Abstract

Automatic forecasting of time series data is a challenging problem in many industries. Current forecast models adopted by businesses do not provide adequate means for including data representing external factors that may have a significant impact on the time series, such as weather, national events, local events, social media trends, promotions, etc. This paper introduces a novel neural network attention mechanism that naturally incorporates data from multiple external sources without the feature engineering needed to get other techniques to work. We demonstrate empirically that the proposed model achieves superior performance for predicting the demand of 20 commodities across 107 stores of one of America’s largest retailers when compared to other baseline models, including neural networks, linear models, certain kernel methods, Bayesian regression, and decision trees. Our method ultimately accounts for a 23.9% relative improvement as a result of the incorporation of external data sources, and provides an unprecedented level of descriptive ability for a neural network forecasting model.

1. Introduction

Univariate forecasting techniques, such as Holt-Winters (Holt, 1957), (Winters, 1960) and ARIMA (Box & Jenkins, 1990), are widely adopted in industry. These methods are used for performing predictions that are crucial for supporting logistical needs, such as product demand and consumer behavior. However, univariate forecasting techniques, by definition, do not take into account multiple data sources, and often come short of providing a fully automatic forecasting method. In fact, (Franses & Legerstee, 2009) conducted a case study where 90% of all forecasts were found to be manually adjusted. Sales for most products seems to vary wildly based on external influences.
There are multiple drawbacks associated with human
driven prediction methods that inspire the need for a fully
automatic solution. One such issue is that humans have
biases when analyzing the impact of external data sources
(Lawrence et al., 2006). For example, humans were found
to often have an optimism bias in their projections of the
impact of promotions in (Fildes et al., 2009), (Trapero
et al., 2011), and (Trapero et al., 2013). Humans also may
not be knowledgeable of external factors such as local or
national events in advance when adjusting a forecast.

Neural networks have a substantial history of quantitatively
superior performance to industry standard demand forecasting
techniques in literature (Xu et al., 2005), (Castillo
et al., 2006), (Sunaryo et al., 2011), (Cortez et al., 2012),
(Marvuglia & Messineo, 2012), (Neupane et al., 2012).
However, they have ultimately not been widely adopted in
industry because their modest improvements have come at
the cost of not being interpretable. In Figure 1 we provide
eamples of the high level of interpretability that our pro-
posed model exhibits. Forecasts, especially in the business
setting, are often used as an aid for human decision making.
Consequently, it is essential that such a system provides in-
formation about which features are contributing to a fore-
cast outcome. For example, interpretability could mitigate
financial and legal ramifications when a forecasting system
commits large errors.

In this paper, we take the first steps towards creating a
neural network-based forecasting system that is (i) scal-
able, (ii) adaptable to multiple data sources and (iii) inter-
pretable. We propose a paradigm where a baseline fore-
cast is adjusted by a series of observations related to an ar-
bitrary number of external feature groups (“factors”), and
each observation has an interpretable additive effect on the
baseline. This is achieved with a novel neural network at-
tention model. To our knowledge, we are the first to show
the power of neural network attention mechanisms in the
domain of time series forecasts.

2. Related Work

Applying a content based attention mechanisms in neural
networks is a recently proposed idea that is having a broad
impact across many disciplines of machine learning. Since,
it was first proposed in the field of Machine Translation in
(Bahdanau et al., 2014), it has been shown useful for exam-
ple in speech recognition (Chorowski et al., 2015), image
caption generation (Xu et al., 2015), reading comprehen-
sion (Hermann et al., 2015), and video description gener-
ation (Yao et al., 2015). As noted by the authors of (Cho
et al., 2015), attention mechanisms can be most beneficial
in scenarios where both the input and output have a rich
structure. However, attention can also be highly beneficial
over other neural network approaches in cases where the in-
put has a rich structure and the output is simple. For exam-
ple, the use of soft attention mechanisms produce state of
the art results for classification of textual entailment (Wang
& Jiang, 2015), (Rocktäschel et al., 2015). As we will de-
scribe in the next section, the problem of forecasting de-
based on many auxiliary data sources should be natu-
really posed as a problem with a rich input, making attention
mechanisms an attractive approach.

