CS230

SalmonNet

Abdulwahab Omira, Ismael Castro, Christopher Shell
Department of Computer Science
Stanford University
{aomira,ismael01,cshell21}@stanford.edu

1 Introduction

For our project we are investigating if Deep Learning (DL) can be applied to improve salmon forecasting models. Salmon are an important ecological, cultural, and economic species. Salmon are considered to be a keystone species which means that they are the essential kog in their ecosystem (Kurlansky, 2020). As previously mentioned, salmon are an essential economic resource. Salmon fishing contributes over $688 million to the US economy (*American seafood industry steadily increases its footprint*, 2015). Due to its commercial significance, salmon managers have to release a yearly forecast of salmon returns to estimate the amount of salmon that can be harvested. This is where our project comes into the fold. We are trying to see if we can use DL to improve these salmon forecasts. The salmon forecast is a vital tool in monitoring salmon stocks for commercial, tribal, and recreational harvests (McCormick & Falcy, 2015). If less salmon return than the predicted amount, the fisheries may over harvest and not allow enough salmon to return to the spawning grounds. If more salmon return than the predicted amount, the fisheries may under harvest the resource which would could cost the local economy millions due to missed fish (McCormick & Falcy, 2015). Because of the significance of the forecast, resource managers have tried to improve the traditional forecasting methods, however, surprisingly to our team, DL remains a relatively unexplored method. For our research, we are going to explore if DL techniques applied to salmon forecasting can improve forecasting models. The input into our model is a time-series dataset of Chinook ("King") Salmon counts at Bonneville Dam on the Columbia River on daily and monthly time-steps. We then used Neural Networks (NN), Recurrent Neural Networks (RNN), Grated Recurrent Units (GRU), and Long-Short Term Memory (LSTM) models to predict the amount of salmon that would return to the dam each day or each month. For example, we use the past 180 days to predict the 181th day.

2 Related work

There has been very little work conducted on using ML to forecast salmon runs and even less work utilizing more modern algorithms. Zhou (2003) and McCormick & Falcy (2015) attempted to show the effectiveness of Artificial Neural Networks (ANNs) when compared to traditional forecasting methods which have relied heavily on standard regression (such as linear regression, sibiling regression, ridge regression, and lasso regression)(Burke et al. 2013; Hand & Haeseker, 2011). However, the ANNs were not able to demonstrate significant improvement over traditional forecasting methods. Xu et al (2020) and Hilborn et al (2020) were the next papers to conduct research on the potential of using machine learning to improve salmon forecasting. Yet, unlike McCormick & Falcy, both papers were able to demonstrate an improvement over traditional forecasting methods and recommend that machine learning techniques be incorporated into current models. In Hilborn et al., for example, the new models were able to improve forecasting on every river system modeled when compared to the traditional forecasts. Yet, no model was able to perform successfully across all river systems and age classes (Hilborn et al., 2020). Both papers primarily focused on using non-DL architectures such as Random Forest and Boosted Regression Trees although Hilborn attempted to

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

use RNNs but did not publish any information about their performance (Hilborn et al., 2020; Xu et al., 2020). Clearly, one of the biggest weaknesses of the current literature is the lack of research being done on using DL to predict salmon runs. Our project attempts to fill this gap in the research and tries to gain insight on whether DL, specifically recurrent neural networks, are a viable option for salmon forecasting.

3 Dataset and Features

For our salmon abundance data we are using Adult Passage data from Columbia River DART (Data Access in Real Time) repository developed by Columbia Basin Research. This adult passage data is a time-series data set with daily salmon counts taken from Bonneville Dam dating back to 1938 (Columbia River DART, 2021). This data was combined into a multivariate input with other correlated covariates to salmon survival. These covariates are Bakun Upwelling index taken from 45N 125W (location outlined in Burke et al (2013)), Northern Oscillation Index (NOI), North Pacific Grye Osciallation (NPGO), Pacific Decdal Osciallation (PDO), Oceanic Niño Index (ONI). These environmental co-factors are known to affect fish survival in the North Pacific and play a correlated role in predicting salmon survival at sea (Burke et al., 2013). This data was gathered from a variety of public data repositories and is on a monthly time-stamp dating back to 1950 (Bakun 1973; Di Lorenzo et al., 2008; ERDDAP 2021; Jacox et al., 2018; Japan Meteorological Agency 2006; Mantua et al., 1997; Newman et al 2016; PSL 2021).

