Astronomy CNN with PyTorch

Axel Pontén

March 27, 2025

Code is found at `https://github.com/axel-ponten/advanced-deep-learning-VT25/tree/main`

I implemented a CNN with two convolutional layers and two fully connected layers. I split 80/20 train/test. A few example spectra are shown in Fig. 1.

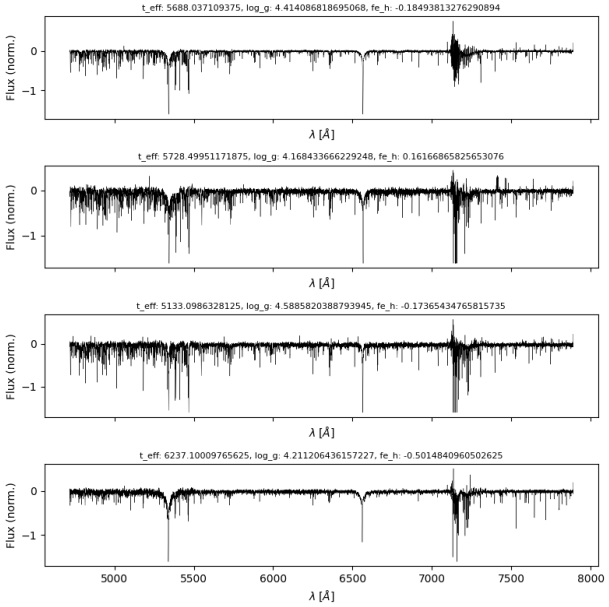

Figure 1: Example spectra.

The input label distribution is shown in Fig. 2.

At first I didn't normalize target when training. I used MSE loss, but I have three labels which all have very different scale. t_eff is on the scale of 5000 and fe_h is on the scale of 0.5. Physically I'm interested in relative differences for the labels, but the network will react harder to differences in

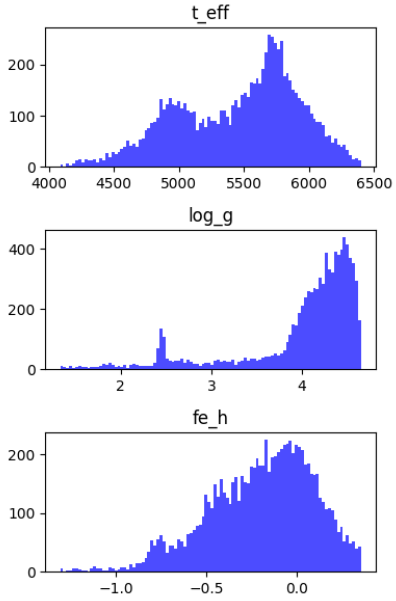

Figure 2: Label distribution.

t_eff than in fe_h due to the absolute difference. For improvement I could normalize the labels to have the same scale during training, and the reverse the transformation when using the network.

As you can see in Fig. 3, only t_eff is predicted well. The other two labels are not predicted well. After normalizing the labels to the label mean and std before training I get much better prediction as seen in Fig. 4.

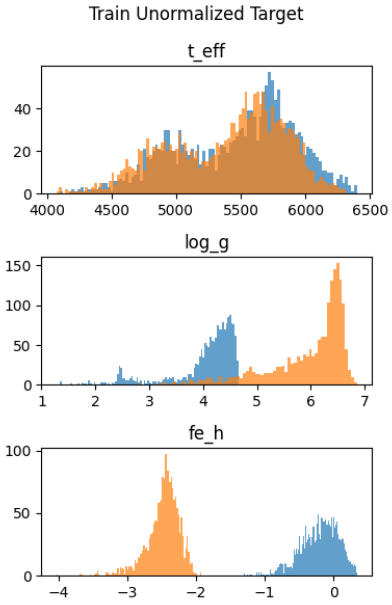

Figure 3: Prediction without normalizing labels.

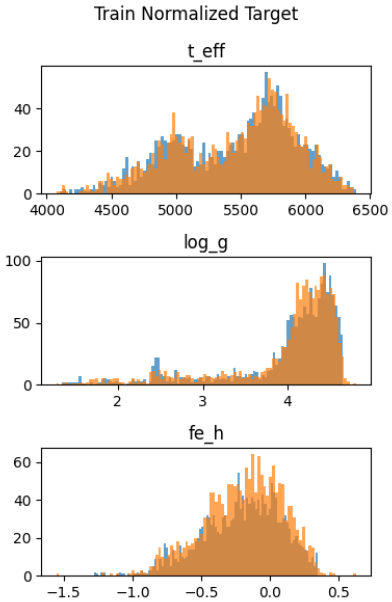

Figure 4: Prediction after normalizing labels and training.