Deep Learning in Data Science

Gautham Naik, Nandan Nayak, Nithesh, Nithin H A, Nagaraj Bhat, K C Gouda

ABSTRACT--- Up until early 2000's climate predictions were made mainly using statistical methods. This prediction wasn't always entirely accurate. With the introduction of deep learning in climate prediction, the prediction accuracy has improved dramatically. The sensors in the weather stations give massive amount of unstructured data. Due to the humungous amounts of sensors and data from it, it's almost impossible to compute all the necessary weather information in time. AI and deep learning help to overcome this problem using different models which can swiftly and accurately make this job simple. Accurate climate prediction is very important to predict is very important to predict any natural calamities or unexpected change in weather. This report highlights few of the deep learning models which can be used for climate prediction by scientists. This paper only takes scratches the surface of the capabilities of AI in climate change. More advancements in this field would lead to better simulations of the weather conditions which can then be useful to predict the extreme weather conditions accurately. Few of the authors have used unique models in their prediction of various temperature, rainfall, pollution levels etc. which have helped them to find the discrepancies in the climate if any.

Index Terms — Deep learning, climate, prediction.

I. INTRODUCTION

The paper describes the various deep learning techniques used in climate prediction which includes ANN, CNN, decision tree which are the major deep learning techniques. Machine learning has many uses in different fields. With the promising results of deep learning in different fields, researchers have used this in climate prediction. With the increase of global warming, it is now more important than ever to predict the accurate climate of a region from the given data and take precautionary measures. Some models give good results even when resolution of the image is poor. A huge amount of data is produced when climate of a region is being noted and would take a lot of time if it's done manually. Deep learning replaces the complex meteorological models with a simple algorithm which reduces the computation time. Deep learning can find even the smallest changes and accurately and predict the climate with more accuracy. These models are still not 100% accurate. Many scientists are trying to find better methods which are more accurate than previous ones. The future of machine learning in climate prediction looks promising and has a scope of improvement yet to come. ANN used to be the most popular method before being replaced by SVR which was a kernel method. The current properties of a location are correlated to the past values of the same location. Purpose of this paper is to analyze the different deep learning model which were used and which are still used currently.

II. DISCUSSION & RESULTS

A. Artificial Neural Network

Authors (Charles Anderson, Imme Ebert-Uphoff, Yi Deng, Melinda Ryan) are trying to learn about the complex dynamics governing the interactions between the radiative flux at the top of the atmosphere (TOA) and air/surface temperature. The authors have trained ANN model for analysis and each sample has 756 components, consisting of six atmospheric fields at all 126 locations (7 latitude and 18 longitude). The output is approximated using mean square and error is reduced using conjugate gradient algorithm. ANN is a set of neurons that act the way neurons work in nature. Neuron is a weighted sum of set of input and a bias with an activation function. Equation of a single neuron is $y_k=f_{act}(b+x_iw_i)$. ANN is effective when many neurons are connected together and the simplest one is feedforward neural network. If xi is the input w_i is the weight in matrix form and bias is $w_0=b$ by adding input $x_0=1$, a layer of neuron can be described as $y=f_{act}(wx)$. Inputs to the network is connected to the first hidden layer which is connected to more hidden layers. Last hidden layer is connected to the last layer which determines the output of ANN. Depending on the output, a single non-linear matrix equation is decided. Initial activation function of ANN was y=sin(x) and a simple neuron in it was called perceptron. Training of ANN is actually a parameter optimization of the network weights such that output of network minimizes a chosen bias function. Number of parameters for ANN is high therefore there exists a large number of local solutions. Finding solutions in ANN is hard so the best solution is found. Sometimes a training set could get stuck in local minima far from the best solution. Various methods have been proposed to deal with this problem. Overfitting happens when the model adapts to random variation in the dataset. After three-dimensional encoding of 3 models, 3 different projections are obtained. The First unit has captured annual cycle. Second unit removes this information from data and is able to learn different relationships. Patterns in this encoding are analyzed in terms of which co-variation of climate variables are being represented.

Revised Manuscript Received on July 10, 2019.

Gautham Naik, Department of Computer Science and Engineering, VTU, SMVITM Bantakal, Udupi, Karnataka, India.

