Volume 5, Issue 12, December 2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Implementation Challenges and Performance Comparison of KNN and DNN Algorithms for Weather Prediction

[1]Aishwarya Kalra, [2]Sagar Gupta, [3]Renu Taneja, [4]Ankur Gupta

[1, 2 4] Student, Department of IT, Bharati Vidyapeeth's College of Engineering, New Delhi, India

[3] Associate Professor, Department of IT, Bharati Vidyapeeth's College of Engineering, New Delhi, India

Abstract–***The different approaches used for weather prediction are challenged by complex weather phenomena with limited observation and past data. Weather phenomena have many parameters that are impossible to enumerate and measure. Expert forecasters have both a general knowledge of large-scale weather systems and specific knowledge about the idiosyncratic behaviour of local scale weather phenomena. The types of forecasts commonly made by expert forecasters include terminal aerodrome forecasts (TAFs), public forecasts and marine forecasts. TAFs are the most accurate. In this research paper, we have mainly described the two approaches for weather prediction phenomenon that are KNN and DNN algorithms, discussed their implementation challenges and compared their performances.***

Keywords – TAF (terminal aerodrome forecasts), KNN (K-Nearest Neighbour) Algorithm, DNN (Deep Neural Networks) Algorithm

I. INTRODUCTION

Weather is one of the most effective environmental constraints in every phase of our lives. We are subject to adjusting ourselves with respect to weather condition from our dressing habits to strategic organizational planning activities, since the adverse weather conditions may cause a considerable damage on our lives and properties. We need to be on alert to these adverse weather conditions by taking some precautions and using prediction mechanisms for early warning of hazardous weather phenomena.

Weather prediction is an indispensable requirement for all of us. There is a general and increasing interest on weather information, since every day we habitually give an ear to weather forecast news for local and large-scale long-term or short-term weather predictions. Leading weather research institutions and companies have been developing weather prediction systems capable of detecting, predicting and forecasting weather phenomena and hazards by utilizing state-of-the-science technologies. Thus weather prediction utilizations fields and prediction accuracy increases monotonically by the time.

We made this research in parallel with our major project about a local airbase short-term weather prediction implementation with KNN algorithm, DNN algorithm and their performance comparisons. That is why our paper mainly focuses on local and short-term predictions when covering all literatures about weather prediction expert systems.

II. WEATHER PREDICTION PROBLEMS

Weather prediction is a complicated procedure that includes multiple specialized fields of expertise. "There are only two methods to predict weather: the empirical approach and the dynamical approach" (Lorenz 19). Lorenz thus separated weather forecasting methodologies into two main branches in terms of numerical modelling and scientific processing (Artificial Intelligence) of meteorological data.

Hansen explains this classification as follows; the empirical approach is based upon the happenings of comparable cases (i.e., similar weather situations). The empirical approach is more useful for the prediction of local-scale weather if the recorded cases are available in required number. (e.g., cloud ceiling and visibility in a few square kilo-meters around an airport). The dynamical approach is generally referred to as computer modelling. It is based upon equations of the atmosphere. The dynamical approach is only useful for modelling large-scale weather phenomena, for an instance, general wind direction over a few thousand square kilo-meters.

The dynamical approach is a NWP (numerical weather prediction) method that gathers inputs readings from various sources like weather stations, weather buoys, satellite images, atmospheric probes and other sources and computes solar energy effects and global rotational effects, latitude effects, ocean currents effects, landmasses effects and produces global weather predictions. NWP models build these computation results as initial conditions to systems of equations that describe the atmosphere. These models are run statically and produce forecasts on a six hourly basis that serves as the foundation of all forecasts. Output from these models is useful for large scale and longer-term forecasts. However, the complete description of the atmosphere in this form requires more improvements, more measurement, and more parameters and is far beyond current capability. These systems are not able to produces results that satisfies everybody in local and short-term cases, since, the accuracy of the models is inherently dependent upon the incompletely inherited initial conditions.

The empirical approach mostly depends on past meteorological data with the assumption that initially similar conditioned weather conditions shall go in parallel for some time enough to make a short range weather prediction.

Huschke defines a "persistence forecast" as "a forecast that the future weather conditions will be the same as the present conditions." This approach is an analogical method of weather forecasting that bases predictions for the present case on the outcomes of similar past cases. Utilizing past meteorological data is not as easy as it seems, since there are many problems to be handled, like deciding similar cases in continuous weather parameters. Here, we are comparing the performances of the two solutions for this, CBR KNN algorithm and DNN algorithm.