The Bakun Upwelling index is the standard upwelling index used in marine science dating back to the 20th century. Upwelling serves as a proxy for how productive the ocean is (Bakun 1973., 1997; Jacox et al., 2018). The NOI measures climate variability between the North Pacific High and the tropics. This is a proxy for the temperature and productivity in the entire North Pacific Basin (Schwing et al., 2002). Similarly, NPGO also servers as a proxy for temperature and the productivity in the North Pacific (Di Lorenzo et al., 2008). Full basin-scale indices were also used such as PDO and ONI. These were proxies for temperature and climate shifts in the Pacific Ocean (Di Lorenzo et al., 2008; Mantua et al., 1997; Newman et al., 2016).

Our dataset consisted of 24,734 days or 992 months of salmon count data. This data was initially prepossed down to 24,369 days for our single-variable daily models and 984 months for our single-variable monthly models. The daily data was broken down on a 180 day time-stamp to predict the 181st day salmon count. This was done from 1939 to 2015 for our training set and 2016 to 2020 for our test set. For the monthly models, the data was broken down to take the last 6 months of data to predict month 7 salmon count. Again, our training set was split to take all months from 1939 to 2015 and our test set was split to predict on 2016 to 2020. No development set was used due to the smaller nature of our dataset.

For the multi-variable models, the monthly salmon data was used starting in 1950. January 1950 was chosen as the starting year for our environmental model because this is the earliest where we could get data for all our variables. Here, we have a total of 852 examples x 6 features (Salmon count, Upwelling, NOI, NPGO, PDO, ONI). This was then split down into 792 x 6 examples for our training set, 54 x 6 examples for our development set, and 54 examples for our test set. We chose to use a development set for these more complicated models to be able to analyze if the model is overfitting due to the increase amount of data.

4 Methods

4.1 Baseline

To evaluate our DL algorithms, we had to develop a series of baselines on the salmon data. We used linear regression, lasso regression, and ridge regression to create our baseline. Although there are other methods for forecasting salmon, such as moving average or Beverton-Holt stock-recruitment models, linear regression made the most sense due to nature of our data (daily and monthly time-step)(McCormick & Falcy, 2015). Furthermore, linear regression, and versions of linear regression, have been used to forecast salmon returns (Hand & Haeseker, 2011). Linear regression models the correlation between two variables by attempting to fit a linear equation to the data. It follows the formula of $Y = a + b*X$, where X is the independent variable and Y is the dependent variable.

We also chose to incorporate lasso (L1) and ridge (L2) regression as other baselines. This gave us more data points to test our model. Lasso regression is very similar to linear regression but adds in L1 regularization. This works by penalizing the model to the absolute value of the magnitude of coefficients. This, in essence, knocks out certain coefficients and makes the model simpler which can improve forecasts. On the other hand, ridge regression use L2 regularization. L2 regularization puts a penalty on the model equal to the square of the magnitude of the coefficients. This shrinks all the coefficients by the same amount.

4.2 Fully Connected Neural Network (NN)

The next model, in order of increasing complexity, is a fully connected neural network. We built a 4-layer neural network for both the daily and monthly data. On the daily data, the model takes in the last 180 days of data and predicts the number of salmon on the 181th day. On the monthly data, the model takes in the last 6 months and predicts the number of salmon that return in month 7. This prediction is then compared to actual number of salmon on that day and the loss is computed. The loss function that we used is mean squared error because it is common in DL time-series problems (Bernhard, 2020; Brownlee 2018).

4.3 Recurrent Neural Network (RNN)

We also built a Recurrent Neural Network on both the daily and monthly data. We choose to build an RNN due to its increasing popularity in time-series forecasting (Bernhard, 2020). This RNN came in a simple and deep variety. The simple network had only a single layer plus one dense single node layer while the deep network used 4 recurrent layers and single node dense layer. The RNN was built in a many to one fashion taking in an input of 180 days (on the daily model) and producing a single output of the 181th day. The loss function was mean squared error.

