train_LaengeNet_all

February 5, 2022

[1]:
```
from Neural_Nets.LaengeNet.Development.LaengeNetTorch import LaengeNet,␣
↪LaengeNetLossFunc
from Neural_Nets.ThermoDataset.Development.ThermoDataset import ThermoDataset
from Neural_Nets.ThermoNetActFuncs.Development.ThermoNetActFuncs import␣
↪ChenSundman, Softplus
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 Rprop
from Data_Handling.SGTEHandler.Development.SGTEHandler import SGTEHandler
import numpy as np
import matplotlib.pyplot as plt
```

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

    for i, (temp, g, s, h, c) in enumerate(dataloader):
        temp = temp.unsqueeze(-1)

        # Input scaling
        #temp /= temp.max()

        # Forward pass
        gibbs_energy, entropy, enthalpy, heat_cap = net(temp, temp, temp, temp,␣
↪debug=True)

        # Output scaling
        #scale = 100000
        scale = 1
        gibbs_energy, entropy, enthalpy, heat_cap = gibbs_energy/scale, entropy/
↪scale, enthalpy/scale, heat_cap/scale
        g, s, h, c = g/scale, s/scale, h/scale, c/scale

        # Get the loss
        loss = loss_func(gibbs_energy.float(), g.float(), entropy.float(), s.
↪float(), enthalpy.float(), h.float(),
```

```
                        heat_cap.float(), c.float(), debug=False)

        # Backward pass
        net.zero_grad()
        loss.backward()
        #torch.nn.utils.clip_grad_norm_(net.parameters(), 100)
        optimizer.step()
        epoch_losses[i] = loss

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

[3]:
```
def train(net, dataset):
    # Hyperparameters
    n_epochs = 100
    lr = 0.01
    batch_size = 64
    loss_weights = [0.01, 100000, 0.01, 100000]

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

    # Optimizer
    optimizer = Rprop(net.parameters(), lr=lr)
    loss_func = LaengeNetLossFunc(weights=loss_weights)

    losses = []

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

[4]:
```
net = LaengeNet(init_args=(-0.2, -0.1))

element = 'Fe'
phase = ['BCC_A2']
dataset = ThermoDataset(element, phase, scaling=False)

train(net, dataset)
```

```
Fe successfully selected!

-----
Epoch 0:
```

```
C:\Users\danie\anaconda3\envs\5_Programmcodes\lib\site-
packages\torch\nn\modules\loss.py:520: UserWarning: Using a target size
(torch.Size([64])) that is different to the input size (torch.Size([64, 1])).
This will likely lead to incorrect results due to broadcasting. Please ensure
they have the same size.
  return F.mse_loss(input, target, reduction=self.reduction)
C:\Users\danie\anaconda3\envs\5_Programmcodes\lib\site-
packages\torch\nn\modules\loss.py:520: UserWarning: Using a target size
(torch.Size([37])) that is different to the input size (torch.Size([37, 1])).
This will likely lead to incorrect results due to broadcasting. Please ensure
they have the same size.
  return F.mse_loss(input, target, reduction=self.reduction)
Mean epoch loss:  523772197.6296296
-----
Epoch 1:

Mean epoch loss:  282913998.8148148
-----
Epoch 2:

Mean epoch loss:  38439752.24074074
-----
Epoch 3:

Mean epoch loss:  59739114.03703704
-----
Epoch 4:

Mean epoch loss:  45348350.96296296
-----
Epoch 5:

Mean epoch loss:  49225874.31481481
-----
Epoch 6:

Mean epoch loss:  49009096.88888889
-----
Epoch 7:

Mean epoch loss:  54060480.37037037
-----
Epoch 8:

Mean epoch loss:  57554486.09259259
-----
Epoch 9:
```

```
Mean epoch loss:  47773865.574074075
-----
Epoch 10:

Mean epoch loss:  59659683.85185185
-----
Epoch 11:

Mean epoch loss:  43242390.59259259
-----
Epoch 12:

Mean epoch loss:  51224792.777777776
-----
Epoch 13:

Mean epoch loss:  140062303.7962963
-----
Epoch 14:

Mean epoch loss:  119473168.12905093
-----
Epoch 15:

Mean epoch loss:  278949527.5341435
-----
Epoch 16:

Mean epoch loss:  132741917.5
-----
Epoch 17:

Mean epoch loss:  337701758.3888889
-----
Epoch 18:

Mean epoch loss:  212891950.43344906
-----
Epoch 19:

Mean epoch loss:  47080417.68518519
-----
Epoch 20:

Mean epoch loss:  47027748.18518519
-----
Epoch 21:
```

```
Mean epoch loss:  41032932.333333336
-----
Epoch 22:

Mean epoch loss:  47980138.643518515
-----
Epoch 23:

Mean epoch loss:  53961567.23611111
-----
Epoch 24:

Mean epoch loss:  52091076.11111111
-----
Epoch 25:

Mean epoch loss:  110385619.2951389
-----
Epoch 26:

