Automated system for forecasting and tracking of number of online search queries at scale

Manisha Gupta  
Department of Computer Science Univeristy of North Dakota Grand Forks, ND 58202 U.S.A, [manisha.gupta@und.edu](mailto:manisha.gupta@und.edu) Eunjin Kim  
Department of Computer Science Univeristy of North Dakota Grand Forks, ND 58202 U.S.A, ejkim@cs.und.edu

# Abstract

Neural networks and long short-term memory (LSTM) have been used intensively in recent years due to their capability to model and forecast nonlinear time-series. The present paper delivers a comprehensive overview of neural network cell derivatives and neural prophet model architectures for time series prediction. With the degree of extension of time series data, explainable forecasting remains a challenging task for business and operational decision making. Hybrid solutions are needed to bridge the gap between interpretable classical statistical methods and scalable deep learning models. The investigated approaches are evaluated against defined requirements for forecasting online search queries being relevant for an accurate time series prediction. These include short-term and long-term memory behavior, the ability for multi-step ahead predictions and the according error propagation. Sequence-to-sequence networks with partially conditioning outperform the other approaches.

Keywords— Explainable, Forecasting, Neural networks, Time series, Deep learning, LSTM, Neural Prophet

# Introduction

Forecasting online user trends is a data science task that aids organizations with planning, goal setting, and anomaly detection. Despite its importance, there are major challenges associated with producing reliable and high-quality forecasts – especially when there are a variety of time series for several geographic regions. To address these challenges, we demonstrate a useful approach to forecasting using Facebook prophet and neural network(LSTM). We propose these two models with interpretable parameters that can be intuitively adjusted with domain knowledge about the time series. We describe performance analyses (Mean absolute percentage error, Root mean square, mean square error etc.) to compare and evaluate our daily forecasting, and automatically flag forecasts for manual review and adjustment. An accurate forecast helps to monitor search engine business-critical metrics and quickly detect and debug noteworthy deviations from the expectation. This will enable the larger search engine team to initiate planning and strategic pivots quickly.

The goal is to accurately forecast online search queries trend on one of the biggest search engines with the capability to address different types of past and future events, trend, several seasonal effects such as weekly, yearly and identification of the anomalies with its solution. Holidays such as Thanksgiving, Christmas etc. are also big predictable shocks to many business time series as online user search pattern differs on holidays as compared to normal day and often do not follow a periodic pattern, so their effects need to be modeled well to capture and explain their effect on the time series.

## Related Work

Time series data is vital in most industrial sectors. Though extensively studied in theory and applications such as economics, practical forecasting in business planning has not received widespread attention until recently. Accurate time series forecasting is essential for decision-making processes, especially in planning, budget allocation and supply chain management. Over-forecasts as well as under-forecasts can both be costly. In applications, where forecasts inform business or operational decisions with potential consequences are fatal, it is often a requirement for the model to be explainable. A common approach is to break the forecast down into components which are individually interpretable.

We have several classical time series algorithms such exponential smoothing, SARIMA, etc. However, we found each methodology has its own pros and cons. Some algorithms such as SARIMA exhibit very good results, however the downside is complexity. Tuning and optimizing SARIMA models is often computationally expensive and successful results can depend upon the skill and experience of the forecaster. It is not a scalable process, but better suited to ad hoc analyses by skilled practitioners. Exponential smoothing is also a good approach as it applies exponentially decreasing weights to the values being averaged over time, giving recent values more weight and older values less. This allows the forecast to be more reactive to changes, while still ignoring a good deal of noise. The downside of these methodologies is either some does not allow for trend or seasonality, or slow to adjust to new trends hence forecasted values lag behind the reality or there are many hyperparameters to tune which can be a difficult and very time-consuming process.

We are proposing using Neural network and Facebook prophet-based customizable prediction that can help to forecast better as compared to above mentioned methodology. The major benefit is that it provides accurate and robust forecasts that completes the whole training fit process in a few mins. Furthermore, it also provides Bayesian statistics to create uncertainty intervals for future predictions to add a data – driven estimate of forecasting risk.

# APPROACH

Identify applicable funding agency here. If none, delete this text box.

In this comprehensive paper, we will study neural prophet and Long short term memory recurrent network. To evaluate any forecasting model, it is crucial to compare to set a baseline method. We would prefer to have a simplistic forecast (moving average of last four value). We will learn the neural network model and Facebook prophet-based model to forecast the number of online search queries. The forecasting timeframe is daily.

