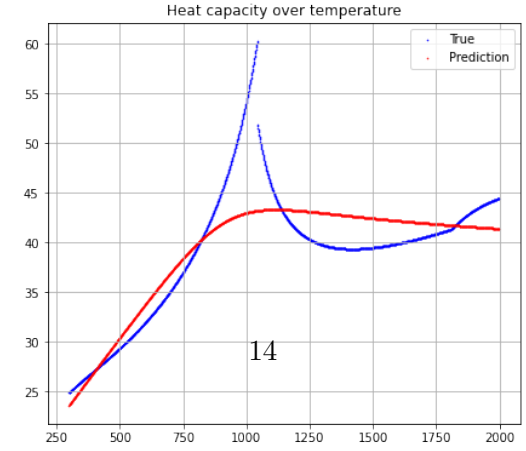
[76]:
```
ph = PlotHandler('Laenge')

ph.properties_temp_modified(best_net, element, phase, start_temp=start_temp,␣
↪end_temp=end_temp)
```

```
Fe successfully selected!
```


Gibbs energy over temperature
True
Prediction
−20
−40
−60
−80
−100
−120


Entropy over temperature
True
Prediction
100
90
80
70
60
50
40
30


Enthalpy over temperature
True
Prediction
60
50
40
30
20


Heat capacity over temperature
True
Prediction
60
55
50
45
40
35
30
25
250
500
750
1000
1250
1500
1750
2000

[39]:
```
torch.save(net, 'Neural_Nets/LaengeNetModified/Models/model_07_02_22_1937')
```

[ ]:
```

```