III. TERMINAL AERODROME FORECAST

TAFs are required to be most precise both in terms of measurable weather conditions and in terms of timing. TAF forecasts of the height of low cloud ceiling are expected to be accurate to within 100 feet; forecasts of the horizontal visibility on the ground, when there is dense obstruction to visibility, such as fog or snow, are expected to be accurate to within 400 meters and forecasts of the time of change from one flying category to another are expected to be accurate to within one hour.

Airports organize air traffic mainly based on local weather condition that has agile parameters that may change in a short time. These parameters such as horizontal visibility and vertical visibility (ceiling), fogginess, precipitation level etc. can be very serious threat for flight safety and cost. When ceiling and visibility at a busy airport are low, in order to maximize safety, the rate of planes landing is reduced. When ceiling and visibility at a destination airport are forecast to be low at a flight's scheduled arrival time, its departure may be delayed in order to minimize traffic congestion and related costs. For Airport, weather forecast timeliness and correctness are the main issues.

IV. WEATHER PREDICTION USING KNN ALGORITHM

K-Nearest Neighbour (KNN) is a simplest machine learning method. The choice of KNN is motivated because of its simplicity and flexibility to incorporate different data types. The main idea behind KNN is to base an estimation on a fixed number of observations, let's say k, which are closest to the desired output. It does not build a model or function previously, but yields the closest k records of the training data set that have the highest similarity to the test therefore, is often considered as a lazy learning algorithm. KNN can be used both in discrete and continuous decision making. Discrete decision making is known as classification whereas continuous decision making is known as regression. For the classification, we select most frequent neighbour, and for the regression, we calculate the average of k neighbour. KNN is a supervised learning algorithm i.e. a training set is given consisting of n pair (x_i, y_i) and the problem is to estimate y(x) from a new input x. In order to apply this technique, it is necessary to have a training set and a test sample, to know the value of k (how many neighbours are used for prediction) and the mathematical formula of the distance calculated between the instances.

4.1. Distance Used In KNN

The three famous distances used in KNN are –

i) Euclidean Distance - $\sqrt{\sum_{i=1}^{k} (x_i - y_i)^2}$
ii) Manhattan Distance - $\sum_{i=1}^{k} |x_i - y_i|$
iii) Minkowski Distance – $(\sum_{i=1}^{k} (x_i - y_i)^q)^{1/q}$

4.2. Features Of KNN

The various important features of KNN are:

i) It is non-parametric in nature.
ii) It is simple to implement.
iii) It is considered as a lazy learning.
iv) It is robust with small error ratio.
v) It is an instance based learning
vi) It is a local learner
vii) It has uniform feature weighting.
vii) It can work with relatively little information.

4.3. Choice Of K In KNN

In order to avoid ties k is preferably odd. If the value of k is smaller then there is higher variance. On the other hand if the value of k is larger, then there is higher bias. Thus the proper choice of k depends on the data.

4.4. Problems Associated With KNN

The standard KNN method suffer from the curse of dimensionality i.e. the neighbourhood of a given point become very sparse in a high dimensional space, resulting in high variance. Thus in high dimensional, the nearest becomes meaningless.

Another problem is over-fitting. It occurs when a learning algorithm performs too good on the training set, compared to its true performance on unseen test data. The most popular method to overcome over-fitting is known as hold-set method.

According to hold-set method:

(i) Randomly choose 30% of the training data set, set it aside.
(ii) Train a classifier with the remaining 70% training data.
(iii) Test the classifier's accuracy on the 30%

4.5. Algorithm

```
for all test example x do
        for all training example ($x_i$,$y_i$), do
compute distance(x,$x_i$ );
        end for
select the k-nearest neighbour of x;
return the average output value among neighbours i.e. 1/k ;
end for;
```

4.6. KNN Implementation

Let us use an example to show how KNN is used in weather prediction. Let us take humidity for example. Humidity is an important metric used in forecasting weather. A little knowledge of amount of humidity can prevent the aforesaid adverse effects for instances; moisture may increase the conductivity of permeable insulator of electronic devices which leads to malfunctioning.

We have used three main parameters that affect highest humidity. These parameters include minimum temperature, maximum temperature and lower humidity. In this, we have used the value of the previous day of these parameters in order to predict the value of the highest humidity of next day.

This data is obtained from the weather section of the local newspaper. The KNN algorithm is trained using the data of previous 10 days and the data of next 7 days are used for testing purpose.

- **Experimental Result**

The following two figures show the experimental result. In figure (a) the running software is predicting the value of highest Humidity. In figure (b) there is a graph between the actual and predicted value of highest humidity. It is found that at the value k = 5, the prediction is very nearer to the actual result as shown in figure (a) and figure(b).