Mean epoch loss:  47365030.89814815
-----
Epoch 27:

Mean epoch loss:  82354042.53703703
-----
Epoch 28:

Mean epoch loss:  38472842.981481485
-----
Epoch 29:

Mean epoch loss:  38817613.11111111
-----
Epoch 30:

Mean epoch loss:  48085134.71759259
-----
Epoch 31:

Mean epoch loss:  45718274.87962963
-----
Epoch 32:

Mean epoch loss:  51692658.675925925
-----
Epoch 33:
```

```
Mean epoch loss:  131337221.29398148
-----
Epoch 34:

Mean epoch loss:  263535167.16666666
-----
Epoch 35:

Mean epoch loss:  260275488.33333334
-----
Epoch 36:

Mean epoch loss:  214800105.16666666
-----
Epoch 37:

Mean epoch loss:  42754645.68518519
-----
Epoch 38:

Mean epoch loss:  53415813.7037037
-----
Epoch 39:

Mean epoch loss:  45768074.722222224
-----
Epoch 40:

Mean epoch loss:  43013134.49768519
-----
Epoch 41:

Mean epoch loss:  43785215.0
-----
Epoch 42:

Mean epoch loss:  43639336.18981481
-----
Epoch 43:

Mean epoch loss:  48068081.06365741
-----
Epoch 44:

Mean epoch loss:  55621350.027777776
-----
Epoch 45:
```

```
Mean epoch loss:  49178655.61111111
-----
Epoch 46:

Mean epoch loss:  51898537.68055555
-----
Epoch 47:

Mean epoch loss:  57719540.94212963
-----
Epoch 48:

Mean epoch loss:  52906026.625
-----
Epoch 49:

Mean epoch loss:  54789118.4212963
-----
Epoch 50:

Mean epoch loss:  60166647.53703704
-----
Epoch 51:

Mean epoch loss:  55539466.26388889
-----
Epoch 52:

Mean epoch loss:  47309640.96296296
-----
Epoch 53:

Mean epoch loss:  49250657.55555555
-----
Epoch 54:

Mean epoch loss:  55078279.99074074
-----
Epoch 55:

Mean epoch loss:  42519164.90740741
-----
Epoch 56:

Mean epoch loss:  55259232.25925926
-----
Epoch 57:
```

```
Mean epoch loss:  59128620.49537037
-----
Epoch 58:

Mean epoch loss:  56831791.511574075
-----
Epoch 59:

Mean epoch loss:  55825596.62615741
-----
Epoch 60:

Mean epoch loss:  56299818.23611111
-----
Epoch 61:

Mean epoch loss:  50384801.208333336
-----
Epoch 62:

Mean epoch loss:  54554048.05787037
-----
Epoch 63:

Mean epoch loss:  52667812.949074075
-----
Epoch 64:

Mean epoch loss:  50952562.03240741
-----
Epoch 65:

Mean epoch loss:  48073648.37037037
-----
Epoch 66:

Mean epoch loss:  54367719.3287037
-----
Epoch 67:

Mean epoch loss:  55642382.6712963
-----
Epoch 68:

Mean epoch loss:  61943302.17939815
-----
Epoch 69:
```

```
Mean epoch loss:  50890193.800925925
-----
Epoch 70:

Mean epoch loss:  48699131.67361111
-----
Epoch 71:

Mean epoch loss:  45046158.12037037
-----
Epoch 72:

Mean epoch loss:  50429693.083333336
-----
Epoch 73:

Mean epoch loss:  49144770.10185185
-----
Epoch 74:

Mean epoch loss:  54870960.69675926
-----
Epoch 75:

Mean epoch loss:  53344347.0462963
-----
Epoch 76:

Mean epoch loss:  52537399.61111111
-----
Epoch 77:

Mean epoch loss:  48240400.25
-----
Epoch 78:

Mean epoch loss:  47690476.9212963
-----
Epoch 79:

Mean epoch loss:  52280680.6712963
-----
Epoch 80:

Mean epoch loss:  54222903.04050926
-----
Epoch 81:
```

```
Mean epoch loss:  51545158.03240741
-----
Epoch 82:

Mean epoch loss:  49504098.16435185
-----
Epoch 83:

Mean epoch loss:  50323647.75
-----
Epoch 84:

Mean epoch loss:  51994863.92824074
-----
Epoch 85:

Mean epoch loss:  52226438.574074075
-----
Epoch 86:

Mean epoch loss:  51377691.0
-----
Epoch 87:

Mean epoch loss:  44964569.56481481
-----
Epoch 88:

Mean epoch loss:  54042575.69675926
-----
Epoch 89:

Mean epoch loss:  47832179.034722224
-----
Epoch 90:

Mean epoch loss:  48578020.7662037
-----
Epoch 91:

Mean epoch loss:  48633812.03356481
-----
Epoch 92:

