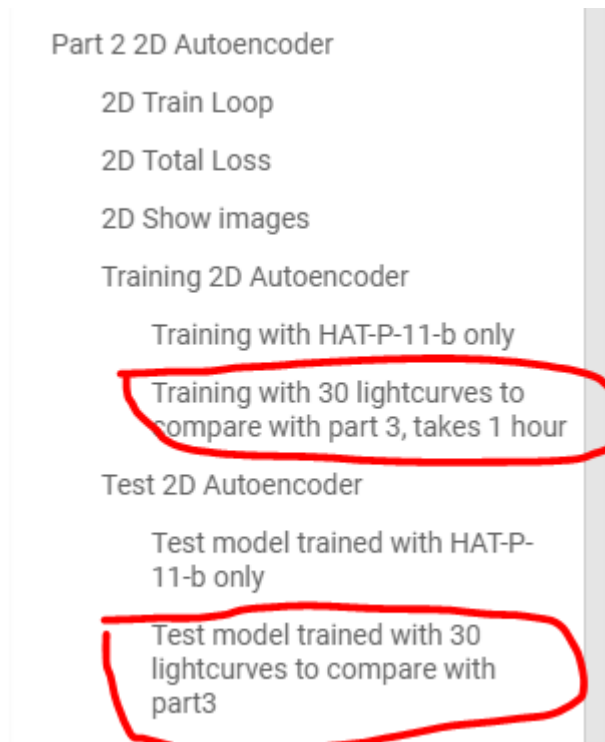


## COSC440 Project README

1. Before running 440\_project\_part\_2&3\_yzh417.ipynb, if want a quick run of 10 min, please comment out codes in the following 2 sections highlighted in red below:

- Training with 30 lightcurves to compare with part 3, takes 1 hour
- Test model trained with 30 lightcurves to compare with part3

Then press Ctrl+F9 to run the whole notebook



2. The example test images shown in section “Test model trained with HAT-P-11-b only” might look the same but they do have a small difference of approx 0.1 between each other. If want to see whether the part2 Autoencoder is actually doing something, uncomment the red highlighted code shown in the screenshot below:

### ▼ Training with HAT-P-11-b only

```
✓ [48] EPOCHES = 10
1m BATCH_SIZE = 30

#download training data HATP-11
tpf = lk.search_targetpixelfile('HAT-P-11', sector=14, cadence='short').download()

#init model
model_part2 = AutoEncoder()
optimizer = tf.keras.optimizers.Adam(learning_rate=0.01)

#prepare training data
flux_images = np.nan_to_num(tpf.hdu[1].data["FLUX"], nan=0)
flux_images = tf.random.shuffle(flux_images)
train_images = flux_images[:int(flux_images.shape[0] * 0.9)]
test_images = flux_images[int(flux_images.shape[0] * 0.9):]

#scale image to 0 and 1
test_images = test_images / np.max(np.abs(flux_images))
train_images = train_images / np.max(np.abs(flux_images))

#show exmple images, note the differences are small
example_images = test_images[:EXAMPLES]
print("Difference between test images shown", np.sum(np.abs(test_images[2]-test_images[1])))

# uncomment below to see if the network is doing sth
# for noise_function in [random_noise, no_noise]:
for noise_function in [no_noise]:
    for i in range(EPOCHES):
        for j in range(0, len(train_images), BATCH_SIZE):
            train(model_part2, optimizer, train_images[j:j+BATCH_SIZE], noise_function)

    print("Epoch: ", i)
    sum_loss = total_loss(model_part2, test_images, noise_function)
    print("Total Loss: {0}".format(sum_loss))
    showImages(model_part2, noise_function, example_images)
```

3. Please note both part2 and part3 Autoencoders contain functions of the same name 'train' and 'total\_loss', so please be careful running the code if cross checking different parts.

4. The code could also be accessed here:

[https://colab.research.google.com/drive/1AseY\\_QzJQwLiE1yOoNNWsP-1oZb0p26x?usp=sharing](https://colab.research.google.com/drive/1AseY_QzJQwLiE1yOoNNWsP-1oZb0p26x?usp=sharing)

Student ID: 97245310

Name: Yuezhang Zhu