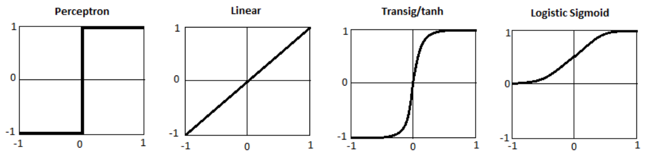
Nandan Nayak, Department of Computer Science and Engineering, VTU SMVITM Bantakal, Udupi, Karnataka, India.

Nithesh, Department of Computer Science and Engineering, VTU SMVITM Bantakal, Udupi, Karnataka, India.

Nithin H A, Department of Computer Science and Engineering, VTU SMVITM Bantakal, Udupi, Karnataka, India.

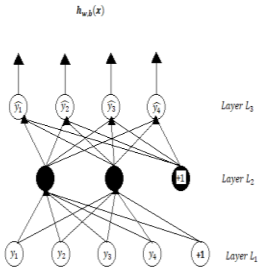
Nagaraj Bhat, Department of Computer Science and Engineering, VTU SMVITM Bantakal, Udupi, Karnataka, India.

Dr. K C Gouda, CSIR Fourth Paradigm Institute, Bengaluru, Karnataka, India.



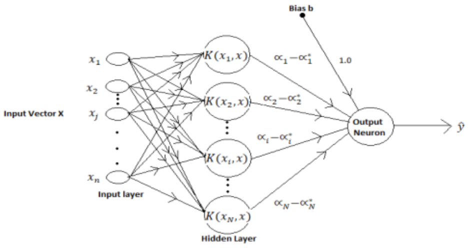
B. Autoencoder Network

Autoencoder learns less co-related functions sequentially. A single unit autoencoder network with bottleneck layer is trained to fit data. Original samples and approximated samples of the trained autoencoders are subtracted. In the second autoencoder, approximation of data by the first autoencoder is removed and the second autoencoder is trained on the residual data. In the third autoencoder, sequential projection and subtraction operation is done using Gran-Schmidt algorithm which is used in the single value decomposition. The first unit captures the annual cycle, by removing this information second unit different relationships and the third unit. The pattern in the encoding are reanalyzed with respect to which co-variation in climate variable are represented. The authors have used this model to predict temperature. 4 different models have predicted the temperature 1,3,6 and 12 hours apart. In the first input temperature is used. In second experiment, exogenous input as precipitation was used. After tuning hyperparameters for 8 models, they are compared. Introducing precipitation as an input improved the results. In the future, data from other places can help in making better prediction.



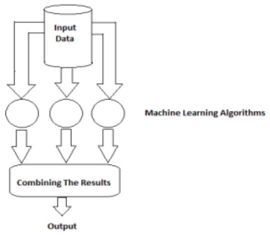
C. Support Vector Machine

The Support Vector Machine (SVM) is proposed by Cortes and Vapnik based on statistical learning theory. It has been widely applied in time series prediction as well as power load demand forecasting and fault prediction.



D. Ensemble Method

Authors Xueheng Qiu, Le Zhang, Ye Ren and P. N. Suganthan have used this model. Their ensemble model consists of deep learning methods from DBN and SVR and evaluated using Mackey-Glass time series dataset. This method combines different learning algorithms in an efficient manner to obtain the result. Statistical reason: Ensemble model reduces the chance of taking an unrelated model by joining all these models. Computational reason: Some algorithms perform local search to obtain optimal solution. Ensemble techniques perform local searches from various starting points. Representational: No single Hypothesis can represent a true function f ; rather, it can be approximated using the mean of different hypotheses. This technique works best for short-term time series load demand forecasting sequence. Ensemble methods have better results compared to single structure algorithms. The ensemble method consisting of DBN (Deep Belief Network) and SVR (Support Vector Regression) has shown stronger outcomes for rebuilding practice and precise outcomes for forecast.



E. Convolutional Neural Network

It comprises of several convolution layers and some fully linked layers. Between two adjacent convolutional layers, rule sampling operations take place. CNN inputs are format (w, h, c) where w is width, h is pixel height, and c is each pixel's color channel number. CNN output is the q probability unit vector depending on the number of categories. The convolution operation between kernel and input image are performed. Typical convolution layer contains k filters with size (*i*,*j*,*c*) where *i* is the width, *j* is the height. Filters are less than width *w* and height *h* of input image. Colour channel of the input image is always *c*. Each filter is combined independently with input images followed by non-linear transformation and generates feature maps k which is the input for the next layer. The dot product is calculated between the filter entry and the local area to which the input image is connected. Learnable filters are set as the parameter. Certain regions produce larger output than others when the filter goes through all the inputs. It can be extracted and preserved in the feature maps and passed to the next layer regardless of where this feature is present in the input. The smaller sample is taken from an existing sample generated from the convolution layer over a common area (s.t) where the width is s and the height is t. As the depth rises, feature map resolution becomes more severe. In a fully connected layer, all hidden units are connected to a previous layer. In the last layer, the fully connected layer features vectors from previous layer on high