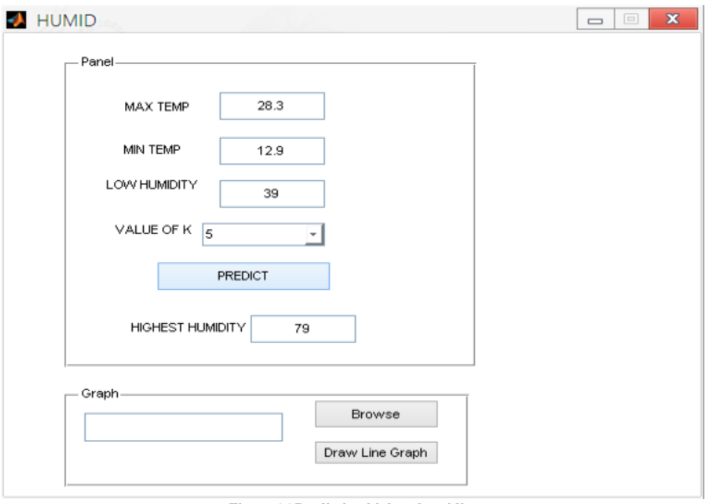
HUMID
Panel
MAX TEMP 28.3
MIN TEMP 12.9
LOW HUMIDITY 39
VALUE OF K 5
PREDICT
HIGHEST HUMIDITY 79
Graph
Browse
Draw Line Graph

Figure (a)Predicting highest humidity

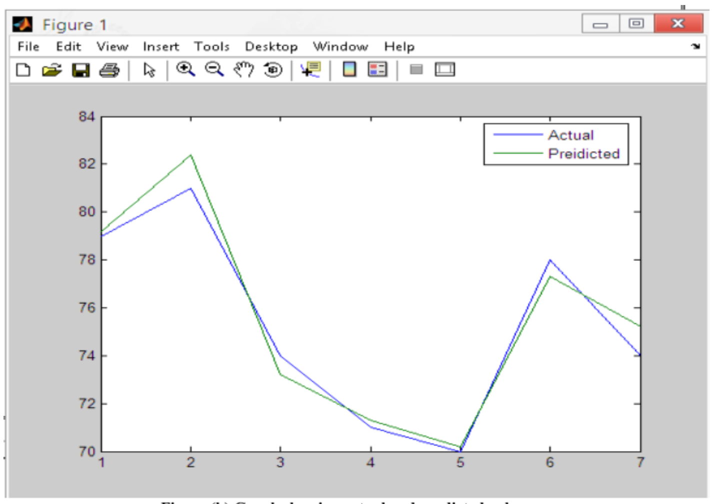

Figure (b) Graph showing actual and predicted value

The Mean Square Error (MSE) is the arithmetic mean of the sum of the square of prediction error. This error measure is popular and corrects the cancelling out effects.

$e_i = | f_i - y_i |$

f_i = prediction

y_i = true value

The MSE index ranges from 0 to ∞, with 0 corresponding to the ideal. Lower MSE is better. In this case MSE is 0.671429.

V. WEATHER PREDICTION USING DNN ALGORITHM

Deep Neural Networks (DNNs) or Deep Learning is the common term for the series of multi-layer architecture neural networks that are trained with the greedy layer-wise unsupervised pre-training algorithms[1], [2], [3]. Albeit controversial, DNNs have won great success in some fields including Computer Vision, Speech Recognition, Natural Linguistic Programming and Bio-Information processing. , DNNs can reconstruct the original raw data set by applying the greedy layer-wise unsupervised pre-training mechanisms. In other words, instead of selecting features manually that we did traditionally, DNNs can learn features with Neural Networks (NNs). And the intelligence models, like classier or regressor usually can obtain higher accuracy and better generalization with the learned features.

As its name suggested, DNN is a kind of NNs that structured by multiple layers. The word "Deep" indicates that such NN contains more layers than the "shallow" ones, which mainly includes the most widely used three-layer Feed Forward NNs in the past 30 years. Actually, multilayer NN is not a new conception, some earlier been conducted since 1990s, but the successful implementation of multi-layer NNs was not realized until the provision of the novel layer-wise unsupervised pre-training mechanism by Hinton in 2006 that is employed to solve the training difficulties efficiently.

People never cease their efforts on predicting the trend of weather changes since, the changes of climate that could impact people's daily life. Unlike data sets in other domain, weather data has some particularities. Specifically, there is season-to-season, and year-to-year variability in the trend of weather data. The cycle could be multi-month, multi-season or multi-year, and the main difficulty of investigations is to capture all the possible cycles.