4.4 Grated Recurrent Unit (GRU)

Going off the RNN, we also explored a Grated Recurrent Unit. GRU, like RNNs, have become more in fashion for time-series forecasting (Bernhard, 2020; Brownlee, 2018). Our GRU model was built in both a simple and robust manner, with the simple having one layer plus one dense single node layer and robust having 4 GRU layers plus one dense single node layer. Our loss function was mean squared error.

4.5 Long-Short Term Memory (LSTM)

The final model we built was a simple and deep Long-Short Term Memory on both the daily and monthly data. LSTM are the most powerful recurrent network and, like other recurrent-based networks, have started to be used more and more for time-series applications. Both our simple and deep LSTM are built in a many-to-one design taking X number of inputs (days, months, years) and produceing a single Y output (number of salmon on a day, month, or year). Our simple LSTM had 1 LSTM layer with one single node dense layer. Our deep layer had 4 LSTM recurrent layers. The loss function was mean squared error.

4.6 Loss Function:

Mean Squared Error

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2$$

	Daily train (RMSE)	Daily test (RMSE)	Monthly train (RMSE)	Monthly test (RMSE)
Linear Regression	520.17909748149	1,417.72345784	27,009.21969347	68,350.227197460
Linear Regression (Lasso)	538.40209475896	**1,356.76206771**	27,009.21964101	68,350.227197460
Linear Regression (Ridge)	520.18060999071	1,416.47162712	70,010.61354592	68,346.745405458
Fully Connected NN	**454.56543041359**	1,400.91040046	7,141.462663113	36,717.236100895
Simple RNN	570.6108536758	1,367.21232579	17,280.78600323	53,707.14979473
Simple GRU	527.80208758208	1,391.49315249	15,174.32081730	**33,819.83961019**
Simple LSTM	664.93200066769	1,375.12810413	17,317.24385919	66,531.72241765
Deep RNN	543.76380032359	1,459.21366956	10,405.46462434	54,480.46181396
Deep GRU	558.64816256459	1,390.34824052	15,649.68598451	42,762.87195427
Deep LSTM	**857.01960270232**	**1,575.76855559**	23,958.18224178	56,947.76536959
Simple Multivar RNN	N/A	N/A	17,301.78071760	87,677.12988003
Simple Multivar GRU	N/A	N/A	9,861.972216549	60,721.38074846
Simple Multivar LSTM	N/A	N/A	**54,361.83837951**	54,361.83837951
Deep Multivar RNN	N/A	N/A	**3,160.481292461**	63,095.49232710
Deep Multivar GRU	N/A	N/A	28,662.78311678	46,821.86702813
Deep Multivar LSTM	N/A	N/A	39,630.12833691	**93,912.36221073**

Figure 1: Results of all models

5 Experiments/Results/Discussion

5.1 Experiments

First and foremost, we want to make it clear that out project is focused more on the comparison across different DL models instead of extensive hyper-parameter tuning, to get a broad idea of whether or not DL is a viable or promising approach for improving salmon forecasting. One of the things we experimented with was our optimizer. We had initially started with SGD, but when we saw that this made our models struggle to converge their loss function, we switched to the Adam optimizer. Adam worked much better and we decided to use it in all of our models for consistency. Another hyperparamter we experimented with was mini batch size. For our all GRU models except our multi variable models, we used a mini batch size of 150. We used a larger mini batch size for the multi variable models of 1,000 in order to speed up the computation and allow for smoother convergence of the loss. We followed the same scheme for our LSTM models, excpet that our multi variable shallow LSTM provided better results when ran on a mini batch size of 2000. For our classic RNN models, we used a small mini batch size of 64 to speed up convergence, at the cost of increased oscillation of the loss. For the multi variable RNN models, we used a mini batch size of 100 as we found it to deal better with the addition of 5 more covariates. Finally, we used a mini batch size of 100 for our fully

connected Neural Networks, however we did not experiment much with these as they predominantly serve as a baseline.

