Forecasting Seasonal Drought Severity in The Western US

Madison Ingling (mmi2114)
EAEE4000
Machine Learning for Environmental Engineering and Science
December 23, 2021
Professor Gentine
Code: https://github.com/madisoningling/MLproject

Abstract

The aim of this study was to build a machine learning model that would be able to forecast the severity of seasonal drought in a particular region in the Western US based on snowpack conditions in the Colorado River Basin. A long short-term memory network model was formulated to capture the crucial temporal information, however it was determined that the model's accuracy was heavily limited by the volume of data used. Sherpa hyperparameter optimization and regularization were used to improve the model's accuracy and showed some signs of success. Further study should be conducted to determine if the model's mean absolute value can be improved beyond 0.20 by giving the model more data to train with.

Introduction

With the consequences of climate change becoming more apparent in recent years, it becomes increasingly important to study how extreme weather events might be affected by the rapidly-changing climate system; drought is one of these events. As the second-most costly weather event[1], drought is especially significant due to its heavy influence on the agricultural sector. Severe drought conditions can lead to large losses in crop production, in addition to significantly increasing the risk of wildfires. The Western United States is a region that typically experiences drought conditions; in fact, more than 52 percent of the West's land area was classified to be experiencing "extreme or exceptional drought" as of October 12, 2021[2]. With this in mind, it should be noted that several western states, including Arizona, California, Colorado, Nevada, New Mexico, Utah, and Wyoming, all heavily depend on the Colorado River Basin as a water source. The majority of this basin's water comes from the region's snowpack, and so observing the conditions of the snowpack certainly could be a useful indicator of water availability for the surrounding region. Because of this, the transition from the snow accumulation season to melt season holds key information. The beginning of the melt season typically occurs in late spring for the Colorado region, allowing meltwater to slowly travel through the basin and be made available for human/land consumption during the summer months[3]. However, with warmer winter temperatures induced by climate change, the melt season shifts and begins earlier, meaning that there is less water available in the latter half of the calendar year when it is most needed. The objective of this project is to explore this relationship further and use machine learning to determine if it is possible to quantitatively relate this snowpack information with the future drought conditions. It is important to highlight that the scope of this project focuses purely on seasonal drought conditions, and is not considering long-term drought.

Data

The region used to characterize the snowpack conditions was defined to be the Colorado River Basin, while the region considered for drought severity quantification consists of the seven following states: Arizona, California, Colorado, Nevada, New Mexico, Utah, and Wyoming. The

input data used to describe snowpack conditions in the Colorado River Basin was collected from the National Water and Climate Center's (NWCC) automated Snow Telemetry (SNOTEL) Network[4]. SNOTEL sites use low-maintenance measuring devices and sensors placed in high-elevation locations throughout the Western US to collect various types of data on the surrounding environment, mainly focusing on the conditions of the present snowpack. The data used from the SNOTEL sites include that of temperature, precipitation increment (snow-adjusted), snow depth, and snow water equivalent (SWE). In addition to these variables, streamflow was also chosen as a value of interest. However, SNOTEL sites do not measure streamflow, and so the streamflow values associated with the SNOTEL data was retrieved from the USGS National Water Information System[5]. The USGS streamflow sites were paired with the SNOTEL sites based on their proximity to each other; proximity was determined using ArcGIS. The snowpack data was collected on a monthly basis, specifically around the end of accumulation season; the values were retrieved for the months of January, February, March, April, and May. The region of interest is shown in Figure 1, where the red sites represent SNOTEL sites and the green sites represent USGS streamflow sites.

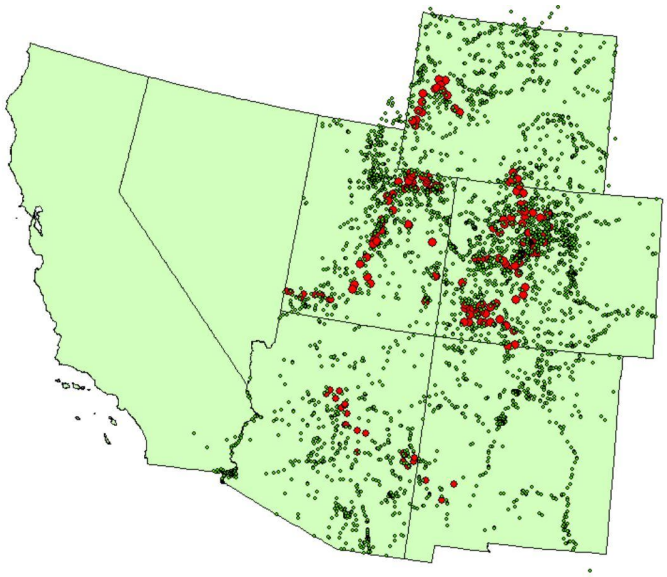


Figure 1: A map of SNOTEL and USGS streamflow sites in the Colorado River Basin.

Although radiation was considered as another input variable, the period of record for NASA's net radiation data only went as far back as 2006, which would have eliminated many data points that were used for this study. Information concerning drought conditions was collected from the U.S. Drought Monitor (USDM)[6]. The severity of drought was characterized by the fractional area of the region that experiences Level D2 (defined as severe drought) during the month of August. A summary of the variables used for this study can be seen in Table 1.

