Questions and Answers for Sharp Transition Analysis.

**This document contains a brief summary on the analysis of the sharp transition. And following the summary, there are several Q&A that will further clarify the details of parts of the summary.**

**A brief summary on the sharp transition analysis:**

For every HARP region with flares, we use its first ever M or X flare and its first ever B flare, if any, to construct a first flare dataset. In detail, for each of these flares, we first use the GOES dataset to pinpoint the peak time of the flare, and then take the time that is 18 hours ahead of the peak time as the starting point, the time that is 12 hours ahead of the peak time as the endpoint. All 20 SHARP parameters during the starting point and the endpoint are the corresponding timeseries data for predicting the flare. The prediction data also include the timeseries of the derivative of the 20 SHARP parameter. So in total there are 40 timeseries data. The M/X first flares are labelled as 1, and the B flares are labelled as 0. The data is of 12-min cadence. The model used to process the 20 SHARP parameters’ timeseries is the LSTM model. With timeseries input, the LSTM model will finally give a scalar output between 0 and 1, which means that the LSTM model is basically a binary classification model.

After the model training process, we go back to the full SHARP parameter history of each of the M/X first flare in the whole dataset, and plot the full predicted score path for this SHARP parameter history prior to the flare peak time. And detect if the predicted score has ever reached a high enough prediction score, and if so, prior to the high prediction score time, is there any time that the prediction score is very low. If so, we define that the history of this flare has experienced a “sharp transition”. We have finally found about 23 M/X flares that have such a sharp transition prior to the flare peak time.

**Q: Summary statistics for the first flare dataset?**

A: There are 787 flares in the first flare dataset, containing 167 M/X first flares and 620 B flares.

**Q: Is the first flare dataset the final dataset used for constructing the training set and testing set?**

A: No. For each flare in the first flare dataset, they have different length of video recording prior to the flare. Some of the first flares happen a couple of days after the video starts to record the HARP region while others may happen only a few hours after the video starts. And since the quality of the data is not very satisfactory at the beginning of some of the videos, namely there are a lot of missing SHARP parameters at the start of the video, **some first flares that happen very early in the video recording of the HARP region are dropped.** More concisely, only the flares that happen at least 48 hours after the video starts are kept for constructing the training set and testing set.

With such a standard, we have 463 flares remaining for constructing training and testing set. With 112 M/X flares and 351 B flares. Let’s call this dataset the **clean dataset.**

**Q: How do we obtain the timeseries of the 20 SHARP parameters? .**

A: For each flare in the clean dataset, there is at least 240 frames, namely 48 hours recording, of the SHARP parameters prior to the flare. The last frame in the clean dataset is the frame that is closest to the flare peak time, prior to the flare. And trace back in time from this frame, we take the time that is 18 hours ahead of the last frame as the starting point, 12 hours ahead of the last frame as the endpoint, and take all SHARP parameters data in between as the timeseries of the SHARP parameters for the flare. So we have a **12-hour prediction time**, and we utilize a 6-hour data for prediction.

**Q: How is the time derivative calculated for the timeseries of the 20 SHARP parameters?**

A: For the 12-hour prediction time, 6-hour data mentioned in the question above, each flare has exactly 30 frames of 20 SHARP parameters data, so the data shape is 30\*20. And for each SHARP feature, say TOTUSJH, we fit a spline curve over the 30 discrete data point, using *scipy.interpolate.UnivariateSpline* function. Without specifying the number knots, we first calculate the standard deviation of the 30 data points of TOTUSJH, denote it as sigma, and then limit the residual sum of square (RSS) to be no larger than 30\*(sigma)^2.

The RSS is the sum of the square of all data points’ residual. The residual of a data point is the difference between the data point’s TOTUSJH value and the fitted TOTUSJH value on the spline curve. So the constraint is to make sure the spline is smoothing the timeseries till that the average square of residual of the typical data point is no larger than the variance of the timeseries. The spline function in python would look for the smoothing solution to this automatically.

After the spline curve is fitted, we could calculate the time derivative at each point by taking the derivative of the smooth spline curve. All these steps are done individually for each flare. Now we have a 30\*40 timeseries data for each of the flare.

**Note that this is done prior to data standardization.**

**Q: How is the training and testing set being constructed?**

A: For the 463 flares in the clean dataset, we random split all flares into training set and testing set with ratio 2:1. The random state of the random splitting is 42. The function used for doing the train-test-split is the *train\_test\_split* function in *sklearn.model\_selection* module.

There are 310 training samples in the training set, including 75 M/X flares and 235 B flares. And 153 testing samples in the testing set, including 37 M/X flares and 116 B flares.