5.2 Evaluation Metric:

Since this is a regression model, we used root mean squared error (RMSE) as our evaluation metric for all models. At one point in the project, we were comparing our models' results to the traditional yearly salmon forecasts (using official forecasting data), however we quickly realized that we were not making a fair comparison since we are using the last 180 days to make a prediction on the following day, whereas the traditional forecast uses the last couple years to predict the following year. After this realization, we created basic linear regression baselines that included a classic linear regression model, a linear regression model with Lasso penalization, and a linear regression model with Ridge penalization that uses the same 180 day input to 1 day output scheme as our models in order to have a more fair comparison.

5.3 Quantitative Results:

See figure 1.

5.4 Qualitative Results:

Overall, our models that use a daily time stamp work significantly better than those with a monthly time stamp. This makes sense since a daily time series significantly increases the amount of data we have, thus making it a more apt model for DL. More specifically, our shallow RNN, LSTM, and GRU, that only take daily salmon count as input, seemed to work the best on daily time stamps. On the other hand, our LSTM models (all LSTM models listed in section above) seem to be performing the worst. We hypothesize that this is due to the cyclical/seasonal nature of our time series, which contains long stretches of zeros. The long term correlation capabilities of the LSTM may be focusing too much on these zeros and doing a worse job on the actual salmon runs. Overall, our models with multi variable input fail to add any level of skill to our established monthly predictions. This is because we are overfitting the training set. Please refer to section 5.3 on row "deep multivar RNN". This model had very low train set RSME, but a much higher RMSE on the test set. If we had more time, it would be great to do our own analysis as to which covariates add the most skill to the model, instead of relying on the analysis of previous studies to select covariates. Although there is a plethora of evidence that our environmental covariates are useful markers for salmon runs, nobody has ever tried using this data in DL models, therefore further analysis may have to be done.

6 Conclusion/Future Work

There is potential that Deep Learning could be utilized to improve salmon forecasting. The simple GRU was able to out perform the baseline on the monthly data. However, on the daily data, our baseline regression methods and DL models performed about the same with Lasso Regression having the lowest error. This was surprising considering that daily models were fed the largest amount of data and DL algorithms tend to perform better in big data scenarios. We think that maybe this occurred due to the lack of environmental data in the daily models. In the multi-variable models, the majority of our DL models outperformed the baseline.

If we had more time, we would have like to build these models further and add in the yearly time-stamp data as well. Furthermore, we would have incorporated more models to compare to our DL algorithms. It would also be interesting to apply transfer learning to see if the model could produce good results on other river systems. Currently, our model is only trained on salmon that return to the Columbia River so it would be interesting to see if we could successfully forecast salmon that return to the Sacramento River, for example. If this ended up being the case, then it would show that using DL as a forecasting tool is scalable and could replace current forecasting methods.

7 Contributions

Everyone contributed to the project. Ismael focused primarily on model development, data preposessing, and cleaning of code. Chris helped with biological background, dataset creation, and model development. Abdul worked on logistics (GitHub, AWS), model development, and model training. However, everyone worked together and collaborated across each other's responsibilities.

8 Code

https://github.com/Abdul-Omira/SalmonNet

References

[1] *American seafood industry steadily increases its footprint.* American seafood industry steadily increases its footprint | National Oceanic and Atmospheric Administration. (2018, December 18). https://www.noaa.gov/media-release/american-seafood-industry-steadily-increases-its-footprint.

[2] Bakun, A. (1973). Coastal upwelling indices, west coast of North America. US Department of Commerce. NOAA Technical Report, NMFS SSRF-671

[3] Bernhard, J. (2020, July 14). Predicting Stock Prices Using Deep Learning Models. Medium. https://medium.com/swlh/predicting-stock-prices-using-deep-learning-models-310b41cec90a

[4] Brownlee, J. (2020, October 20). Multivariate Time Series Forecasting with LSTMs in Keras. Machine Learning Mastery. https://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras/.

[5] Brownlee, J. (2018). Deep Learning for Time Series Forecasting.