Variable	Name	Units	Source
site	SNOTEL site name	n/a	NWCC
year	Year	n/a	NWCC
month	Month	n/a	NWCC
airT	Average monthly air temperature	℉	NWCC
precip	Monthly precipitation increment (snow-adjusted)	inches	NWCC
snowDepth	Average monthly snow depth	inches	NWCC
SWE	Average monthly snow water equivalent	inches	NWCC
streamFlow	Average monthly streamflow	ft/s	USGS
D2Area	Fractional area of region experiencing D2 Level drought conditions	n/a	USDM

Table 1: A table summarizing the key variables used in this study.

Methodology

Formulation

Since the objective of this study is to use information collected during the first half of the year to try to forecast future drought conditions in the latter half of the year, it is evident that a model that works well with temporal information would be the best choice; in this case, a long short-term memory (LSTM) network was chosen. It was determined that the data for a batch of SNOTEL sites would be used as a single sample, rather than using a single site, as an attempt to better represent the overall conditions of the basin; the sites were randomly ordered according to their respective year and the size of the batch was first arbitrarily chosen to be four, however the model was later used with a batch size of one. Since it was not guaranteed that the total number of sites for each year would divide evenly by batch size, padding was implemented to keep the batches uniform. More specifically, pre-padding was used because it has been shown that LSTM models work better with pre-padding rather than post-padding[7]. After padding was added to the batches, the data was manually reshaped so that the data within each batch was organized by timestep; Figure 2 helps visualize this reconfiguration.

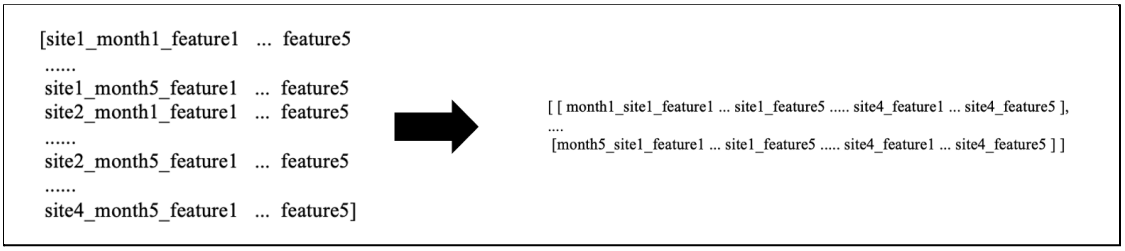


Figure 2: A visualization of the reconfiguration of the data before reshaping it for LSTM input.

The reconfigured input data was then reshaped to match the input shape that is expected for LSTM models, which is (number of samples, timesteps, features at each timestep)[8]. The associated output vectors all consist of a single value: the fraction of area that experiences severe drought. The data was split into training and testing data using a 0.7:0.3 split, and the training data was normalized before being passed into the model. When defining the basic architecture of the model, the hyperparameter values were first chosen arbitrarily with the use of early stopping as the only regularization technique; this was done to observe the model's performance with no sort of optimization. Following this, Sherpa hyperparameter optimization was conducted to observe how the basic model might be improved upon. Since the dataset nor the number of input variables is large, a principal component analysis was not conducted. It should be noted that the optimizer used for all model variations was Adam.

Results

Model A

For the first round of analysis, the batch size was assumed to be four; with padding, the total number of data points was 302. As previously discussed, the hyperparameters were chosen arbitrarily for the model variation; the values are summarized in Table 2.

Hyperparameter	Value
Number of LSTM layers	3
Number of nodes	32
Learning rate	0.001

Table 2: Hyperparamter values chosen for the first LSTM model variation.

Although the dropout technique was not used, early stopping was used for this model to help combat overfitting. Figure 3 shows the progress of the model's accuracy during training.

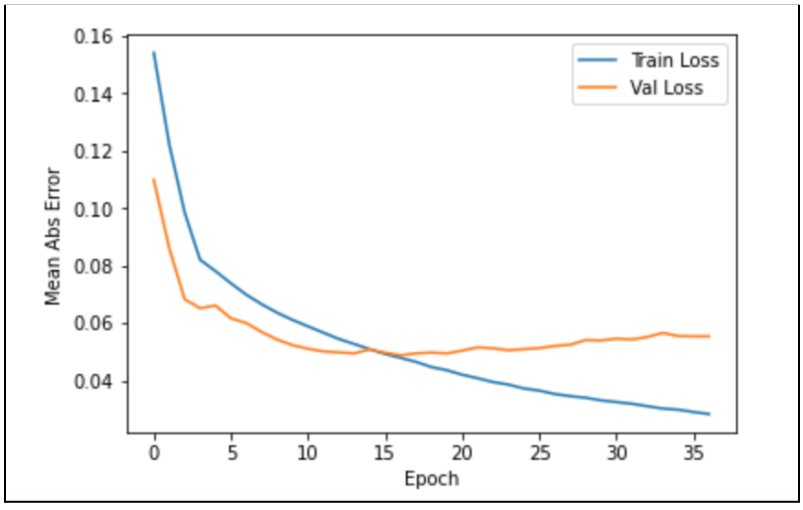


Figure 3: A line graph showing how well the model fits the training and validation set of data.