**Q: How is the dataset standardized? (Normalized so that each SHARP feature and its derivatives will have zero mean and variance 1.)**

A: After we obtain the training set and testing set, we take **only the training set for standardization.** Specifically, for all flares in the training set, each has a 30\*40 timeseries, namely 30 frames and 40 features (20 SHARP, 20 derivatives), and for each feature, across all 30 frames of all flares in the training set, we take the mean and standard deviation value of it. So for each feature, **we have a mean value and a standard deviation value only based on training set**. These values are stored in the working directory and would be useful for case study. And for all samples in the training and testing set, we subtract the mean and divide the standard deviation so that each feature in the training set has mean 0 and variance 1, but not necessarily so in the testing set. This standardization is done to avoid information leaking from the testing set during the model validation and evaluation.

**Q: What is the architecture of the LSTM binary classifier?**

A: The LSTM is a kind of Recurrent Neural Network (RNN). **Our model has two LSTM units.** And basically, when a (30 frames \* 40 features) timeseries sample is passed into the first LSTM unit, it outputs (30 frames \* X features) timeseries. In our case, we set X to be 50. **So basically LSTM encodes the timeseries** and projected it into a higher dimensional space to create more features. With this (30 frames \* 50 features) “encoded” timeseries, we randomly drop half of the features to 0 to avoid overfitting. After random drop, we apply the (30\*50) timeseries to the second LSTM unit. This time, we obtain a (30\*50) timeseries as output again. **But we only take the 50 features from the last frame,** do a random dropout to push half of the features to 0, and pass the 50 dimensional vector into a dense layer. The dense layer take the 50 features inside and output a value from 0 to 1 for prediction.

**Q: What is the training process, optimizer and training metrics?**

A: The LSTM model described above is **trained based on mini-batches.** Basically, we pass a batch of 10 flares into the LSTM model, calculate the average binary cross-entropy of the whole batch, and take the derivative of the average binary cross-entropy with respect to all parameters in the LSTM model which has millions of parameters, and update the parameters with a certain step size towards the direction of the gradient. The optimizer of the LSTM model is “adam” optimizer. For the whole training set, we have roughly 31 batches, so each round through the training set will update parameters 31 times. We loop over the training set 8 times, which is 8 epochs, in machine learning term. The accuracy of the model after 8 epochs is 82% on the training set, and 81% on the testing set. This means that 82% of the training flares are classified correctly, and 81% of the testing flares are predicted correctly. The prediction rule is that as long as the LSTM outputs anything above 0.5, the flare is classified as M/X, and B otherwise.

**Q: How do we predict the score for a whole history of a HARP region prior to its first B/M/X flare? In other words, how are the path of the prediction score getting plotted?**

A: For each first flare of a HARP region in the clean dataset, we pinpoint the time closest and prior to the flare peak time, and query a 48 hour SHARP parameter history from the original dataset. So for a video for AR11158, for example, if the video of AR11158 starts at 2011.02.10 10:00AM, and the first flare happened at 2011.02.13 17:05PM, then we only cares about the time from 2011.02.11 17:00PM to 2011.02.13 17:00PM.

And then starting from the first frame in this 48 hour window, w**e collect SHARP parameters from 30 consecutive frames.** And with this (30 frames\*20 features) timeseries, we fit the spline curve, calculate the derivative as specified above, and get a (30 frames\*40 features) timeseries, **and then subtract the mean and divide the standard deviation of the training set for LSTM model fitting**. And pass the (30\*40) standardized timeseries into the previously trained model and get a prediction score.

This prediction score is for the first frame of the 48 hour window. We now slide the window 12 minutes later, so we start from the second frame and collect 30 frames of SHARP parameters and do the same thing. And we get the second frame’s prediction score. These recursive steps can be down until the sliding window can no longer slide forward in time.

**Note that we link the prediction score of a sliding window to the time of the start of the window, not the end of the window.** Whenever we say a prediction score for a time, we are saying the prediction score for the sliding window starting from that time.

To make the plot, we plot each time’s prediction score, which is forecasting the flare probability in 12 hours, and also plot the real-time TOTUSJH and SAVNCPP for that time.

**Q: How do we define sharp transition?**

A: **For each M/X first flare**, we plot the path of prediction curve as described above. And then we find the time, if any, where the prediction score exceeds **0.7 and persist for at least 24 minutes**, which is a sign of strong prediction. We define this time as the **time after sharp transition**. And then starting from this time, we move backward in time to the past, 12 minutes at a step, and stop whenever we find the time that the prediction score is below **0.3 and persist for at least 24 minutes**. We define this time as **the time before the sharp transition**.

Note that we only look at M/X first flare. And we only care about the level of the prediction score reaching a certain upper bound or lower bound. We do not distinguish between cases where some prediction score paths rise much more rapidly than others. In other words, our transition is sharp only in the sense of the magnitude of the prediction score, not in the rate of the change of the prediction score. In some sense, some transitions are not literally “sharp”, but the score transitions do occur.