Many significant research efforts are utilized to develop weather forecasting methods including artificial intelligence technologies that have been accepted as appropriate means for weather forecasting and reported encouraging results since 1980s [12]. Among many different intelligence models, single variable time series regression is the most fundamental and most widely applied one in weather forecasting, especially short-term predictions.

Since our initial investigation is an exploration on the application of DNN in the area of weather forecasting, in this paper, we mainly concentrate on employing DNN to represent the feature space for single variable time series regression problem.

Generally speaking, for a certain variable, the objective of single variable time series regression is to find the relationship between its status in a certain future time point and its status in a series of past time points, and estimates its future status via: $v_t = f(v_{t-1}, v_{t-2}, \ldots, v_{t-n})$

The essential challenge in training deep architectures is to deal with the strong dependencies that exist during training between the parameters across layers [15]. Multi-layer NN shave more parameters than shallow NNs. Moreover, in a multi-layer NN, due to the non-convexity of the complex model, the optimization with traditional BP training approach may fall in a local minimum rather than global minimum. This may bring poor generalization to the model.

This problem isn't well solved until Hinton et al. introduced Deep Belief Network (DBN) that greedily trained up one layer with a Restricted Boltzmann Machine (RBM) at a time in 2006. Shortly after, strategies for building deep architectures from related variants were proposed by Bengio and Ranzato. They solved the training problem of deep NN in two phases: in the first phase, unsupervised pre-training, all layers are initialized using this layer-wise unsupervised learning signal; in the second phase, fine-tuning, a global training criterion (a prediction error, using labels in the case of a supervised task) is minimized. Such training approach is called the Greedy Layer-wise Unsupervised Pre-training.

Fig.1 [15] shows the comparison among different training methods for NNs with deep architectures. There is a family of training models categorized into the family of Greedy Layer-wise Unsupervised Pre-training approaches. In our investigation, with the consideration of the attribute type of the weather data, i.e., the collected data are all real numbers, we choose the Stacked Auto-Encoder to build the deep architecture of our NN model.

The Stacked Auto-Encoder, as its name suggested, is a stacked architecture NN that applies Auto-Encoder in each layer. In NN, a single "neuron" is a computational unit that taken as input vector $x = x_1, x_2, \ldots, x_n$ (and a+1 intercept term), and outputs $h_{W,b}(x) = f(W^T x) = f(\sum^3_{i=1} W_i x_i + b)$ with a nonlinear function f : ℜ覫→ℜ. W is the weight matrix that stands for the connection among different neurons in the network. In most of cases, sigmoid function f(z)= 1/1+exp(−z) is employed. Atypical Auto-Encoder tries to learn a function $h_{W,b}(x)) \approx x$. In other words, it is trying to learn an approximation to the identity function, so as to output ˆx that is similar to x. The identity function seems a typically trivial function trying to learn; but by placing constraints on the network, such as by limiting the number of hidden units, we can discover interesting structure about the data, e.g., for a data set, suppose that the original samples are collected from a 100-dimensional feature space ,i.e. $x \in \Re^{100}$, set that there are50 hidden units in the hidden layer, based on the requirement $h_{W,b}(x) \approx x$, the network is forced to learn a compressed representation of the input. That is, given only the vector of hidden unit activation $sa^{(2)} \in \Re^{50}$, it must try to reconstruct the 100-dimensionalinput x. An illustration of Auto-Encoder is shown in Fig.2.If the inputs were completely random, each x_i comes from an I.I.D. Gaussian independent of the other features, then this compression task would be very difficult. But if there is a certain structure hidden in the data, for example, if some of the input features are correlated, such as in the feature space of time series analysis, then this algorithm will be able to discover some of those correlations.

The loss function of Auto-Encoder is:

$$J(W,b) = \left[\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}\left\|h_{W,b}(x^{(i)}) - x^{(i)}\right\|^2\right)\right] + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}\left(W_{ji}^{(l)}\right)^2$$

where m is the number of training samples. The objective of the Auto-Encoder is to minimize the above equation in order to make sure that the output h W,b(x^i) can approximate the x^i as far as possible. The second term in the above equation is a regularization term (also called a weight decay term)controlled by the weight decay parameter λ that tends to decrease the magnitude of the weights, and helps preventover-fitting. We can minimize above equation by gradient descent to compute the configuration of the network.

Consequently, we need to combine Auto-Encoders layer by layer with a stacked structure to build the DNN. For each layer, we use an Auto-encoder to train the parameters in this layer, and then have these layers combined together.