## LSTM Neural networks

Neural networks have been applied in the scope of numerous applications to model and predict complex system dynamics. There is a wide range of network types available, but modeling accuracy is strongly dependent on the fit of network architecture and considered a problem. This paper presents an overview of neural networks, with a focus on Long short-term memory (LSTM) networks, and neural prophet model that have been used for dynamic system modeling in diverse application areas. The common aim in the scope of all investigated problems is the setup of machine learning forecasting models based on time series data or data sequences to forecast nonlinear time-variant system output. This paper examines existing approaches regarding the following properties:

• Nonlinear and time-variant prediction ability

• Short-term and long-term memory behavior

• Multi-step ahead prediction and error propagation

The main neural network model in Keras requires six different steps: preparing training data, creating a basic network model, calculating loss function, selecting the learning rate, and optimizing the loss function with respect to the parameters of the model. It also includes of several layers, and pointwise operations which act as gates for data input, output and forget which feed the LSTM cell state. This cell state is what keeps the long-term memory and context across the network and inputs. The n\_feature shows the number of neurons in the input layer. We use the n\_hidden equal to 100, which means 100 neurons in the hidden layer 1, and the output neuron is 1. The role of the optimization function is to minimize the loss function defined with respect to the parameters and the learning rate. The learning rate chosen here is 0.01. We also pass the neural network parameters into the optimizer. There are various optimization functions.

* SGD . Implements stochastic gradient descent. The parameters could be momentum, learning rate, and weight decay.
* Adadelta . Adaptive learning rate. It has five different arguments, parameters of the network, a coefficient used for computing a running average of the squared gradients, the addition of a term for achieving numerical stability of the model, the learning rate, and a weight decay parameter to apply regularization.
* Adagrad . Adaptive sub gradient methods for online learning and stochastic optimization. It has arguments to optimize the learning rate and learning rate decay with weight decay.
* Adam. A method for stochastic optimization. This function has six different arguments, to optimize, learning rate, betas (known as coefficients used for computing running averages of the gradient and its square), a parameter to improve numerical stability, and so forth.
* ASGD.Acceleration of stochastic approximation by averaging. It has five different arguments, sequentially iterate of parameters to optimize, learning rate, decay term, weight decay, and so forth.
* RMSprop. Uses a magnitude of gradients that are calculated to normalize the gradients.
* SparseAdam. Implements a lazy version of the Adam algorithm suitable for sparse tensors. In this variant, only moments that show up in the gradient are updated, and only those portions of the gradient are applied to the parameters.

Then we introduce the back-propagation function using the rectified linear unit as the activation function in the hidden layer.

The tuning parameters that can increase the accuracy of the supervised learning model, which is a regression use case, can be achieved with the following methods.

• Number of iterations

• Type of loss function

• Selection of optimization method

• Selection of loss function

• Learning rate

• Decay in the learning rate

• Momentum requires for optimization

## NeuralProphet

Neural Prophet is a hybrid forecasting framework based on PyTorch and trained with standard deep learning methods, making it simple for developers to extend the forecasting framework. Local context (such as how many lag variables are associated with the next term) is introduced with auto-regression and covariate modules, which can be configured as classical linear regression or as Neural Networks.

### The NeuralProphet Model

Neural Prophet is fit with stochastic gradient descent - more precisely, with an AdamW optimizer and a One-Cycle policy. When the parameter learning rate is not specified, a learning rate range test is analysed to determine the optimal learning rate. Gradient Descent for optimization via using PyTorch as the backend. It is modelling Auto-Regression of time series using AR-Net and modelling lagged regressors using a separate linear or Feed-Forward Neural Network. We can train a single model on many related time-series (global modelling).

Due to the modularity of the code and the extensibility supported by PyTorch, any component trainable by gradient descent can be added as a module to Neural Prophet. Using PyTorch as the backend makes the modelling process much faster compared to original Prophet which uses Stan as the backend.

### Model Components

A basic concept of the Neural Prophet model is its modular decompose ability. The model is consists of modules which each contribute an additive component to the forecast. Most components can also be configured to be scaled by the trend for a multiplicative effect. Each module has its individual inputs and modelling processes. However, all modules must produce h outputs, where he defines the number of steps to be forecasted into the future at once. These are summed up as the predicted values ˆyt , ..., yˆt+h−1 for the time series future values yt , ..., yt+h−1. If the model is only time-dependent, an arbitrary number of forecasts can be produced. In the following descriptions, that special case will be treated mathematically equivalent to a one-step ahead forecast with h = 1.

yˆt = T(t) + S(t) + E(t) + F(t) + A(t) + L(t)

where, T(t) = Trend at time t S(t) = Seasonal effects at time t E(t) = Event and holiday effects at time t F(t) = Regression effects at time t for future-known exogenous variables A(t) = Auto-regression effects at time t based on past observations L(t) = Regression effects at time t for lagged observations of exogenous variables