Our work is related to a possible incarnation of a one-
level Hierarchical Mixture of Experts (HME) model (Ja-
cobs et al., 1991), (Jordan & Jacobs, 1994) where the ex-
pert networks are each learned over a different group of
features that are explicitly parsed during the instantiation
of the model. While there are many implementation dif-
ferences, the most significant architectural differences are
between the HME gating network and our proposed soft
attention mechanism. Soft attention mechanisms learn at-
tention weights from a classifier on top of the hidden rep-
resentations, rather than basing it on the input representation
as done in the analogous HME gating network. Our ex-
periments show that our same setup trained like a gating
network, where attention units are based off the input rep-
resentation, achieves substantially worse performance than
the model trained with hidden representation based atten-
tion. Our intuition is that utilizing the hidden representation
should be more powerful due to more learnable parameters
and more generalizable because our hidden layers tend to
be small relative to the input feature size. Additionally, be-
cause the hidden layer weights are shared between both the
attention score and output vector, the representation is bi-
as in trying to solve for the attention score in a way that
may improve generalization as demonstrated for multi-task
neural networks in (Caruana, 1997).

Weather (Starr-McCluer et al., 2000), (Taylor & Buizza,
2003), and social media signals (Chen & Du, 2013), (Si
et al., 2013), have been considered in literature for time
series prediction applications before. However, the authors
are not aware of any attempt in literature to use both of
them on the same task before our work.

3. The Multifactor Neural Network Attention
Model

One limitation of most traditional predictive modeling
techniques, including Lasso Regression, Logistic Regres-
sion, Support Vector Machines, and MLP models is that
they require input features to be represented with a vector
for each prediction step. Although it is possible in prin-
ciple to turn any matrix or high order tensor into vectors
through flattening, it is not possible without significant fea-
ture engineering on top of raw features to express within the
data that there are realistic limitations in the search space
of how input features could possibly be combined. This in
turn makes it difficult to learn features during training that generalize to runtime conditions.

A central goal of this work is to develop a model that is both sufficiently powerful to achieve superior empirical results, and restricted to reasoning that would be interpretable to end user analysts. This approach has the important benefit of ensuring that users of the model understand the logic by which results are derived, enabling them to more properly address circumstances when the model predicts unusual outcomes with confidence. To do this we assume that each observation of each factor considered can ultimately be expressed as having an additive relationship with the expected forecast. This constraint enforces an important quality of being hierarchically interpretable. This implies, for example, that the model is interpretable both on the level of providing a prediction for the expected downturn in sales because of the predicted heat wave next week, and even further down to the specifics of the expectation based on the temperature that Wednesday.

3.1. Independent Observation Multifactor Model

Consider the broad class of models where a set of \(N_f\) observations are hierarchically attributed among a set of factors \(F\) in the previous period of relevant time \(P_f\) that are assumed to account for the difference between a baseline forecast and the true signal. Each observation \(i\) for factor \(f\) at time instant \(\tau\), \(x_{if}(\tau)\) is assumed to have an independent effect \(y_{if}(\tau)\) in modifying the baseline forecast \(B(\tau)\) to produce prediction \(p(\tau)\).

\[
y_{if}(\tau) = G(x_{if}(\tau)) \tag{1}
\]

\[
p(\tau) = B(\tau) + \sum_{f} \sum_{P_f} \sum_{N_f} y_{if}(\tau) \tag{2}
\]

This independent observation model in equations 1 and 2 ensures that the additive effect of each observation of each factor can be treated as independent and thus analyzed for all factors, at the granularity of a single observation, and for any potentially interesting subset of observations and factors (simply by adding up the effects of the observations in the subset). Although observations are treated as having an independent impact on the forecast, there is no restriction that the factors be viewed in isolation or without proper context, providing the model with sufficient power through the function \(G\) to express complex interactions and correlations. This form extends ideas in Generalized Additive Models (Hastie & Tibshirani, 1990) to functions over matrices and bias vectors respectively.

Our notation in this paper is that \(W\) and \(b\) refer to learned matrices and bias vectors respectively.

3.2. Simple Neural Network Independent Observation Model

Let us now consider a straightforward extension of the above independent observation model to utilize neural networks trained end to end in a supervised fashion.