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

level reasoning to produce final class score for image object. This CNN architecture was used by authors to classify tropical cyclones, atmospheric storms and weather fronts. The deep CNN has 4 learnable layers, 2 convolution layers and 2 fully connected layers.. Authors have used this model to classify atmospheric rivers and weather fronts. Since the model chosen is small, the extreme event classification showed high accuracy. No overfitting was observed because of 4 layers and weight decay regulation. Good train and test results says that CNN is able to learn representation of labelled climate pattern label and predict based on feature learned. Deep CNN avoid subjective threshold by learning from the labeled data. For tropical cyclone, the train and test accuracy were 99% and 99% respectively for around 30 min. for atmospheric river, train and test percentage was 90.5% and 90% respectively for 6-7 hours. For the event weather front, train and test accuracy was 88.7% and 89.4% respectively. In the future, authors want to create a CNN architecture that can discriminate between different variables based on the type of event and that can handle events at different spatial scales.

F. Decision Tree

Decision tree repeatedly divides a multidimensional dataset into smaller subdomain by selecting variables and decision requirements that tries to get as many dissimilarities as possible. In each node, every predictor is compared with dissimilarity metric. The splitting condition for a node is selected by selecting the node with highest prediction and threshold. After a reasonable number of splitting, prediction is rounded off to a single value. One of the advantages is that it is human readable and perform variable selection as a part of model selection process. Disadvantage is that there is a high chance of overfitting and if there is a small variance, it could lead to a large error.

G. Multi-Task Neural Network

MTTN is presented as series of fully connected hidden layer in which the activations at level l are a function of previous layer, h_l=h(h_l-1; W_l) parameterized by a weight matrix. The first hidden layer h_1 is connected to the input and the last hidden layer is connected to K outputs. Every hidden layer has a feature map and each output y_k is a prediction. The main advantage of MTTN is that it can learn a feature map and different prediction model together. The authors used a MTNN with two layers of 1000 and 100 hidden units each and four outputs. Performance of MTTN is compared against linear and logistic regression. It is observed that MTTN reduces MSE by 70% and improves AUC by 11%. To show that MTTN has interpretable features backward variable selection is used to identify second layer feature that best predicts the next month’s temperature. 50 input patterns are selected that maximize the feature’s activation and mean and standard deviation of each variable is plotted. It is concluded that increasing solar radiation means increased future temperature.

H. Stacked Autoencoder

Authors are Moinul Hossain, Banafsheh Rekabdar, Sushil J. Louis, Sergiu Dascalu. A neural network having an input, an output and a few hidden layers is called autoencoder network. Autoencoder constructs its own input and output. Useful features of input data can be learnt by keeping hidden layers narrower than input data. Stacked auto encoders have various layers of sparse auto encoders in which inputs from one layer are entered into the next layer.

The encoding step of each layer is:

$$a^{(l)} = f(z^{(l)})$$
$$z^{(l+1)} = W^{(l,1)}a^{(l)} + b^{(l,1)}$$

While the decoding step is given by:

$$a^{(n+l)} = f(z^{(n+l)})$$
$$z^{(n+l+1)} = W^{(n-l,2)}a^{(n+l)} + b^{(n-l,2)}$$

If a(l) is the node activation in layer l, z(l) is the total weighted input sum for the unit in layer l (for the first layer, z is the input data), W(l, k) ->weight b(l, k)->bias.

Authors focused on the applicability of deep learning to predict hourly air temperature with real sensor data. The air temperature was modelled according to temperature, wind speed, relative humidity and barometric pressure. They observed stacked auto encoder performed better than neural network in temperature prediction because of 97.94% accuracy rate compared to 94.92% accuracy of neural network. They suggest that more variables like relative humidity can improve the accuracy.