Although early stopping was implemented in this model, it is evident that overfitting begins before training is stopped; this is seen where the validation loss ceases to decrease alongside the training loss, around epoch 15. Unsurprisingly, the model's predictive power was found to be poor when evaluating it based on the test data. This is visualized in Figure 4, where the true output values of the test data are compared against the model's predicted values.

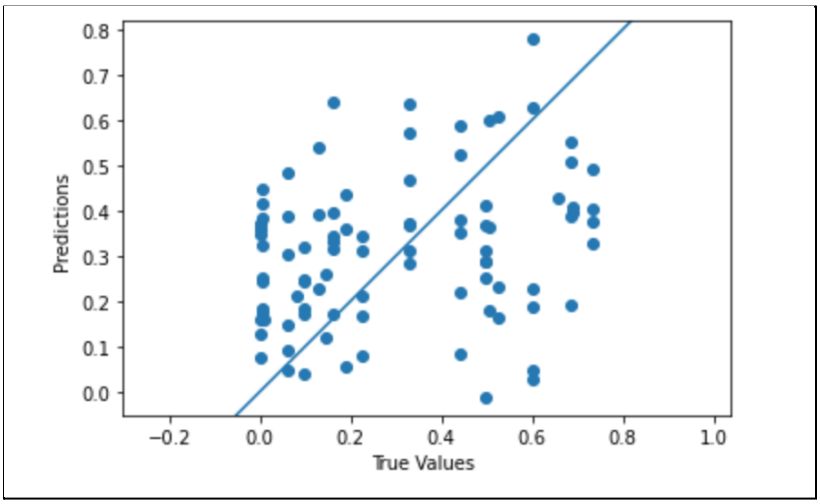


Figure 4: A scatter plot of true test values plotted against predicted values.

The mean absolute error of the predictions made for the test data was found to be 0.222, while correlation coefficient between the true and predicted values was determined to be -0.082. In the context of the study, this MAE value shows that the model predicts the fractional area of the region experiencing severe drought with an average error of 0.222. It is clear that this model is inadequate, and so the Sherpa optimization was utilized to determine if the model could be improved by fine-tuning the model's hyperparameters.

Sherpa's random search algorithm was used to optimize the following hyperparameters: dropout rate, learning rate, number of LSTM layers, and number of nodes in each layer. The dropout rate was defined as an ordinal parameter whose value can be 0, 0.1, 0.2, 0.3, 0.4, or 0.5. The learning rate was defined as a continuous variable that can range from 0.0005 to 0.001, with values being sampled uniformly on a log scale. The number of nodes per layer was constrained to 8, 16, 32, 64, or 128 as an ordinal parameter. The number of LSTM layers, a discrete hyperparameter, was allowed to range from 1 to 8 layers. The number of trials completed in this optimization analysis was chosen to be 500. The objective value was defined as the mean absolute error, while the mean squared error value was used for the loss. Early stopping was still used in this model. Figure 5 visualizes all of the trials conducted during the random search algorithm.

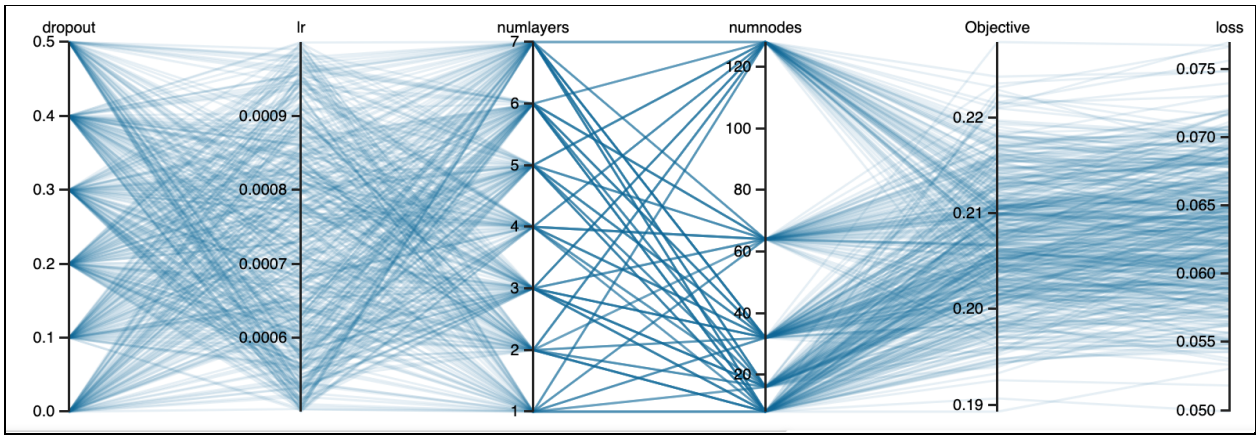


Figure 5: The results of Sherpa optimization analysis for the data using a batch size of 4 after 500 trials.