**Trend-** A classic approach to modelling trends is to model it as the combination of an offset m and a growth rate k. The trend effect at a time t1 is given by multiplying the growth rate by the difference in time since the starting point t0 on top of the offset m. T(t1) = T(t0) + k · ∆t = m + k · (t1 − t0)

**Seasonality-** Every time series has a certain rhythm to it. That rhythm may be yearly, weekly, monthly where the pattern follows a certain cycle. On the weekdays we will receive different search queries trend vs weekend. User search behavior could differ on weekday vs weekend. We could expect on Monday the search queries volume would be up by 15%, however it would be down by 35% on Saturday. Furthermore, Summer weekends could have a dissimilar effect than winter season weekend. Momentum is not linear. At times we can have huge growth and on other times despite doing everything right, we can become stagnant series. If we don’t capture these seasonal effects correctly it will affect the business for goal planning/ organization in the long term. We can capture this seasonal swing most robustly and we have in our model the most robust seasonality mode enable.

Seasonality in Neural Prophet is taken care of by using Fourier terms. In this technique, a number of Fourier terms are defined for each seasonality, where k refers to the number of Fourier terms defined for the seasonality with periodicity p. Fourier terms are defined as sine, cosine pairs and allow to model multiple seasonality’s as well as seasonality’s having non-integer periodicities such as yearly seasonality with daily data (p = 365.25) or with weekly data (p = 52.18). In a multiple seasonality scenario, different values for n can be defined for each periodicity.

**Holidays-** The holidays have a large role to play on user search behavior and business activity. Furthermore, each holiday impact is different from each other. For example, July 4 (Independence Day) we can expect 40% reduction in search queries volume however, during Columbus day, it has only 5% reduction. The other important part of a holiday is sometimes, it is not a single day event rather than it will span over several days. For example, good Friday to easter Monday or thanksgiving to cyber-Monday. Hence these are 4 to 5 days’ effect, and we need to account for in our forecast.

Chart, line chart

Description automatically generated

Figure . Neural Prophet model detecting holidays.

**Changepoint detection-** Changepoint are the locations that frequently exhibit abrupt changes in their trajectories in time series. There could be many reasons why these changepoints occur. It could be a new feature launch, or a new growth initiative etc. We see the trend is changing during this period. It could be a downward or upward trend. For example, modeling the number of daily active users and seeing a sudden change of trend upon the release of a new initiative such as rewards. These changepoints require special care. Our model has a feature to capture these changes automatically, calculate the magnitude of the effect to adjust the trend. We also provided hybrid technique let say model could not capture some of the initiate, we can feed those dates in the model and get the trend fitted forecast accordingly.

**Regressor-** To refine our forecast further, we have added a feature called external regressor to provide additional information to the model to adjust the forecast. Yet simple but more robust.

For example, non-occurring evet- such queen ‘s death or new Growth initiative for which we know the future dates and estimated effect. By adding this information in the model leads to greater predictive power. If we don’t provide this information to the model, errors will continuously pile up and the forecast will not provide an accurate snapshot of business for next year.

This feature improved the confidence interval of the forecast.

# Obtained Result Comparison

### Implementation

The implementation phase has several steps such anomaly detection, missing data treatment etc. During data preprocessing we detected some of the data anomalies that degrade forecasting accuracy. It includes discovering the anomalies, correcting outliers that cause seasonality swing and correcting outliers that cause wide uncertainty intervals. To identify the anomaly, we utilized seasonal trend loss technique in which we decompose the time series using seasonal decomposition and remove trend and seasonality to generate e residual time series and detects the points in the residual which are outside of 3 times of interquartile range.

Timeline

Description automatically generated

Figure

We also examined the stationarity test by understanding the Fourier transform. To solve missing value treatment there are several methods are used such as forward fill and backward fill that is nothing but the simplest way to fill in missing value is to carry forward the last known value prior.

Data has been explored by plotting the line chart between number of search queries and dates to understand the trend behavior.

Chart, histogram

Description automatically generated

Figure

The above chart depicts there is a trend and seasonal component present in the data set as each year is showing similar user behavior. We also investigate the data at a deep level by decomposing it into trend, seasonal and residual parts. Trend chart shows that there is an increasing trend from Oct 2021. We also illustrated the seasonal part of weekday by analyzing Monday to Sunday data. During the weekend, we observed that the trend is positive, and it slows down during the weekend.