[6] Burke, B. J., Peterson, W. T., Beckman, B. R., Morgan, C., Daly, E. A., Litz, M. (2013). Multivariate models of adult Pacific salmon returns. PloS one, 8(1), e54134.

[7] Columbia River DART, Columbia Basin Research, University of Washington. (2021).

[8] Di Lorenzo et al., 2008: North Pacific Gyre Oscillation links ocean climate and ecosystem change, GRL.

[9] ERDDAP, NOAA (2021). *Oscillation Indices (NOI, SOI, SOI*, *Monthly, 1950 - 2020* (ERDDAP) [Data Access Form]. https://coastwatch.pfeg.noaa.gov/erddap/griddap/erdlasNoix.html

[10] Folland, C. K. and D. E. Parker, 1995: Correction of instrumental biases in historical sea surface temperature data. Q. J. R. Meteorol. Soc., 121, 319-367.

[11] Hand, David and Haeseker, Steve, 2011. Retrospective Analysis of Preseason Run Forecast Models for Warm Springs stock Spring Chinook Salmon in the Deschutes River, Oregon. USFWS-Columbia River Fisheries Program Office. https://www.fws.gov/columbiariver/publications/Analysis$_o f_W arm_S prings_S tock_S pring_C hinook_S almon_F orecasting_M ethods.pdf$

[12] Hilborn, Ray, et al. BBRSDA, 2020, Improving Preseason Forecasts with Artificial Intelligence Methods and Ecosystem Information.

[13] McCormick, J.L. and Falcy, M.R. (2015), Evaluation of non-traditional modelling techniques for forecasting salmon returns. Fish Manag Ecol, 22: 269-278. https://doi.org/10.1111/fme.12122

[14] Jacox, M. G., C. A. Edwards, E. L. Hazen, and S. J. Bograd (2018) Coastal upwelling revisited: Ekman, Bakun, and improved upwelling indices for the U.S. west coast. Journal of Geophysical Research, doi:10.1029/2018JC014187.

[15] Kurlansky, M., Lichatowich, J., Gayeski, N. (2020). Salmon: A Fish, the Earth, and the History of Their Common Fate. Patagonia.

[16] Mantua, N.J., S. R. Hare, Y. Zhang, J. M. Wallace, and R. C. Francis, 1997: A Pacific Interdecadal Climate Ooscillation with Impacts on Salmon Production. Bull. Amer. Meteor. Soc., 78, 1069-1079.

[17] Newman, M., M. A. Alexander, T. R. Ault, K. M. Cobb, C. Deser, E. Di Lorenzo, N. J. Mantua, A. J. Miller, S. Minobe, H. Nakamura, N. Schneider, D. J. Vimont, A. S. Phillips, J. D. Scott, and C. A. Smith, 2016: The Pacific Decadal Oscillation, Revisited. J. Clim., DOI: 10.1175/JCLI-D-15-0508.1

[18] Humdata, OCHA (2021). *Oceanic Niño Index Data, Monthly 1950 - 2017* (OCHA Services) [Datasets]. https://data.humdata.org/dataset/monthly-oceanic-nino-index-oni

[19] PSL, NOAA (2021). *Oceanic Niño Index Data, Monthly 2017 - 2020* (PSL) [Data Access Form]. https://psl.noaa.gov/data/correlation/oni.data

[20] Schwing, F. B., Murphree, T., Green, P. M. (2002). The Northern Oscillation Index (NOI): a new climate index for the northeast Pacific. Progress in oceanography, 53(2-4), 115-139.

[21] Xu, Y., Hawkshaw, M., Fu, C., Hourston, R., Patterson, D., Chandler, P. 68. ESTIMATING FRASER RIVER SOCKEYE SALMON RUN SIZE USING A MACHINE LEARNING METHOD. State of the Physical, Biological and Selected Fishery Resources of Pacific Canadian Marine Ecosystems in 2019, 273.

[22] Zhou, S. (2003). Application of artificial neural networks for forecasting salmon escapement. North American Journal of Fisheries Management, 23(1), 48-59.