From just viewing Figure 5, it is difficult to identify optimal hyperparameter values, as there does not seem to be any one value (for any of the hyperparameters) that appears to be favored over the others. Table 3 shows the optimal values identified in the top 10 trials that had the lowest loss.

	TrialID	Dropout Rate	Learning Rate	Layers	Nodes	Objective	Loss
0	437	0.4	0.000781	2	8	0.191	0.0499
1	345	0.3	0.000679	2	8	0.193	0.0518
2	132	0.3	0.000932	7	8	0.189	0.0530
3	372	0.5	0.000826	4	16	0.196	0.0533
4	394	0.5	0.000541	2	8	0.199	0.0536
5	236	0.5	0.000603	3	8	0.199	0.0538
6	404	0.4	0.000687	4	8	0.196	0.0543
7	63	0.1	0.000640	2	8	0.195	0.0543
8	207	0.4	0.000831	3	16	0.198	0.0544
9	426	0.3	0.000624	7	8	0.196	0.0545

Table 3: A Table of optimal hyperparameters for the best-performing trials in the Sherpa random search analysis.

The optimal hyperparameter values for the next model were chosen based on the values in Table 3. The dropout rate, number of layers, and number of nodes were determined based on the frequency of values. If multiple values were tied for highest frequency, both values were tested in the model to determine which one gave better results. The learning rate was also chosen via trial and error, however instead of using the exact rates found in Table 3, the rate was rounded to the ten thousandths place to narrow down the options. With this logic, the model's optimal hyperparameters were determined, and can be seen in Table 4.

Hyperparameter	Value
Dropout rate	0.5
Learning rate	0.0008
Layers	2
Nodes	8

Table 4: A summary of optimal hyperparameters found using Sherpa's random search algorithm.

As depicted in Figure 6, it is clear to see that this new model trains much better than the previous model.

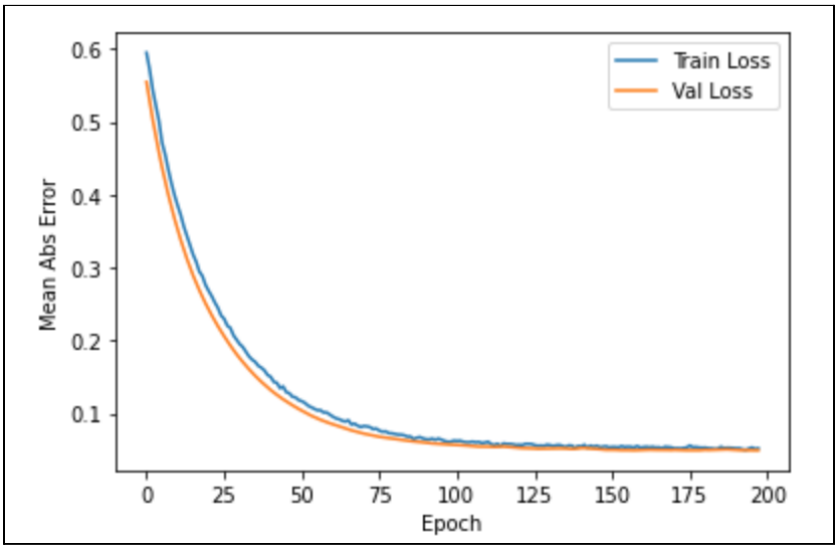
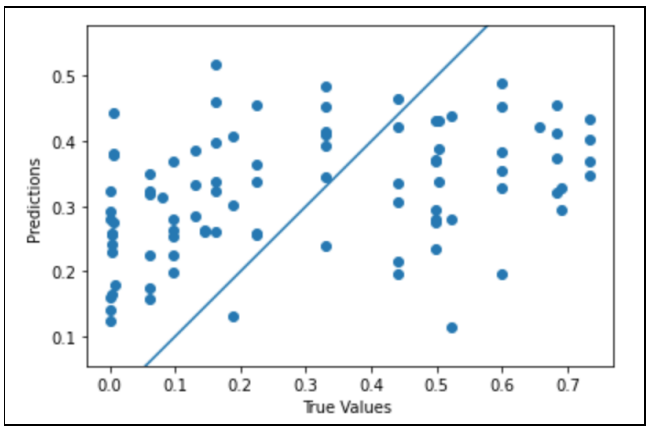


Figure 6: A line graph showing how well the model fits the training and validation set of data with the optimized hyperparameters.

There no longer appears to be any sign of overfitting, as the validation loss now continues to decrease with the training loss. The overall shape of both the training and validation losses also decrease much more smoothly.

When evaluating this new model on the test data, a mean absolute error of 0.203 was calculated. Although this is an improvement from the previous model, it is only an 8.6% decrease, which still is not ideal. Figures 7a and 7b below display the disparity between true test data values and the model's predicted values.



