CSP Ex06

To run the code: install pytorch and pytorch-lightning then run the jupyter notebook. NOTE: If no gpu available set gpu=0.

Task1: See folder “data/”. It is pretty basic and works pretty well. I used L=32 and generated for each 40 temperature 4000 samples and then split in train & validation in 50-50.

Task2: I used pytorch-lightning because it is such a clean template of pytorch and very practical. Highly recommended if not known to any DL libraries. Pytorch is also the language I learned last semester at the lecture “Deep Learning”, hence I think I went a little bit overkill in this exercise xD.

Task3: My NN is a basic CNN of the form:

```
Classifier(
  (model): Sequential(
    (0): Conv2d(1, 6, kernel_size=(5, 5), stride=(1, 1))
    (1): ReLU()
    (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
    (3): Conv2d(6, 16, kernel_size=(5, 5), stride=(1, 1))
    (4): ReLU()
    (5): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
    (6): Flatten(start_dim=1, end_dim=-1)
    (7): Linear(in_features=400, out_features=120, bias=True)
    (8): ReLU()
    (9): Linear(in_features=120, out_features=84, bias=True)
    (10): ReLU()
    (11): Linear(in_features=84, out_features=1, bias=True)
    (12): Sigmoid()
  )
  (loss_fct): BCELoss()
)
```

I used only one single output with a Sigmoid activation function. This is much cleaner and with this you can use the binary cross entropy loss function.

class0: 0 if T < T_c
class1: 1 if T > T_c

Task4: My binary classifier trained for like 50 epochs with Adam(lr=1e-4) and at 39 epochs it got its best result of BACC = 91.6%. I didn’t neglect lattices close to the critical temperature to challenge my network. Also there are 72.5% of T > T_c data points and 27.5% of T < T_c due to linspace sampling in range (0.0, 8.0), where T_c = 2.27. Hence, the classifier is clearly more confident in classifying the samples as 0. Batch_size is set to 2048, because the bigger the better. My GPU (gtx 980) could have handled even more but I kept it at this size.

```
Test Error:
 Accuracy: 98.4%, Avg loss: 0.038395

 Balanced Accuracy:  0.9164900157868089

 Classification report:
              precision    recall  f1-score   support

         0.0       0.98      1.00      0.99     72323
         1.0       1.00      0.83      0.91      7677

    accuracy                           0.98     80000
   macro avg       0.99      0.92      0.95     80000
weighted avg       0.98      0.98      0.98     80000
```