Mean epoch loss:  43199555.832175925
-----
Epoch 93:
```

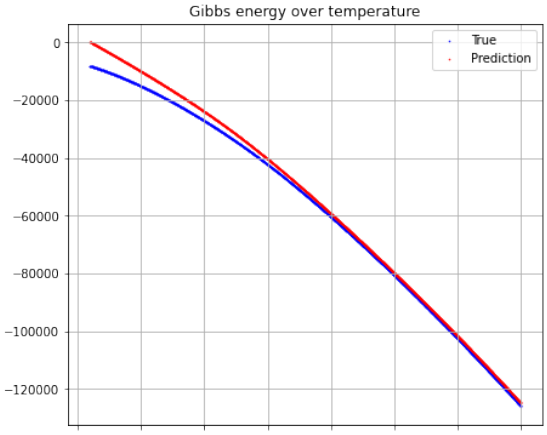
```
Mean epoch loss:  26484176.841435187
-----
Epoch 94:

Mean epoch loss:  15151356.494791666
-----
Epoch 95:

Mean epoch loss:  15178176.835648147
-----
Epoch 96:

Mean epoch loss:  10504396.436921297
-----
Epoch 97:

Mean epoch loss:  10930463.606481481
-----
Epoch 98:

Mean epoch loss:  14576867.394675925
-----
Epoch 99:

Mean epoch loss:  12935513.086805556
```

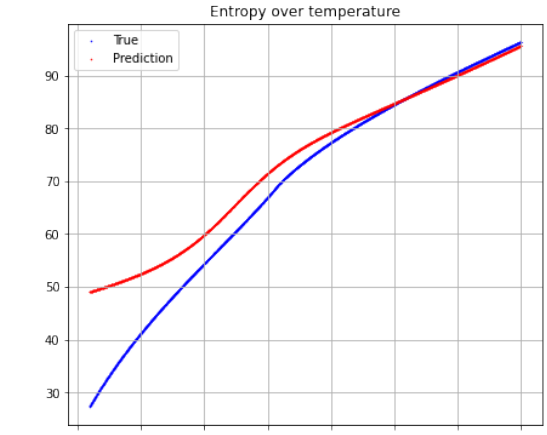
[8]:
```
print('theta_E: ', net.sub_net_1.act_1.theta_E)
print('E0: ', net.sub_net_1.act_1.E0)
print('a: ', net.sub_net_1.act_1.a)
print('b: ', net.sub_net_1.act_1.b)
```

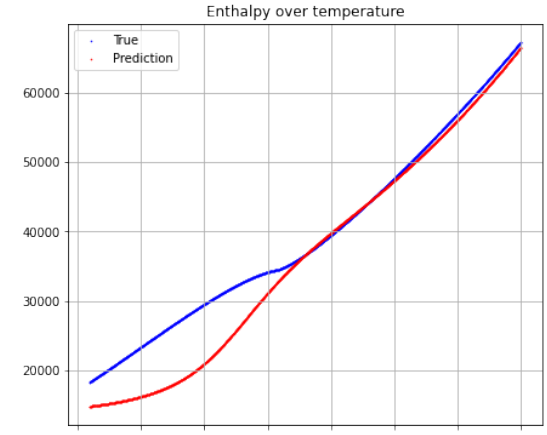
```
theta_E:  Parameter containing:
tensor(-1., requires_grad=True)
E0:  Parameter containing:
tensor(-18888.2500, requires_grad=True)
a:  Parameter containing:
tensor(-40.1167, requires_grad=True)
b:  Parameter containing:
tensor(0.5357, requires_grad=True)
```

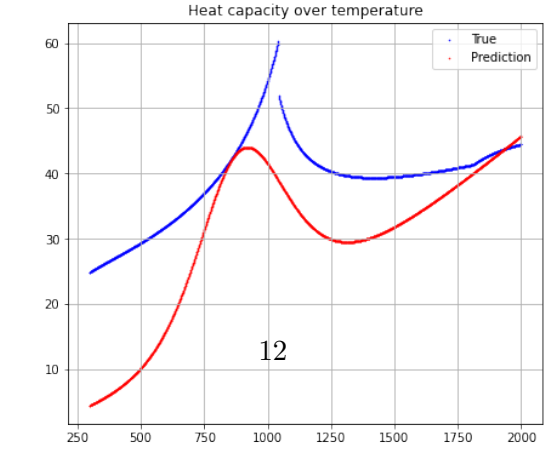
[6]:
```
ph = PlotHandler('Laenge')

ph.properties_temp(net, element, phase, scaling=False)
```

```
Fe successfully selected!
```


Gibbs energy over temperature
True
Prediction
0
−20000
−40000
−60000
−80000
−100000
−120000


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


Enthalpy over temperature
True
Prediction
60000
50000
40000
30000
20000


Heat capacity over temperature
True
Prediction
60
50
40
30
20
10
250
500
750
1000
1250
1500
1750
2000

[7]:
```
#torch.save(net, 'LaengeNet/Models/model_12_01_22_1535')
```

[ ]:
```

```