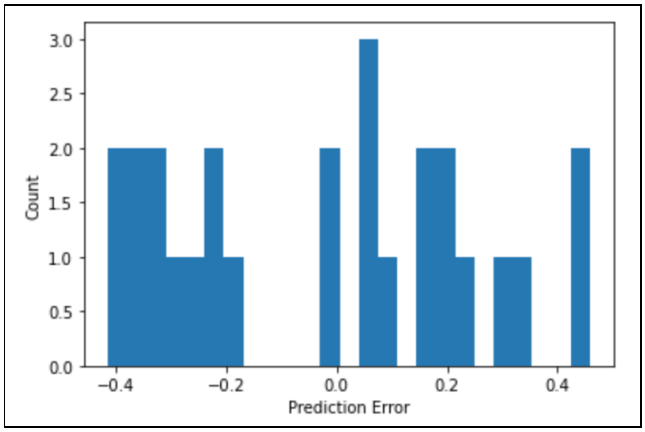


Figure 7(a): A scatter plot of true test values plotted against predicted values for the optimized model. Figure 7(b): A histogram showing the distribution of prediction error for the optimized model.

The R^2 value for the true and predicted output values was calculated to be 0.139. This value is significantly better than that of the original model, although it is still quite poor; the lack of correlation can be viewed in Figure 7a. Figure 7b does not show any obvious trend in prediction error, with most observations having a prediction error that is relatively far from zero.

It is more than likely that these poor results are due to the small sample size, as neural networks often require a large volume of data in order to build a model with decent predictive power. Since collecting more data is extremely time-consuming, a different approach was taken to attempt to address this issue; instead of using a batch size of 4, each site was used as a single sample, meaning that there would now be 1127 total data points available.

Model B

This new configuration of input data was first used in the original LSTM model that included no hyperparameter optimization; these arbitrary hyperparameter values are listed in Table 2. Figure 8 visualizes the training of the model, which is stopped early after 75 epochs. The graph shows signs of overfitting, similar to Figure 3, as the validation loss begins to rise above the decreasing training loss after around epoch 30.

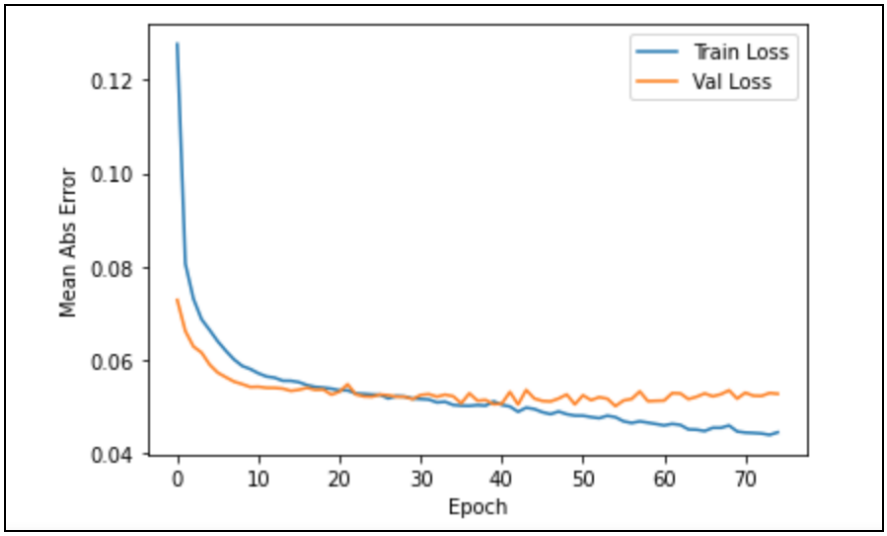
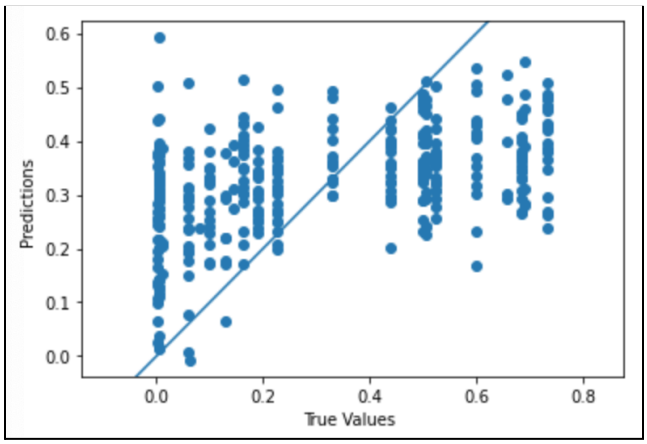


Figure 8 A line graph showing how well the model fits the training and validation set of data with batch size of 1.

The mean absolute error for the training data set was about 0.178, which is lower than that in the earlier models that used a batch size of 4, indicating that there may be potential for this model to be better, however it is still overfitted, so further optimization will be required. The mean absolute error for the test data was found to be 0.194, which is a 13.5% decrease from the very first model that was built. Figures 9a and 9b visualize the error of the predicted values for this model.