A first distinction we will make is that it is generally insufficient to analyze the raw signal \(r_{if}\) of an observation in isolation, so we formalize that the observation input also includes a concatenation with a vector that represents the context.

\[
x_{if}(\tau) = \text{concatenate}(r_{if}(\tau), \text{context}_{if}(\tau)) \tag{3}
\]

As an example, it is impossible to figure out if a 50°F temperature in Ohio is relatively hot or cold without knowing both the time of the year, and the recent weather trends in the region. In our experiments, we consider a context vector that consists of a 107 dimensional one hot vector representing which store the prediction is for, a 4 dimensional vector representing the season and percent progress through that season, and computed differences between the observation in question and the average observation over both a one week and one month history. The use of differences with average values as opposed to a full sequence of values may seem like feature engineering, which we try to avoid wherever possible in our models. We actually also considered a recurrent neural network model over the entire sequence instead, but saw no increase in accuracy with a large increase in computation time. Manually specifying the comparative contexts to look over for each factor is an extremely minimal one-time human burden (which we set fixed at 1 week and 1 month for all factors) that is well worth the increased computational efficiency over data that has minimal meaning.

Armed with a more powerful expression of the observation, we can now apply a neural network paradigm to develop a formulation of \(G\), which we detail below for a neural network with a single hidden layer of dimension \(D\).

\[
h_{if}(\tau) = \tanh(W_{hf}x_{if}(\tau) + b_{hf}) \tag{4}
\]

\[
y_{if}(\tau) = \tanh(W_{ff}h_{if}(\tau) + b_{ff}) \tag{5}
\]

3.3. Soft Attention over Multifactor Models

As opposed to hard attention, we focus on soft attention methods in this work to ensure that all input features have
been given consideration at prediction time. A straightforward implementation of a soft attention mechanism for our independent observation model can be achieved with the following system of equations:

\[ m_{if}(\tau) = \text{sigmoid}(W_{mf}h_{if}(\tau) + b_{mf}) \]  

\[ d_{if}(\tau) = \text{tanh}(W_{df}h_{if}(\tau) + b_{df}) \]  

\[ a_{if}(\tau) = \frac{m_{if}(\tau)}{\sum_f \sum_{\tau} \sum_i m_{if}(\tau)} \]  

\[ y_{if}(\tau) = a_{if}(\tau)d_{if}(\tau) \]  

We model our attention mechanism after the soft attention model proposed in (Chorowski et al., 2015). Intuitively, \( d_{if} \) can be interpreted as determining the relative directional impact of the observation, and \( a_{if} \) can be interpreted as modulating the amplitude of its impact in the context of the other observations and factors. We can see the soft attention mechanism as recreating the general logic a human would follow if asked to do the same problem. First, each observation is considered in isolation and \( m_{if} \) is determined as a measure of how interesting or unusual the observation is. Next, all of the observations are considered in context and a small subset is picked that are most likely to have influence on the forecast. Finally, the individual impact of each important observation given its importance in the context is assessed and added together to determine a prediction.

### 3.4. Promoting Sparse Attention

As observed in section 3.3, intuitively only a relatively small subset of observations and factors should realistically be considered to influence the prediction at a given time step. There for, we introduce a new L1 regularizarion over the importance for all observations in our loss function. We illustrate an example of this loss function for the case of minimizing mean squared error, target \( t \), and \( m_{if} \) constrained to the always positive range of 0 to 1 by the sigmoid function:

\[ \text{Loss}(\tau) = (t(\tau) - p(\tau))^2 + \beta \sum_f \sum_{\tau} \sum_i m_{if}(\tau) \]  

Here \( \beta \) is a regularization parameter representing the coefficient of the attention regularization. We made the choice of using a mean squared error loss function in our experiments, but other functions may have advantages in some problems. If the attention units are not constrained to be positive, the absolute value of \( m_{if}(\tau) \) should instead be considered in equation 10.