Specifically, in the training process of each layer, as shown in Fig.1, the input vectors have to pass through three layers, and the vectors in hidden layers (layerL2, and for simplicity, we call the vectors in layer L2 as the transformed vectors of the initially input vectors) are representations of the input vectors and can be used to reconstruct the input vectors.

Thus, in every layer of the deep NN, the input of the current layer is the output of the previous layer, then we train the input data via an Auto-Encoder, and use the transformed vectors as the output of the current layer. Fig.4 shows the detailed mechanism of stacked Auto-Encoder based DNN.

We can see that through a DNN, the raw data can be represented in new feature spaces layer by layer. In other words, DNN can learn features from the original data sets.

Consequently, we apply any proper intelligence models with the learned features.

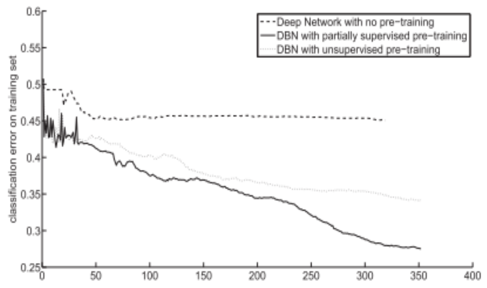

Fig. 1: Training classification error vs training iteration on DNNs, which shows the optimization difficulty for DNNs and the advantage of pre-training methods.

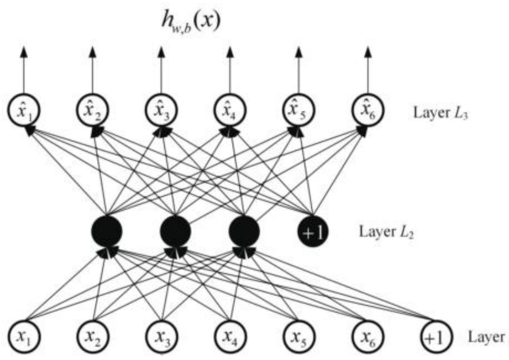

Fig. 2: An illustration of Auto-Encoder Algorithms. Layer L_1 is the input layer, and L_3 is the output layer. Via hidden layer L_2, we hope to represent the information x in layer L_1, so that the output $\hat{x}$ in L_3 can approximate the raw data x.

VI. CONCLUSION

Both KNN and DNN can be used for weather forecasting.

In our investigation, we explore an approach that using an ovel computational intelligence technology to process massive volume of weather data. The proposed DNN model may represent the features of the raw weather data layer by layer, and experimental results show that the obtained features can improve the performances of classical computational intelligence models. The contribution of our investigation is significant: we give an approach that using computational intelligence method to learn features for weather forecasting, and our experiments demonstrate that the DNN algorithm also has the potential on time series problem.

KNN is the most promising approach with successful test results. All reviewed test results are limited to past meteorological data that will represent all possible weather condition distributions. Data collecting and acquisition are initial and one of the most critical parts of expert systems computations. Automated data collecting systems must be available rather than using human manual inputs. Local data collection is very important if we want to make very precise forecasting.

REFERENCES

[1] Lorenz, E. N. (1969a) Three approaches to atmospheric predictability, Bulletin of the American Meteorological Society, 50, 345–349.

[2] Eng Int Syst (2002) 3: 139–146 © 2002 CRL Publishing Ltd "A fuzzy case-based system for weather prediction" Denis Riordan and Bjarne K Hansen Faculty of Computer Science, Dalhousie University, Scotia, Canada,

[3] R. E. Huschke (editor), Glossary of Meteorology, American Meteorological Society, Boston, Massachusetts, USA, pp 106, 419, 1959.

[4] B. J. Conway, Expert systems and weather forecasting, Meteorological Magazine, 118, pp23–30, 1989.

[5] T. Hall, Precipitation forecasting using a neural network, Weather and Forecasting, Boston, 14, Iss. 3; 338–346, Jun 1999.

[6] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," Science, vol. 313, no. 5786, pp. 504–507, 2006.

[7] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in Proceedings of the 26th Annual International Conference on Machine Learning. ACM, 2009, pp. 609–616.

[8] Y. Bengio, Learning deep architectures for AI. Now Publishers Inc.,2009, vol. 2, no. 1.

[9] X. Wang and Q. He, "Enhancing generalization capability of svm classifiers with feature weight adjustment," in Knowledge-Based Intelligent Information and Engineering Systems. Springer, 2004, pp.1037–1043.

[10] J. Schmid huber, "Curious model-building control systems," in Neural Networks, 1991.1991 IEEE International Joint Conferenceon. IEEE,1991, pp. 1458–146