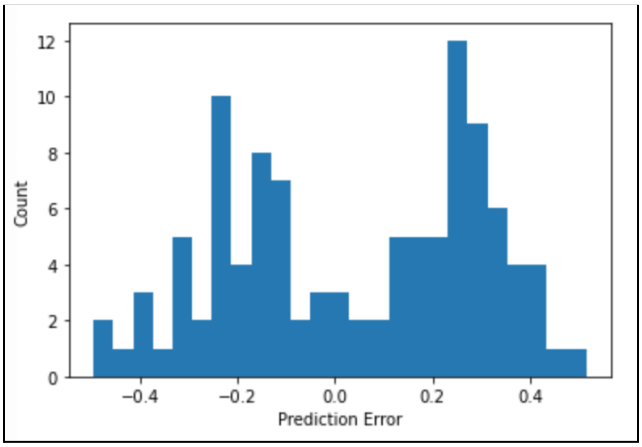


Figure 9(a): A scatter plot of true test values plotted against predicted values for the base model with no optimization. Figure 9(b): A histogram showing the distribution of prediction error for the model.

The R^2 value to describe the relationship between the true and predicted values was found to be 0.216, which is an improvement from the previous optimized model that used a batch size of 4. On the other hand, the model is still quite poor. The true and predicted values do not appear to be any more correlated than the previous models, indicating that the number of samples likely helped increase the correlation coefficient. Figure 9b shows that the distribution of error values

appears to be more bimodal, centered around values that are on either side of 0.0 prediction error, instead of the ideal unimodal shape that would be centered around the point of zero error.

Sherpa optimization was then conducted for this new model variation. Figure 10 displays the results of the random search.

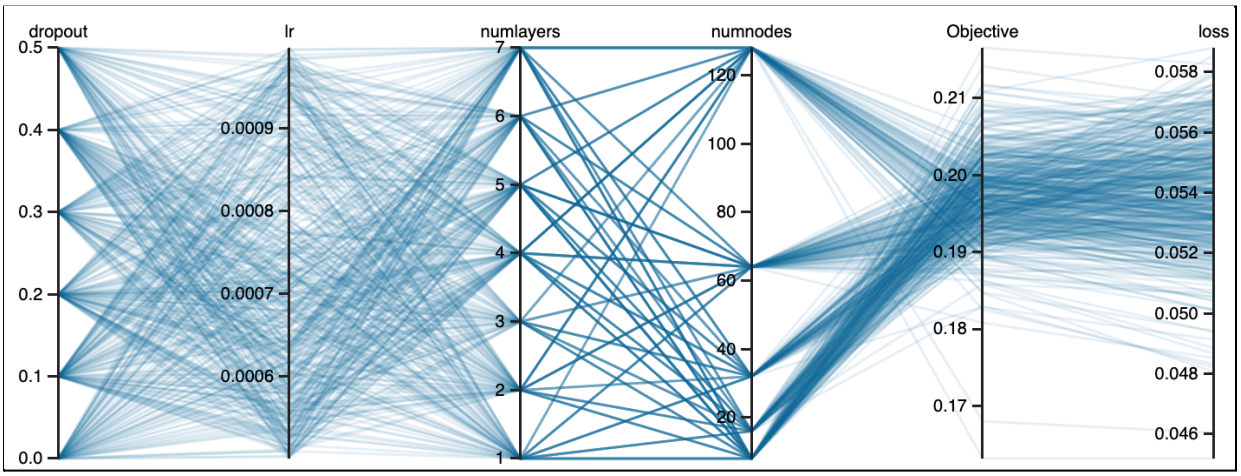


Figure 10: The results of Sherpa optimization analysis for the data using a batch size of 1 after 500 trials.

Similar to the previous Sherpa optimization, it is difficult to determine if there are obvious optimal hyperparameter values from this graph. This may be because it is difficult to optimize the model with a small data volume, or perhaps not enough trials were conducted to identify any favored values. However, running 500 trials was computationally expensive, and so the analysis was not repeated to determine if the quality of the results are limited by the number of trials. Table 5 summarizes the best results from this analysis.

	TrialID	Dropout Rate	Learning Rate	Layers	Nodes	Objective	Loss
0	486	0.2	0.00093	4	128	0.163	0.0452
1	346	0.5	0.00078	4	128	0.168	0.0460
2	291	0.5	0.00087	1	128	0.185	0.0481
3	385	0.4	0.00098	2	16	0.183	0.0482
4	136	0.1	0.00100	2	128	0.183	0.0483
5	357	0.4	0.00096	4	64	0.181	0.0485
6	405	0.5	0.00065	1	16	0.186	0.0487
7	450	0.5	0.00076	2	64	0.183	0.0489
8	463	0.5	0.00064	3	32	0.186	0.0491
9	236	0.3	0.00072	2	128	0.185	0.0494

Table 5: A table of the best results from the Sherpa random search analysis for the data using a batch size of 1.

The optimal hyperparameters were chosen based on these best results, and can be seen in Table 6.

Hyperparameter	Value
Dropout rate	0.5
Learning rate	0.001
Layers	2
Nodes	128

Table 6: A summary of optimal hyperparameters found based on Sherpa's random search algorithm for the data using a batch size of 1.

Although the new model would be expected to improve when using optimized hyperparameters, Figure 11 reveals that the model still was overfitting during training.