### 3.5. Addressing Unexplained Factors

One clear issue with the initial formulation of the independent observation model in section 3.1 is that it implicitly assumes that the differences between the baseline forecast \( B \) and the actual targets \( t \) can be accounted for entirely by the external factor observations. In actuality, it is quite possible that only a small percentage of the difference can be explained by the current set of external factors. In this case, even our sparse attention model will be very inclined to over fit on the training data in a non-generalizable and non-interpretable attempt to account for the bulk of the error between the target and baseline signals. We attempt to combat this tendency by allowing our model to modify our baseline forecast in time periods of high uncertainty. We achieve this by introducing a simple attention mechanism that balances our baseline forecast at the current time step \( B(\tau) \) with the actual value at the last time step \( L(\tau) \). Moreover, the following system of equations shows how it integrates with our soft attention mechanism over observations and factors:

\[ g(\tau) = \text{concatenate}(u(\tau), \text{context}(\tau)) \]  

\[ m_B(\tau) = \text{sigmoid}(W_{mB}g(\tau) + b_{mB}) \]  

\[ m_L(\tau) = \text{sigmoid}(W_{mL}g(\tau) + b_{mL}) \]  

\[ m_{total}(\tau) = m_L(\tau) + m_B(\tau) + \sum_f \sum_{\tau} \sum_i m_{if}(\tau) \]  

\[ a_L(\tau) = \frac{m_{L}(\tau)}{m_{total}(\tau)} \]  

\[ a_B(\tau) = \frac{m_{B}(\tau)}{m_{total}(\tau)} \]  

\[ a_{if}(\tau) = \frac{m_{if}(\tau)}{m_{total}(\tau)} \]  

\[ p(\tau) = a_L(\tau)L(\tau) + a_B(\tau)B(\tau) + \sum_f \sum_{\tau} \sum_i y_{if}(\tau) \]
Here $m_{ij}$ estimations for the observations are the same as before. For the case of the baseline $m_{B}$ and the last value $m_L$, we utilize a concatenation of a vector representing the local uncertainty $u$ and a context vector. In our experiments, the uncertainty vector $u$ was fixed to a vector represented by $[B(\tau), L(\tau), B(\tau) - L(\tau), B(\tau - 1) - L(\tau)]$ at time instant $\tau$ and the context vector consisted of the concatenation of a store vector and seasonal vector as described in section 3.2. As detailed in the above equations, we combine consideration of the attention over the uncertainty with the attention over the observations of external factors. This allows the model to dampen focus on the last value in times of large uncertainty when there are in fact highly interesting observations in external factors, and allows for attention on external factors to fall to zeros (as opposed to being fixed at a total of 1) when the modified baseline accurately models the target values. The process flow of the model described above is illustrated in Figure 2.

$$\text{Loss}(\tau) = (t(\tau) - p(\tau))^2 + \beta m_{\text{total}}(\tau)$$  (19)

Here $m_{ij}$ estimations for the observations are the same as before. For the case of the baseline $m_{B}$ and the last value $m_L$, we utilize a concatenation of a vector representing the local uncertainty $u$ and a context vector. In our experiments, the uncertainty vector $u$ was fixed to a vector represented by $[B(\tau), L(\tau), B(\tau) - L(\tau), B(\tau - 1) - L(\tau)]$ at time instant $\tau$ and the context vector consisted of the concatenation of a store vector and seasonal vector as described in section 3.2. As detailed in the above equations, we combine consideration of the attention over the uncertainty with the attention over the observations of external factors. This allows the model to dampen focus on the last value in times of large uncertainty when there are in fact highly interesting observations in external factors, and allows for attention on external factors to fall to zeros (as opposed to being fixed at a total of 1) when the modified baseline accurately models the target values. The process flow of the model described above is illustrated in Figure 2.

3.6. Regularization Considerations

We found that even with sparse attention, it is possible for our model to over fit and learn co-adapted representations of factors that can effectively combine to produce a helpful but not directly interpretable offset. To prevent our model from learning these co-adapted representations, we draw inspiration from the dropout regularization technique of (Hinton et al., 2012). Intuitively, if we do not allow external factors to rely on the presence of one another we can promote independence of the learned representations. We achieve this during training by drawing a sample from a binomial distribution for each factor and converting to a binary representation with a success threshold determined by a grid search from 0.1 to 0.9 utilizing the validation set during training. We also do not want observations within a factor like different weather indicators, local events, and national events to become co-adapted and enforce dropout of the input observations as well. We see significant gains in generalization performance using this approach. We also experimented with multiplying through by the dropout factor as done in (Hinton et al., 2012) during test time, and got very similar performance.

4. Experiments

We conducted our experiments utilizing two years of transaction data from 107 stores and 20 commodity classes of one of the largest retailers in America in the state of Ohio spanning May 2012 to May 2014.

We leveraged historical weather information from 16 stations in the region and considered the weather at each store to be equal to the weather of the closest station by distance. We conducted experiments using the General Description field (including over 100 different descriptive categories), the feels like temperature, the wind speed, the visibility, the relative humidity, and the UV Description. We collected daily minimum, maximum, and mean values for numerical factors and a daily average of the one hot vectors representing the hourly values for categorical factors.

For local event information we analyzed metadata about regional events that were registered a week in advance on Eventful.com. This included 301,327 local events over the two year span. Each event includes two descriptive fields with 29 categories, a distance from the store in consideration, and a popularity. One major limitation of the Eventful data is that only a few examples in our entire dataset had a popularity that was not null, so we only have a direct measurement of the expected turnout in extreme cases.

In our experiments we also collected a random 10% of all English tweets on Twitter over the two year period. We used this data to consider 51 national events in our experiments. In addition to national holidays, observances, and events like the Super Bowl or Oscars, we included what we considered to be a secondary set of yearly event signals including Hummus day and Pie day. We also used Twitter data to mine trends in social chatter about each commodity we consider and develop a word match based extractor leveraging a list words related to the commodity.

4.1. Design of Baseline Forecast

Holt-Winters or ARIMA models require a minimum of two observation periods worth of data in order to provide an initial fit (in our case, two years worth of data). Consequently, we developed a model that works reasonably well
in forecasting the demand based only on univariate transaction data without this requirement. It also follows core logic and principles shared with Holt-Winters and ARIMA models.

The baseline model discussed here works by decomposing the signal into multiple components that represent the internal factors towards the forecasts: level, trend, and the seasonal (periodic) components. The level component represents the constant demand value over the entire time period. The trend component represents the linearly increasing demand over time. The seasonal (periodic) components correspond to the periodic increase and decrease in the sales values due to seasonal demands. More precisely, if the true sales at time instant $\tau$ is $y(\tau)$, then the baseline model assumes that this signal value is generated as follows:

$$y(\tau) = l(\tau) + t(\tau) + \sum_{p=1}^{P} s_p(\tau) + e(\tau)$$  \hspace{1cm} (20)

where $l(\tau) = L$ is the level component, $t(\tau) = \tau T$ is the trend component, $s_p(\tau) = s_p(\tau - \tau_p)$ is the seasonal, $e(\tau)$ represents the anomalies attributed to the sparse unexplainable external factors, $L$ is the constant level value, $T$ is the linear trend value, $P$ is the number of periodic components, and $\tau_p$ is the unknown period of the $p$-th periodic component. The baseline forecasting method at time instant $\tau$ uses all the data available until time $\tau - 1$ to estimate the level, trend and seasonal components of the decomposition, with the resulting $e(\tau)$ capturing the residual anomalies. Once the parameters are estimated, the forecast prediction for the next time instant is given by

$$b(\tau) = \hat{L}^{\tau-1} + \hat{T}^{\tau-1} + \sum_{p=1}^{P} \hat{s}_p^{\tau-1}(\tau)$$  \hspace{1cm} (21)

where $\hat{L}^{\tau-1}$, $\hat{T}^{\tau-1}$, and $\hat{s}_p^{\tau-1}$, are the estimates of $L$, $T$, and $s_p$ respectively based on data observed until time $\tau - 1$.

From the above discussion, we can see that the prediction depends on the estimates of the unknown parameters. To estimate these values, we implemented a 2-step process that involves Fourier based synthesis and sparse regression (Tibshirani, 1996).

4.2. Input Features

Our independent observation model discussed in section 3.1, is capable of reasoning about data in its natural hierarchy. In our experiments each of the 6 weather signals discussed earlier is considered its own distinct factor that includes a sequence of daily observations. Local events are represented as a matrix detailing a sequence of observations for the upcoming events slotted for the next week. National events are also a matrix representing a 51 length signal detailing patterns in the social chatter about each event present a week in advance. We have confirmed that chatter heights always correspond with the actual week of the national event. The commodity social signal is represented as a single observation vector including analysis of mentions, sentiment, and intent to buy with the context available a week in advance. In all of our experiments the true sales value, demand forecast, and last sales value are normalized by subtracting the mean sales value for the store and dividing by 10 times the standard deviation.

Many existing regression models that we would like to compare our method against could not handle input of the format described in the previous paragraph. As such, we did some feature engineering to compress these observations down to a single vector that regressors can use for prediction. For each weather observation we took a weekly average to project down to a vector. For local events we computed a sum weighted by the popularity of each observation to compress down to a single vector. The national events were just considered as a flattened version of the matrix. The rest of the features were considered without modification and concatenated together. Without PCA our flattened vector contained 3,139 elements.

4.3. Training Details

In all of our experiments we used the same 93 stores for training and 14 stores for validation. Our neural network models were all trained with Stochastic Gradient Descent (SGD) until convergence on the validation set. Hyperparameters are selected based on a grid search over the validation set. We ensure that the baseline neural network has potential access to 5 times as many total parameters than our model during training to ensure that our superior performance is not simply about the quantity of parameters. In practice none of our neural network models find it useful to have large hidden sizes and are generally optimal between 10 and 100 units. Additionally, we determined the optimal PCA compression dimension for generalization by testing training set based accuracy on the validation set for generalization. We note specifically in section 5 which models used PCA features as we did not find it useful for the other models. We train all of our models first over a year of data with full parameter tuning, and then after each passing 3 months initialize with the old model and update the model based on the updating training set (or retrain from scratch with the updated dataset when initialization is not possible). When we update, we keep the same tuned parameters determined during our initial training.

Frequent retraining is not very beneficial as most of our models have pretty time invariant learned representations for the influence of external factors, but we showcase every 3 months so even the simpler models reach their optimal update frequency. Our shared baseline forecast model is re-
trained weekly. However, it is likely not necessary to train the baseline model at that frequency.

5. Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Features</th>
<th>MAPE</th>
<th>Anomaly %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline Forecast</td>
<td>FV</td>
<td>26.79</td>
<td>7.13</td>
</tr>
<tr>
<td>Our Model</td>
<td>IO</td>
<td>20.40</td>
<td>5.11</td>
</tr>
<tr>
<td>- Attention Sparsity</td>
<td>IO</td>
<td>23.69</td>
<td>5.65</td>
</tr>
<tr>
<td>- Soft Attention</td>
<td>IO</td>
<td>33.49</td>
<td>11.05</td>
</tr>
<tr>
<td>Our Gating Network</td>
<td>IO</td>
<td>24.98</td>
<td>6.74</td>
</tr>
<tr>
<td>+ Attention Sparsity</td>
<td>IO</td>
<td>24.01</td>
<td>6.23</td>
</tr>
<tr>
<td>Random Forest</td>
<td>FV</td>
<td>24.87</td>
<td>5.77</td>
</tr>
<tr>
<td>Neural Network</td>
<td>FV</td>
<td>28.27</td>
<td>5.78</td>
</tr>
<tr>
<td>SVR (RBF Kernel)</td>
<td>FV</td>
<td>31.53</td>
<td>6.60</td>
</tr>
<tr>
<td></td>
<td>PV</td>
<td>31.46</td>
<td>6.60</td>
</tr>
<tr>
<td>Decision Trees</td>
<td>PV</td>
<td>34.17</td>
<td>9.62</td>
</tr>
<tr>
<td>Bayesian Regression</td>
<td>FV</td>
<td>38.74</td>
<td>14.24</td>
</tr>
<tr>
<td>Lasso Regression</td>
<td>FV</td>
<td>46.76</td>
<td>16.49</td>
</tr>
</tbody>
</table>

Table 1. Comparison of models by average forecasting percent error and the frequency of unpredicted anomalies when predicting a week in advance over one year of testing. The Baseline Forecast is an input given to each model.

<table>
<thead>
<tr>
<th>Model</th>
<th>Features</th>
<th>MAPE</th>
<th>Anomaly %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline Forecast</td>
<td>FV</td>
<td>50.76</td>
<td>16.77</td>
</tr>
<tr>
<td>Our Model</td>
<td>IO</td>
<td>34.87</td>
<td>11.56</td>
</tr>
<tr>
<td>- Attention Sparsity</td>
<td>IO</td>
<td>38.66</td>
<td>12.03</td>
</tr>
<tr>
<td>- Soft Attention</td>
<td>IO</td>
<td>55.72</td>
<td>25.98</td>
</tr>
<tr>
<td>Our Gating Network</td>
<td>IO</td>
<td>38.29</td>
<td>12.95</td>
</tr>
<tr>
<td>+ Attention Sparsity</td>
<td>IO</td>
<td>35.89</td>
<td>11.69</td>
</tr>
<tr>
<td>Random Forest</td>
<td>FV</td>
<td>43.14</td>
<td>13.14</td>
</tr>
<tr>
<td>Neural Network</td>
<td>FV</td>
<td>49.42</td>
<td>12.86</td>
</tr>
<tr>
<td>Decision Trees</td>
<td>PV</td>
<td>52.91</td>
<td>19.89</td>
</tr>
<tr>
<td>SVR (RBF Kernel)</td>
<td>PV</td>
<td>61.60</td>
<td>19.58</td>
</tr>
<tr>
<td></td>
<td>PV</td>
<td>61.50</td>
<td>19.58</td>
</tr>
<tr>
<td>Bayesian Regression</td>
<td>FV</td>
<td>74.87</td>
<td>31.20</td>
</tr>
<tr>
<td>Lasso Regression</td>
<td>FV</td>
<td>89.67</td>
<td>34.48</td>
</tr>
</tbody>
</table>

Table 2. Comparison of models on the 5 hardest commodities for the Baseline Forecast to model. We report average forecasting percent error and the frequency of unpredicted anomalies when predicting a week in advance over one year of testing.

Table 1 and Table 2 describe the main results of our experiments. For each model, we describe the features used where FV refers to the feature vector explained in section 4.2, and IO refers to a PCA compression of the feature vector. IO refers to the Independent Observation Model’s feature representation as described in section 3.1 and section 4.2. MAPE represents the Mean Absolute Percent Error. The Anomaly % is defined as the percent of weeks considered where there was either an oversell or an undersell. We use an industry rule of thumb in which the prediction being at least two times smaller than the actual sales constitutes an oversell, and the prediction being at least two times bigger than the actual sales constitutes an undersell.

5.1. Comparison To Other Models

Our model ultimately accounts for a 23.9% relative improvement and 28.3% reduction in the frequency of apparent anomalies over the baseline forecast. The surprising aspect is that none of the group of Lasso Regression, Bayesian Ridge Regression, Support Vector Regression, and Decision Tree alternatives are able to surpass the baseline forecast on average over the year. This is so surprising because this means these models would be better off learning a representation that was just copying one element of their input than learning what they did. A neural network with L1 regularization and dropout is needed to show any value over the baseline forecast using the feature vector as input over the 20 commodities by being robust to anomalies. The Random Forest regression model is our strongest baseline that is not a neural network as it has a powerful mechanism of preventing decision trees from over fitting. However, our attention model achieves significantly superior results. In taking an average over all commodities we obscure one of the underlying stories in the data. In fact, the traditional models do surpass or equal the baseline forecast for many commodities, but tend to have a particularly hard time modeling the highly volatile commodities that have the highest baseline forecast errors detailed in Table 2. For these commodities, many of the baseline models over fit significantly. Our proposed multifactor attention approach, however, generalizes extremely well to the year of test data, making the forecasts 31% better. Many of the other algorithms have a hard time teasing out real signal from this noise and produce huge errors on the testing set.

Table 1 and Table 2 also showcase the critical importance of the comparative attention mechanism to the success of our model. The L1 regularization of attention seems to improve generalization quite consistently as well. Without an attention mechanism of some kind it does not seem possible to constructively leverage the more natural semantics associated with the independent observation model. Our neural network initially loses performance across the board by working with the more complex structure. However, with the incorporation of attention we utilize this rich structure in a generalizable way without the additional feature engineering for each source that would clearly be needed to get reasonable performance from many of the baseline models tested. Moreover, we validate that although the Gating Network of a HME model can solve the same problem fairly effectively, it performs significantly worse than our model with attention based off the hidden representation. These
results seem to also suggest that our proposed sparse attention paradigm can improve certain incarnations of HME models when the data is volatile.

5.2. Analysis of the Influence of Factors

In Table 3 we show the amount of impact each major group of observations had in influencing the forecast over the data set. Although we considered quite a few data sources, it is unsurprising that a large chunk of the error is still considered unaccounted for (for example promotional information is not present in our experiments). We would naturally expect weather to be highly impactful on retail demand. Social chatter on Twitter about the commodity is also an important and frequently used indicator. It is also perhaps unsurprising that the various event sources would have a low average impact because they are occasional by their nature. It is hard to know to what extent the lack of a reliable popularity metric impacted the usefulness of the local event data. We can see that our model achieves bigger gains on the more volatile commodities because it finds more occasions where it is useful to correct the signal based on weather and social chatter in this data.

<table>
<thead>
<tr>
<th>Component</th>
<th>All 20</th>
<th>Hardest 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unexplained Correction</td>
<td>54.7%</td>
<td>48.7%</td>
</tr>
<tr>
<td>Weather</td>
<td>22.8%</td>
<td>26.1%</td>
</tr>
<tr>
<td>Commodity Social Signal</td>
<td>18.6%</td>
<td>22.4%</td>
</tr>
<tr>
<td>National Events</td>
<td>2.4%</td>
<td>1.1%</td>
</tr>
<tr>
<td>Local Events</td>
<td>1.5%</td>
<td>1.6%</td>
</tr>
</tbody>
</table>

Table 3. Relative total contribution of each group of observations to the prediction of all 20 commodities and the hardest 5 commodities to model over 2 years of data.

6. Discussion

6.1. Intuition about “Noisy” Data

In our experiments, attention-based neural networks perform significantly better than standard neural networks. However, the bulk of these gains came on the five most volatile commodities as shown in Table 2. These commodities are noisy in that their sales are highly volatile, with little training data, and thousands of possible explanatory features to consider. Our intuition is that attention-based neural networks should play a role in combating this noisy data problem, especially with the imposed sparsity that should push many attention values near zero early in training. The sparse attention mechanism forces entire observation vectors to have zero influence on the prediction—effectively shrinking the number of explanatory variables considered by the model at that point. At times, a small number of values in an observation vector may by chance have a high correlation with the volatility in the signal over a small period and this becomes more probable as volatility increases. The attention mechanism makes a holistic judgment based on a group of features to dismiss the entire group and shield the model from reacting to spurious correlations in a small subset of the observation vector. Our experiments seem to support this hypothesis, but a more rigorous theoretical analysis of the properties of this model will be left to future work.

6.2. An End to End Model

To this point, our focus has been on a neural network module that corrects an existing time series signal with no sharing of the latent parameters used for time series prediction. However, it is of theoretical interest whether or not it is possible to train this model end to end with a neural network that is also responsible for the time series prediction itself. We experiment with a GRU (Cho et al., 2014) recurrent neural network with sparse regularization as our time series modeler that is sent the entire prior history of the store’s time series concatenated with a one hot store encoding at each time step. We find it useful to adjust our architecture slightly to allow for sharing of latent parameters by concatenating both the output and last hidden representation of the GRU to context (τ) for all observations of external feature groups. Additionally, in equation 12 g(τ) is replaced by the final hidden representation of the GRU. Moreover, we observe that when the model has more power over modifying the time series component itself, the uncertainty vector and unexplained factors add less value. As such, we do not compute equations 13 and 15, and remove the term over the last value in equations 14 and 18. B(τ) is also replaced by the output of the GRU. Empirically, we find this model achieves 20.28 MAPE with a 5.05 anomaly percentage. This result indicates both that our proposed multifactor attention module can be used to augment a recurrent neural network and that it can potentially surpass precision achieved with a traditional univariate system through tighter integration of prediction elements.

7. Conclusion

We have presented a novel multifactor attention model for neural networks that incorporates external data sources for time series prediction problems. The model provides evidence for the reasoning behind adjustments to the time series forecast output by leveraging a comparative attention mechanism over the external factors in an additive model. Our model achieves a 23.9% improvement of forecasts due to external data sources and helps predict 28.3% of the anomalous events. Moreover, our model offers superior descriptive capabilities in comparison to other neural networks proposed for time series forecasting to date.
Correcting Forecasts with Multifactor Neural Attention

References


Correcting Forecasts with Multifactor Neural Attention