Chart, line chart

Description automatically generated

Figure

Graphical user interface, application, table, Excel

Description automatically generated

Figure

Chart, line chart

Description automatically generated

Figure

### Hyper-parameter tuning

Neural Prophet provides several parameters to train the model for forecasting. It is very important to get the right value for reliable forecast in the future. To do so, we apply grid search algorithm. Different hyperparameter values produce different forecasts for a given data set. Hyperparameter tuning consists of finding a set of optimal hyperparameter values for a learning algorithm while applying this optimized algorithm to any data set.

On a typical laptop, we can expect it will take around 8-12 hours to complete; to speed up the example, you may consider reducing the number of parameters. With these parameters, a grid search will iterate through each unique combination, use cross-validation to calculate and save a performance metric, and then output the set of parameter values which resulted in the best performance.

### Evaluation

To evaluate forecasting model performance, we will use three error metrics as follows-

1. **Mean absolute percentage error** (MAPE) is a statistical measure of how accurate a forecast system is. It is a measure in terms of percentage. It is mostly used for time-series forecasting. The closer to zero the error is, the better the model.

2. **Root mean square error (RMSE)** is a quadratic scoring rule that also measures the average magnitude of the error. It’s the square root of the average of squared differences between prediction and actual observation. The closer to zero the error is, the better the model.

**3. Coverage** - Coverage is simply the percentage of actual values that lie between the predicted upper and lower uncertainty bounds.

We implemented a baseline model by taking the last four days of moving average to forecast next value. The LSTM model network architecture we define with 100 neurons with one hidden lay, adam optimizer, Relu activation and 50 Epoch and batch size is 1. The min max scaler has been used to normalized the data. The backend framework is TensorFlow and karas library. The neural prophet is using Pytorch framework and neural network.

Chart, line chart

Description automatically generated

Figure

In the above graph, yellow line is showing baseline model prediction, grey line showing LSTM model prediction, orange line showing neural prophet prediction and blue line is showing actual data we can see that baseline model and LSTM model does not fit very well. However, the orange line (Neural prophet) follows the Actual line(blue line) very closely and depicts good model fit. We witnessed that neural prophet prediction outperformed the baseline and LSTM forecast. The below table demonstrate the mean absolute error percentage comparison among the different models-

## 

Table

# Conclusion

##### This paper has analyzed the moving average, LSTM and Neural prophet algorithm in the domain of machine learning and deep learning techniques used to forecast time series. The reviewed papers and works have been classified as baseline models for forecasting. Neural network ANN and neural prophet are techniques used to forecasting daily timeframe, and the evaluation techniques employed. In regard to the employed machine learning technique, the use existing artificial neural network models which are enhanced with new training algorithms or combined with emerging technologies into hybrid systems is limited. This finding indicates that neural network-based neural prophet algorithm outperformed and is well suitable in the domain of online search queries forecasting. Lagged data has been identified as the most popular input parameters in the literature. The paper presents an overview of LSTM architectures and Neural prophet that are developed to predict nonlinear time series behavior. There is a several architectures available that have been used in wide application areas such as image processing, manufacturing, or autonomous systems. In the scope of this paper, the approaches are categorized and evaluated with regard to defined properties. In summary, there seems to be a consensus to use deep learning methods for forecasting purposes. Neural prophet developed by Facebook has been identified as the dominant machine learning technique in this area. The main finding of this paper is that there is a lack of literature examining if machine learning techniques can provide the correct user behavior under real world constraints such as covid, holiday etc. and we implemented the holiday effect or any special event to model the forecast and calculate the impact accordingly in relation to forecast. The correlation with weather data and market indicators such as Dow jones, or Nasdaq have not studied yet.

# References

1. Financial time series forecasting with machine learning techniques: A surveyBjoern Krollner *(references)* [*https://pure.bond.edu.au/ws/files/27498056/Financial\_time\_series\_forecasting\_with\_machine\_learning\_techniques.pdf*](https://pure.bond.edu.au/ws/files/27498056/Financial_time_series_forecasting_with_machine_learning_techniques.pdf)
2. <https://www.semanticscholar.org/reader/c441b5763d02b19b024971256ffcd65c60370f43>
3. NeuralProphetExplainableForecastingatScale <https://arxiv.org/pdf/2111.15397.pdf>
4. Survey on Automated Machine Learning (AutoML) and Meta learning
5. <https://www.researchgate.net/publication/351294749_A_survey_on_long_short-term_memory_networks_for_time_series_prediction>
6. <https://pure.bond.edu.au/ws/files/27498056/Financial_time_series_forecasting_with_machine_learning_techniques.pdf> Financial time series forecasting with machine learning techniques