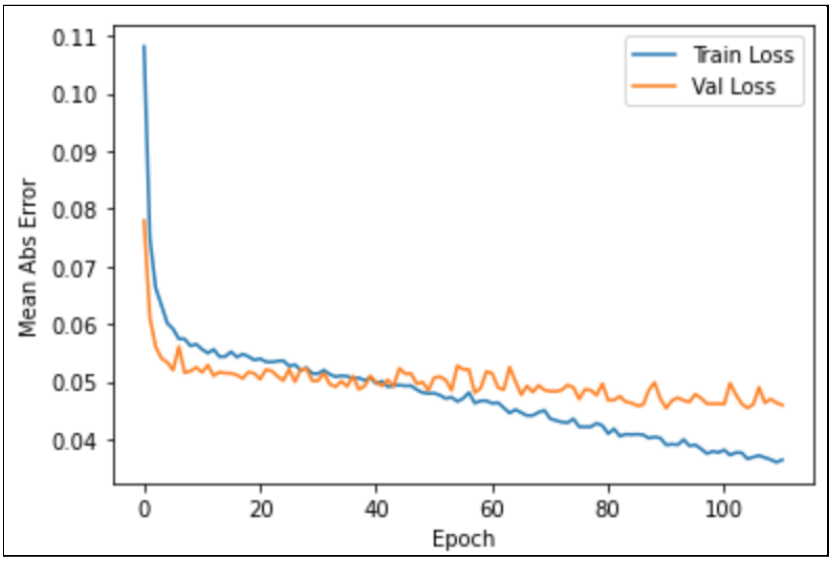


Figure 11: A line graph showing how well the model fits the training and validation set of data for the optimized model using batch size = 1.

The figure also shows a much noisier loss function, indicating that the gradient descent struggled to converge smoothly towards the global minimum. The model is associated with the lowest test mean absolute error seen thus far, a value of 0.175, in addition to the best R^2 value of 0.349. Despite this, the model is still insufficient based on the sole fact that it is overfitting because it would probably not predict well when given new data.

In order to combat this overfitting, an attempt to manually choose the hyperparameters was made. To address the noisy loss function, a smaller learning rate of 0.0003 was chosen,

while the other previously chosen values of hyperparameters were kept the same. Figure 12 shows an improvement in the model with no clear signs of overfitting as seen in Figure 11.

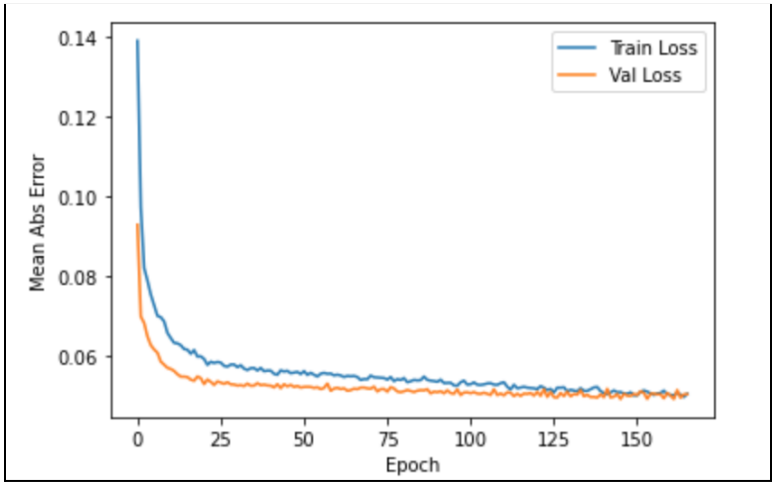
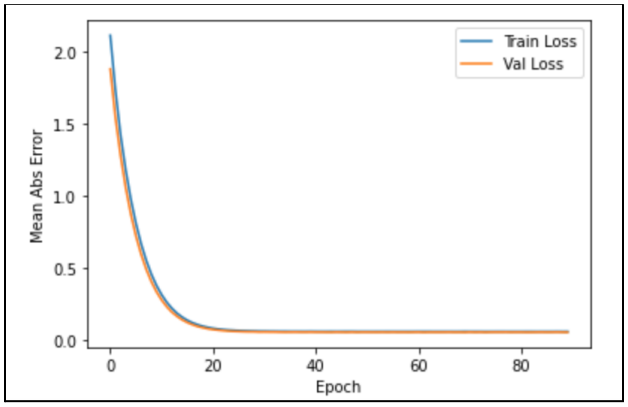


Figure 12: A line graph showing how well the model fits the training and validation set of data for the manually optimized model using batch size = 1.

The mean absolute error for the test data was found to be 0.199, while the R^2 value was 0.190. Although these values indicate that there is more error associated with this model, prediction accuracy must be sacrificed in order to avoid an overfitted model. While this model is an improvement from the previous one, it is worth noting that the updated model still has a noisy loss function. In order to address this, a different regularization technique, L2 Regularization, was used in place of using dropout layers; this regularizer was added to each LSTM layer. Figure 13a shows that this regularization technique helped improve the smoothness of the gradient descent since there is little to no visible noise in the loss function.