I. Bayesian deep learning

Authors emphasizing on this topic are Thomas Vandal, Auroop R Ganguly. Direct results are difficult to obtain for more than one hidden layer in a network. First, in the neural network, weights are defined as w={w1,w2, ... wL } where w ~ N(0,1) and L is the number of layers.

- Random outputs are denoted by fw(x) and the probability is given by P(y|fw(x))

If there are data X and Y, we conclude that the subsequent p(w|X, Y) describes the parameter distribution.

- Regression task with a predicative Gaussian posterior is p(y|fw(x)) = N$\left[\hat{y}, \hat{\sigma}^2\right]$ with random output given by:

$$\left[\hat{y}, \hat{\sigma}^2\right] = f^w(x)$$

Variational inference is applied to the weights and approximate and trackable distribution $q_\theta(w) = \prod_{l=1}^{L} q_{M_i}(W_l)$ where $q_{M_i}(W_l) = \mu_l$X diag[bernoulli(1-p_l)k_1] parameterized by $\theta_l = (M_l, p_l)$ containing the weight mean of shape k_l x k_{l+1} where k is the number of hidden layers in l and p_l is the dropout probability are defined. Furthermore, Kullback-leibler divergence is minimized between $q_\theta(\omega)$ and true posterior p(w|X,Y).

- The optimization objective of variational interpretation is:

$$\hat{\mathscr{L}}(\theta)=-\frac{1}{M}\sum_{i\in S}\log p(y_i|f^{\omega}(x_i))+\frac{1}{N}KL(q_\theta(\omega)\|p(\omega))=\mathscr{L}_x(\hat{\theta})+\frac{1}{N}KL(q_\theta(\omega)||p(\omega))$$

A well estimated pl is important to obtain well calibrated uncertainty estimates. Concrete distribution prior to which the Bernoulli distribution is continuously approximated is preferred over pl as constant.

- The kl divergence term by Gal et.al is:

$$KL(q_\theta(w)\|p(w))=\sum_{l=1}^{L}KL(q_{M_l}(w_l)\|p(w_l))$$

$$KL(q_{M_l}(w)\|p(w))\propto\frac{l^2(1-p_l)}{2}\|M_l\|-K_l\,\mathscr{H}(p_l)$$

Where the entropy of Bernoulli random variable with probability p is $\mathscr{H}$ (p)= -p log p=(1-p) log(1-p). Given the entropy term for the desired effect, the learning dropout probability cannot exceed 0.5. The three Bayesian neural networks-MC-Dropout, Concrete Dropout, and Dropout with Alpha-Divergence are taught. On sample information and adversarial examples, the 3 techniques have excellent outcomes. Predictive uncertainty about their statistical downscaling data set, however, is bad. Currently the model provides value in predicting uncertainty but more experimentation needs to be done. The current model assume output variable follows a normal distribution which doesn't work in real life. The authors suggest that adaption to a well-suited distribution and network architecture at extremes will improve performance. Nevertheless, this model makes better prediction than vanilla dropout methods so they are going to continue with the result.

J. Recurrent Neural Network

Elman network is an RNN form consisting of one or more layers that are hidden. The input layer provides the first layer weights and likewise the prior layer provides the present layer the value. Elman network has functions: continue and discontinue which acts as an activation function. The delay in the previous time(t-1) may be used in the current time(t) in the first hidden layer. The main advantage of RNN is that noise will be adjusted through feedback connection at the previous input to the next level.

Let x(t) and y(t) be time series input and output respectively ; WIH, WHH and WHO are the three link weight matrices.The training sequence begins at time t0 ends at time t1 and the sum of the standard error function $E_{sse/ct(t)}$ over time at each time step:

$$E_{total}(t_0,t_1)=\Sigma_{t=t_0}^{t_1}E_{sse/ce}(t)$$

$$\Delta W_{ij}=-\eta\frac{\partial E_{total}(t_0,t_1)}{\partial W_{ij}}=-\eta\Sigma_{t=t_0}^{t_1}\frac{\partial E_{total}(t_o,t_1)}{\partial W_{ij}}$$

Heuristically optimizing technique for predicting rainfall is performed on the basis of the weather dataset ENSO variable. The authors concluded that with medium accuracy, RNN can be applied to prediction of rainfall. They hope to use Conditional Restricted Boltzmann Machine (CRBM) and Convolutional Neural Network (CNN) in future experiments that provide better representation.