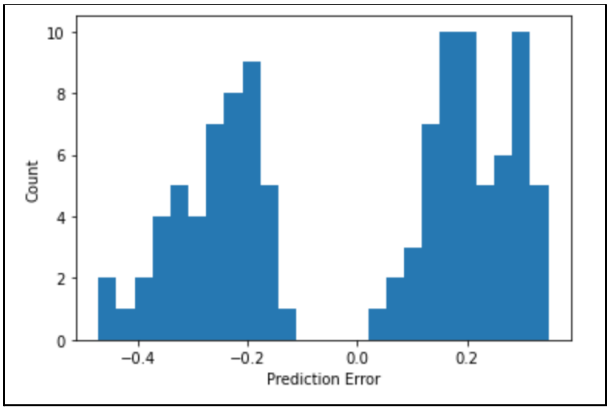


Figure 13(a): A scatter plot of true test values plotted against predicted values for the model with L2 regularization. Figure 13(b): A histogram showing the distribution of prediction error for the model.

On the other hand, the test data's mean absolute error increased to 0.218, while the R^2 value drastically dropped to 0.081. Figure 13b also shows that the distribution of prediction error moved further from the ideal value of zero; these results indicate that although the loss function converges more smoothly, the model's accuracy suffers.

Conclusions

While all of the LSTM model variations are quite poor in being able to accurately predict drought severity, it highlights how crucial a large data set is for LSTM modeling. It cannot be emphasized enough how more data would have likely improved the results. While an attempt was made to artificially increase the sample size by decreasing the batch size, this did not appear to improve model accuracy. Although more SNOTEL sites could have been included in the study, it is likely that the period of record is what limits the models' accuracy the most. It was easy to discern the data points that were associated with each year because they all have the same true value. Each year's set of samples could be identified by each vertical line in the true vs. prediction scatter plots; this is especially noticeable in Figure 9a. This highlights a key weakness in the data set. The period of record for the majority of SNOTEL sites barely cover the last 20 years, which severely limited the sample size. Although some SNOTEL sites had a longer historical record, nearly all of them were not able to measure snow depth until 2000, which decreased the potential sample size. Being able to include more data from different years would have allowed for more unique drought severity values, which likely would have improved the model since it would have been exposed to a larger variety of output values during training. Nevertheless, it was determined that the best model variation was Model A (batch size=4) after Sherpa optimization. This model showed the best improvement in accuracy after using the optimal hyperparameters. However, it is not advisable to use this model to predict drought severity, as the error is far too large for any reasonable prediction. Especially in the crucial context of drought severity, the predicted values should have minimal error so that the region of interest can prepare accordingly. It appears that all of the models' accuracy (MAE) remained around ~0.20, which is not particularly good, however it reveals that the model does have some potential for accurate prediction. Further study should certainly be conducted to determine if additional data can improve the model's accuracy, as forecasting drought in this region would undoubtedly be a powerful tool for the people that depend on the water from the Colorado River Basin.

Despite the inaccuracy of the model, knowledge was gained through the study's exploration of various optimization techniques. While the Sherpa hyperparameter optimization did not provide clear optimal results (likely due to data volume), this analysis did help improve Model A. The original model was overfitted, but was significantly improved after optimizing the hyperparameters and adding dropout as a regularization technique in addition to early stopping. Although the optimized hyperparameters for Model B could not resolve the overfitting issue, manual tweaking of the hyperparameters found a solution to avoid overfitting. The model's

training loss function was further improved upon after replacing the dropout layers with L2 regularization, although this decreased the model's accuracy. This exploration of various hyperparameter values and regularization techniques highlights the crucial refinement step in machine learning. Each and every decision made in building the architecture of a model can significantly impact the quality of the model, and so special care should be taken towards optimizing these choices.

References

[1] National Geographic Society. “Drought.” *National Geographic Society*, 9 Sept. 2019, https://www.nationalgeographic.org/encyclopedia/drought/.

[2] “Drought in the Western United States.” *USDA ERS - Drought in the Western United States*, 2021, https://www.ers.usda.gov/newsroom/trending-topics/drought-in-the-western-united-states/.

[3] “Increased Winter Snowmelt Threatens Western Water Resources.” *CU Boulder Today*, 8 Apr. 2021, https://www.colorado.edu/today/2021/04/05/increased-winter-snowmelt-threatens-western-water-resources.

[4] “Report Generator 2.0.” *Natural Resources Conservation Service National Water and Climate Center*, 2021, https://wcc.sc.egov.usda.gov/reportGenerator/.

[5] “USGS Water Data for the Nation.” *USGS*, 2020, https://waterdata.usgs.gov/nwis.

[6] “Historical Data and Conditions.” *Drought.gov*, National Integrated Drought Information System, 2021, https://www.drought.gov/historical-information?dataset=1&selectedDateUSDM=20101221&selectedDateSpi=19580601.

[7] Reddy, Dwarampudi, and N Reddy. *Effects of Padding on LSTMs and CNNS*. 18 Mar. 2019, https://arxiv.org/abs/1903.07288.

[8] Brownlee, Jason. “How to Reshape Input Data for Long Short-Term Memory Networks in Keras.” *Machine Learning Mastery*, 14 Aug. 2019, https://machinelearningmastery.com/reshape-input-data-long-short-term-memory-networks-keras/.