K. Back-Propagation

A random number is generated and assigned to the network node. It is usually a small number. First the forward propagation is done for the input node throughout the network. The expected outcome and the obtained results are compared and error is calculated. Backpropagate the error value through the network. Weights are updated in the end. Few of the problems faced here is that getting the final answer may take a lot of time. Final weights might reach a local minima if the error surface is too complex. This can be solved by Levenberg-Mearquarde method.

III. CONCLUSION

In this paper, we conducted a survey of profound learning-based climate prediction research. We have taken few of the papers written by authors to study the models they have used and the area where these models can be used. These models are not 100% accurate so there is still a scope of improvement. There are more models proposed by different authors. Since machine learning is still an evolving field, we can expect better models to come and perform better than these models. This paper can motivate researchers about different uses of deep learning and to take deep learning in different fields of science. Deep learning can lead to an efficient and improved results which can help them create a better tomorrow.

REFERENCES

1. Afan Galih Salman, Bayu Kanigoro, Yaya Heryadi "Weather forecasting using deep learning techniques" 2015 International Conference on Advanced Computer Science and Information Systems (ICACSIS) https://ieeexplore.ieee.org/abstract/document/7415154/references#references
2. Machine Learning in Python for Weather Forecast based on Freely available Weather Data- E. B. Abrahamsen, O. M. Brastein, B. Lie Proceedings of ;The 59th Conference on Simulation and Modelling (SIMS 59), 26-28 September 2018, Oslo Metropolitan University, Norway http://www.ep.liu.se/ecp/article.asp?issue=153&article=024&volume=
3. M. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," Neural networks, vol. 6, no. 4, pp. 525–533, 1993
4. G. H. Golub and C. F. V. Loan, Matrix Computations. Johns Hopkins University Press, fourth ed., 2012
5. Application of Deep Convolutional Neural Networks for Detecting Extreme Weather in Climate Datasets. Yunjie Liu, Evan Racah, Prabhat, Joaquin Correa, Amir Khosrowshahi, David Lavers, Kenneth Kunkel, Michael Wehner, William Collins https://arxiv.org/abs/1605.01156
6. Vandal, T., Kodra, E., Dy, J., Ganguly, S., Nemani, R., & Ganguly, A. R. (2018). Quantifying Uncertainty in Discrete-Continuous and Skewed Data with Bayesian Deep Learning. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining - KDD'18. doi:10.1145/3219819.3219996

7. Xueheng Qiu ; Le Zhang ; Ye Ren ; P. N. Suganthan; Gehan Amaratunga, Ensemble deep learning for regression and time series forecasting 2014 IEEE Symposium on Computational Intelligence in Ensemble Learning (CIEL)
8. J. C. Sousa, H. M. Jorge, and L. P. Neves, "Short-term load forecasting based on support vector regression and load profiling," International Journal of Energy Research, vol. 38, no. 3, pp. 350-362, 2014.
9. C. Cortes and V. Vapnik, "Support-vector networks," Machine learning, vol. 20, no. 3, pp. 273-297, 1995.
10. H. Drucker, C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," Advances in neural information processing systems, vol. 9, pp. 155-161, 1997.
11. P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," The Journal of Machine Learning Research, vol. 11, pp. 3371-3408, 2010.
12. T. G. Dietterich, "Ensemble methods in machine learning," in Multiple classifier systems. Springer, 2000, pp. 1-15.
13. S. Chatterjee, A. Dash, and S. Bandopadhyay, "Ensemble support vector machine algorithm for reliability estimation of a mining machine," Quality and Reliability Engineering International, 2014.
14. An examination of deep learning for extreme climate pattern analysis Gilberto Iglesias David C. Kale Yan Liu
15. X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” inInternational Conference on Artificial Intelligence and Statistics,pp. 315–323, 2011.
16. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting, ”The Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929–1958, 2014
17. Sutskever, J. Martens, G. Dahl, and G. Hinton, “On the importance of initialization and momentum in deep learning, ”in Proceedings of the 30th international conference on machine learning, ICML 30, pp. 1139–1147, 2013
18. Machine Learning in Python for Weather Forecast based on Freely Available Weather Data E. B. Abrahamsen, O. M. Brastein, B. Lie Proceedings of The 59th Conference on Simulation and Modelling (SIMS 59), 26-28 September 2018, Oslo Metropolitan University, Norway

Retrieval Number: B11170782S319/19©BEIESP
DOI : 10.35940/ijrte.B1117.0782S319

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication