International Journal of Innovations & Research Analysis (IJIRA)
ISSN : 2583-0295, Impact Factor: 5.449, Volume 02, No. 03(II), July- September, 2022, pp 85-92

BAYESIAN NEURAL NETWORKS FOR STOCK PRICE FORECASTING

Dr. K. Sampath Kumar*
Dr. J. Purushottam**
Mr. R . Madhusudhana Chary***

ABSTRACT

The research paper shows a few different applications of them for classification and regression problems. The probabilistic graphical mode would be given be the Bayesian Network. BNNs are comprised of a probabilistic model and a Neural Network. It indicates the stock price returning and their conditional independencies through a cyclic graph extending standard networks with posterior inference in order to control over fitting. Bayesian Neural Networks for stock prices forecasting before and COVID -19 pandemic during Markov Chain Monte Carlo's (MCMC) sampling methods have been standing out in implementing inference of Bayesian Neural Network, many parameters and the need for better computational resources. The COVID-19 pandemic had a drastic impact in the world economy and stock markets given different levels of lockdowns due to rise and fall of daily infections. It is important to investigate the performance of related forecasting models during the COVID-19 pandemic given the volatility in stock markets. In this paper, we use novel Bayesian neural networks for multi-step-ahead stock price forecasting.

Keywords: *MCMC (Markov Chain Monte Carlo).*

Introduction

Bayesian Neural Network using Markov chain Monte Carlo methods is also described. Presupposing only basic knowledge of probability and statistics. From a practical standpoint this type of training is often susceptible to overfitting, as BNNs often do, one partial fix for this is to introduce regularization, this is equivalent to inducing priors on the weights (say Gaussian distributions). Again from a probabilistic perspective, this is not the right thing to do, though it certainly works well in practice.

This means that predictions by neural networks have biases which cannot be trivially distinguished from being due to the true nature of the creation and observation of data or not. In presenting such techniques, we show how uncertainties which arise in the predictions made by neural networks can be characterized in principle. We provide descriptions of the two favored methods for analyzing such uncertainties. We will also describe how both of these methods have substantial pitfalls when put into practice, highlighting the need for other statistical techniques to truly be able to do inference when using neural networks. Some examples of these cases are decision making systems, (relatively) smaller data settings, Bayesian Optimization, model-based reinforcement learning and others.

Structure Learning

Stochastic models allow direct specification of a model with known interaction between parameters to generate data. Thus, BNNs are a unique combination of neural network and stochastic models with the stochastic model forming the core of this integration. That means, in the parameter space, one can deduce the nature and shape of the neural network's learnt parameters. These two

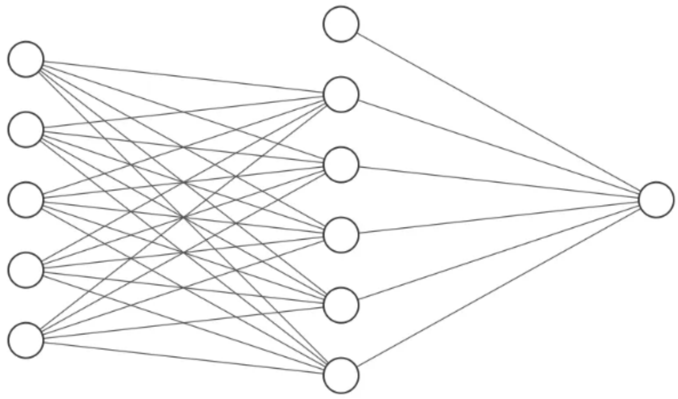
---

* Head of Department, Associate Professor of Statistics, Telangana University, Nizamabad, Telangana State, India.
** Assistant Professor of Statistics, Telangana University, Nizamabad, Telangana State, India.
*** Head of Department, Assistant Professor of Statistics, Nishitha Degree College (Autonomous), Nizamabad, Telangana State, India.

characteristics makes them highly attractive to theoreticians as well as practitioners. Recently there has been a lot of activity in this area, with the advent of numerous probabilistic programming libraries. Further this area is rapidly gaining ground as a standard machine learning approach for numerous problems.

Bayesian Neural Network has the ability to quality the uncertain in their predictive output. Bayesian Neural Network diagram with one input layer, one hidden layer, and an output layer with standard neural network, the weights between the different layers of the network take single values.

In a Bayesian neural network, the weight take on probability distribution. The process of finding this distribution is called marginalization, one important factor for training these networks is having a large enough set of training data to produce these networks is having, want to learn about methods for quickly increasing the size of your training data set.



A neural network diagram with one input layer, one hidden layer and one output layer A Bayesian convolutional neural network uses the Bayesian network techniques of expressing the network weight as probability distribution with a conventional neural network.

- **Neural Networks for Forecasting**

Deep learning commonly refers to outstanding models such as recurrent networks (RNNs), convolutional neural networks (CNNs), deep belief networks, and extended short-term memory networks (LSTMs) In recent years, with more computing power and massive datasets, deep learning models have demonstrated excellent performance in different fields, such as sentiment analysis image analysis.

The main advantage of deep learning models is the ability to automatically extract the goodfeatures of input data through the general-purpose learning procedure. Therefore, deep learning has also been widely used in various forecasting applications. The financial community has received a boost in developing solutions with deep learningmodels for financial forecasting research. Ding et al. utilized CNNs to evaluate the impactof different events on stock price behavior in the short, middle and long term.

Bayesian neural networks have strength in forecasting due to promising prediction with uncertainty quantification. Different Bayesian neural networks such as recursive Bayesian recurrent neural networks and evolutionary MCMC Bayesian neural network have been used for time series forecasting. Liang et al. proposed an MCMC algorithm for Neural Networks for selected time series problems.

- **Description**

Selection of some priors that are available in the library. Here, θ denotes the weights of the BNN, denotes the set of all priors and all the other variables denote hyper parameters of the priors.

Prior Density	p(θ)
Gaussian	Z^{-1} exp($-\|\theta-\mu\|^{2}/ 2\sigma 2$)
Laplace	Z^{-1} exp($-\|\theta-\mu\|^{2} /b$)
Student-t	Z^{-1} ($1 + \theta^{2}/ \nu$)$^{-(\nu+1)/2}$
Cauchy	Z^{-1} ($1 + \theta^{2}/ \gamma^{2}$)$^{-1}$
Multivariate Gaussian	Z^{-1} exp($-$ ½ $(\theta-\mu)^{\top}\Sigma^{-1}(\theta-\mu)$) 111
Multivariate t	Z^{-1} (1 + 1/ν $\{(\theta-\mu)\}^{\top}\Sigma^{-1}(\theta-\mu)$)$^{-(\nu+p)/2}$

Our library includes a range of predefined priors within a modular taxonomy, such that new priors can be easily defined and can inherit from existing super classes, such as location-scale distributions or multivariate distributions. Moreover, the hyper parameters of our priors can themselves be prior objects, which allows for the definition of hierarchical prior models. Also, the Mixture class allows to define mixture priors from all the other existing priors. A short overview over some popular prior distributions included in our library.

Methodology

To perform inference on the standard Bayes posterior, the temperature parameter of the sampler has to be set to $T = 1$. However, one can also use cold posteriors by setting the temperature to $T < 1$. In order to assess the accuracy of the inference in every single experiment, the sampler will estimate diagnostics such as the kinetic temperature and the configurational temperature. State-space reconstruction State-space reconstruction refers to embedding a time series into a vector so that it can be trained by machine learning models. According to the Taken's theorem, the embedding process must ensure that the original characteristics of the time series. Given a univariate time series, we can construct a multi-dimensional space vector by taking a point on the fixed delay of the original system.

Using Taken's embedding theorem, the state-Space reconstruction is given as follows. Suppose the actual series of closing stock price is $[x_1, x_2, \ldots, x_N]$, where N is the length of the series. First, we choose the embedding dimension m and a time lag T, and then capture windows of size m denoted by vector x for every T delay until N is reached. Our problem is multi-step prediction where we have n prediction horizons denoted by vector y. The reconstructed vector by state-space embedding is denoted by [x ; y]. Hence, for the first instance, we have

$X = [\, x_1, x_2, \ldots, x_m]$

$y_1 = [\, x_{m+1}; x_{m+2}; x_{m+3}; \ldots ; x_{m+n}]$

In the same way, we can obtain the rest of the instances for the entire time series as given below.

$X_1 = [\, x_{1+(t-1T+1}, x_{2+(t-1)T+2}, x_{3+(t-1)T+3}, \ldots, x_{m+(t-1)T+m}]$

Neural Networks

The current library is of course limited in the number of predefined priors it contains, since new priors can easily be defined in our modular framework. The hidden layer could be one layer or feature multiple layers.

A neural network model $f(x)$ can be defined as a composition of other functions. Given a series of input-output pairs $\{x, y\}$, the model is trained to approximate the function f such that $f(x) = \mathbf{y_t}$ for all pairs. In our setting,

$$f(\bar{\mathbf{x}}_t) = g\left(\delta_o + \sum_{h=1}^{H} v_h \times g\left(\delta_h + \sum_{i=1}^{m} w_{ih} \bar{x}_{t,i}\right)\right)$$

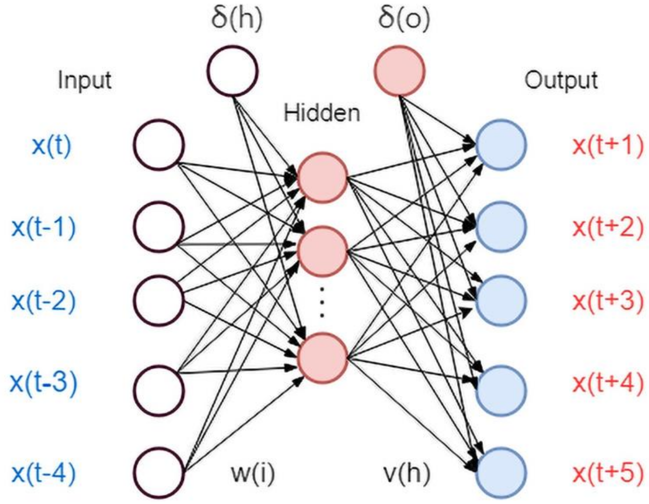
Where, m is the input number and H is the number of hidden layers. The function $g(.)$ is the Sigmoid activation function which is used in the hidden and output layers. The setup for multi-step ahead time series prediction problem using neural networks with one hidden layer.The complete set of parameters for the neural network model is shown in $\theta = (w, v, \delta)$ where

$\boldsymbol{\delta} = (\delta_o, \delta_h)$. **w** is the weight of the input to hidden layer. **V** is the weight of the hidden to output layer. $\boldsymbol{\delta}_h$ is the bias for the hidden layer, and $\boldsymbol{\delta}_o$ is the bias for the output layer.

Stochastic gradient descent (SGD) is one of the prominent methods of training neural net- works. SGD is an iterative method to optimize differentiable objective function with help of gradients. In some high-dimensional optimization problems, SGD reduces the computational burden by achieving faster iterations with a lower convergence rate. Training neural networks also can be considered as solving the non-convex optimization problem min $L(w)$, where $w \in R^n$ is the set of parameters and L is the loss function. The iterations of SGD can be given as

$w_k = w_{k-1} - a_{k-1} \Delta L(w_{k-1})$

where, w_k denotes the k^{th} iteration, a_k is the learning rate, and $L(w_k)$ denotes the gradient at w_k. We note that the learning rate is user defined parameter which depends on the type of problem and typically it is determined in trial experiments. Hence, extension of the SGD consider adapting the learning rate automatically during the learning process. Adaptive moment estimation is an effective stochastic optimization method that only requires first- order gradients with a small amount of memory requirement focused in adapting learning rate. This calculates the individual adaptive learning rates of the parameters from the estimates of first and second moments of the gradients. The weight update is expressed as follows:



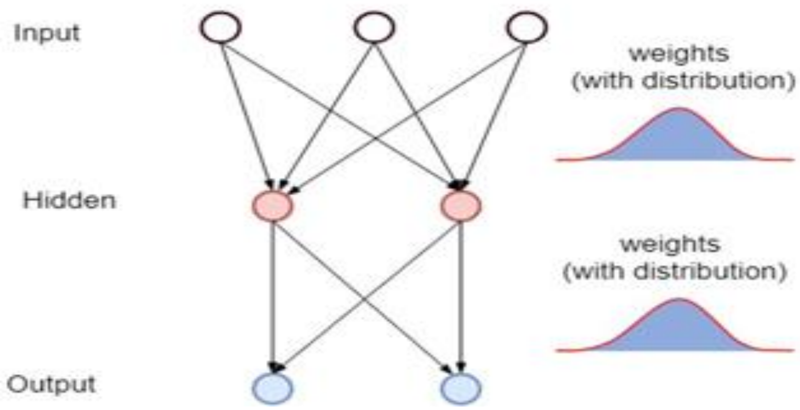
The time series {shown in red circles} is used as input for neural network 5 steps – ahead in time.

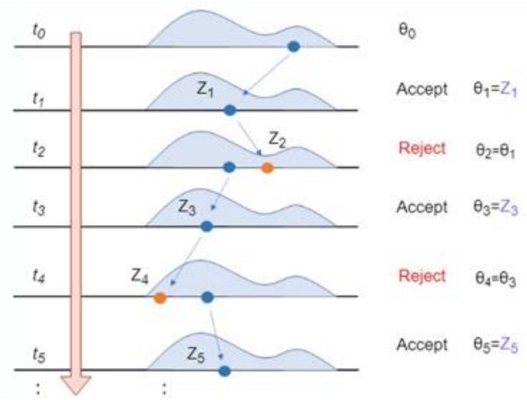
$u_{k-1} = \beta_1 u_{k-2} + (1 - \beta_1)\, \Delta\, f(w_{k-1})$

$n_{k-1} = \beta_2 n_{k-2} + (1-\beta_2)\, \Delta\, f(w_{k-1})^2$

- **Bayesian Inference**

Bayesian neural network provide a probabilistic implementation of standard neural network with the key difference where the weight and biases are represented posterior probability distributions rather than single point estimates similar to standard Bayesian neural network also have universal continuous function approximation capabilities. The challenge of Bayesian inference is in sampling to approximation the posterior distribution of neural network weights and biases. The inference procedure begins by setting prior information over the weights and biases. Then the sampling scheme such as MCMC employs a likelihood function that takes into account the training data accepting or rejecting a proposed sample. The implementation of Bayesian neural network using MCMC sampling is a non-linear activation function in the Bayesian neural network. More over due to large number of parameters given different application. It is difficult to get information priors and hence we can construct the likelihood function using the set of weights and biases for M network parameters and S training instances. We note that we use a signal plus model.





f(x) with input function x_1 , we have

$$p(\mathbf{y_S}|\boldsymbol{\theta}) = -\frac{1}{(2\pi\tau^2)^{S/2}} \times \exp\left(-\frac{1}{2\tau^2}\sum_{t\in S}(\mathbf{y_t} - f(\bar{\mathbf{x}}_\mathbf{t}))^2\right)$$

Which satisfies the multivariate probability density function. The prior is based on Gaussian distribution in the case of θ and Gamma distribution in the case τ^2 is given below.

$$p(\boldsymbol{\theta}) \propto \frac{1}{(2\pi\sigma^2)^{L/2}} \times \exp\left\{-\frac{1}{2\sigma^2}\left(\sum_{i=1}^{M}\theta\right)\right\} \times \tau^{2(1+\nu_1)} \exp\left(\frac{-\nu_2}{\tau^2}\right)$$

Where, σ^2 is determined by exploring variance in weights and biases of training neural network and for similar applications.

Experiment and Results

In this section, we provide details about the datasets and present research design with computational result.

- **Data**

We choose 4 stocks from 4 different countries to evaluate the performance of the respective methods. These include 3M Company (MMM) from United States, China Space sat Company Limited (600118.SS), Commonwealth Bank of Australia (CBA.AX), and Daimler AG (DAI. DE) from Germany. The respective datasets feature the closing price with the time period from 01/01/2012 to 01/07/2020 from Yahoo Finance. MMM is an American world-renowned multinational company with diversified products, which cover various fields such as house- hold goods and medical supplies. China Space sat Company Limited is an aerospace high- tech enterprise specializing in the development and application of small satellites in China. Commonwealth Bank is the largest commercial bank in Australia. Daimler AG is a German company which is the largest commercial vehicle manufacturer and the largest luxury car manufacturer in the world.

- **Experimental**

In parallel tempering MCMC, we use burn-in rate of 0.5 with maximum of 100,000 samples with maximum temperature value of 2. Note that the burn-in rate also defines the first-phase where global search is enforced by parallel tempering in of Algorithm 1. We use replicas with swap interval is 5. The Langevin-gradient proposals use learning rate of 0.1 and applied with a probability of 0.5. We use one hidden layer feedforward neural network with 5 output units where each output unit denotes a prediction horizon for the 5-step-ahead stock black- price prediction problems. We apply Taken's embedding theorem to reconstruct data with dimension D = 5 and time-lag T = 2. Hence, 5 input neurons with 5 hidden neurons are used in the respective neural network models.

The experiments are executed as follows.

- Evaluate multi-step-ahead stock price prediction using novel Bayesian Neural Networks before 2020.
- Compare the results with feedforward neural network using Adam optimizer (FNN- Adam) and feedforward neural network using stochastic gradient descent (FNN-SGD) training algorithm. Evaluate multi-step-ahead stock price prediction using novel Bayesian neural networks during 2020

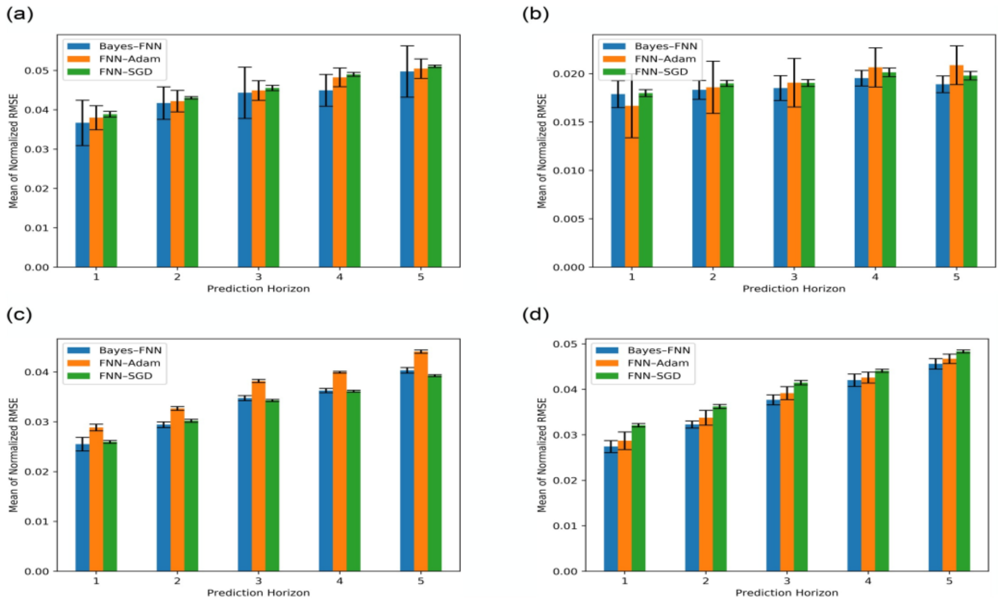
The results report the mean and 95% confidence interval from 30 experimental runs with different weight and bias initialization in the respective models for 5-step ahead prediction.

- **Prediction Results before 2020 Year**

Prediction result before 2020 year, the first 80% of the data is used for training and remaining is used for testing, gives further details of the time frame considered and indicates the exact dates for the respective stocks, reports the performance of the respective method in prediction of future trends of given stock prices they include horizon (step). the mean with a 95% confidence interval as error-bars. further presents the results numerically. In the case of MMM Bayesian neural network (Bayes-FNN) performs the best with the lowest RMSE compared to FNN-Adam and FNN-SGD where the error increases with the prediction horizon. Presents results for stock 600118.SS that show that Bayes-FNN

Time span of data considered for each stock-price before 2020 (A Sample of 4 Companies)

	MMM	**600118.SS**	**CBS .AX**	**DAI.DX**
Train (80 %)	3.1.2012- 24.5.2018	4.1.2012 - 28.5.2018	3.1.2102- 24.7.2018	2.1.2012- 22.5.2018
Test (20 %)	25.5.2021 -31.12.2019	29.5.2018 - 31.12.2019	25.7.2018- 31.12.2019	23.5-2018- 30.12.2019



Prediction Performance (Mean of normalized for the three models (Bayes-FNN, FNN-Adam, and FNN-SGD). Note that the predictions were normalized in range of [0, 1]. Gives best performance in all the prediction horizons. . In this case for stock, shows the Bayes-FNN shows the best performance in the prediction horizon increases. In the case for stock DAI.DE, shows that Bayes-FNN gives best performance and the RMSE increases as the prediction horizon increases. Stock MMM , Stock 600118.SS, Stock CBA.AX), and Stock DAI. DE show the prediction on the test dataset for the prediction horizons, Bayes-FNN. We find out that the uncertainty is relatively small for stock DAI.DE and CBA. AX when compared to stock MMM and 600118.SS. In case of stock 600118.SS, there is similar level of accuracy in prediction but higher uncertainty while the stock MMM shows poor prediction and uncertainty quantification in the first half (less than 100 days).

Problem	Step	Bayes-FNN	FNN-Adam	FNN-SGD
MMM	1	0.03669±0.00577	0.03800±0.00305	0.03889±0.00073
	2	0.04173±0.00413	0.04221±0.00274	0.04305±0.00032
	3	0.04434±0.00654	0.04491±0.00249	0.04555±0.00067
	4	0.04495±0.00404	0.04826±0.00237	0.04902±0.00050
	5	0.04975±0.00652	0.05047±0.00249	0.05107±0.00031
600118.SS	1	0.01789±0.00139	0.01669±0.00331	0.01800±0.00036
	2	0.01833±0.00098	0.01860±0.00269	0.01900±0.00032
	3	0.01852±0.00128	0.01908±0.00251	0.01904±0.00035
	4	0.01954±0.00081	0.02066±0.00202	0.02015±0.00045
	5	0.01891±0.00087	0.02088±0.00200	0.01980±0.00044
CBA.AX	1	0.02551±0.00136	0.02886±0.00067	0.02597±0.00028
	2	0.02942±0.00058	0.03270±0.00037	0.03021±0.00031
	3	0.03475±0.00050	0.03820±0.00031	0.03429±0.00021
	4	0.03628±0.00043	0.03996±0.00017	0.03616±0.00020
	5	0.04034±0.00055	0.04409±0.00033	0.03929±0.00020
DAI.DE	1	0.02743±0.00131	0.02871±0.00194	0.03212±0.00037
	2	0.03228±0.00078	0.03378±0.00163	0.03626±0.00046
	3	0.03771±0.00109	0.03918±0.00142	0.04150±0.00050
	4	0.04204±0.00136	0.04261±0.00121	0.04410±0.00035
	5	0.04562±0.00115	0.04674±0.00104	0.04836±0.00033

- **Result**

We apply Taken's embedding theorem to reconstruct data with dimension D = 5 and time T = 1. In certain stocks, we included January and February 2020 data for training was not widespread in countries such as USA, Germany and Australia during that period. We provide an investigation to check how the stock price changes during 2020, And effect of the stock price trend before on the stock price.

Consequently, we use the data previous for different stocks as training data to set the model performance during 2020 and with all the training data from where the exact dates are given general, we appended with data from March and April, 2020 that covers the first phase of the pandemic in the respectivecountries that affected the stocks. We have dataset the test dataset, while second half of data is used as test data set. In this case, we only provide results using the Bayes-FNN method.

Conclusions

In conclusion Bayesian Neural Networks are useful for integrating and modeling uncertainties. We applied the methods in Bayesian neural networks for multi-step-ahead stock price forecasting. The Bayesian neural network used state of art sampling strategy that incorporated parallel computing, Langevin and parallel tempering MCMC for improving sampling which provided very promising results when compared to novel neural networks methods. Our investigation revealed that it is important to incorporate data during an extreme event for better model building. In the experiments training dataset improved the prediction accuracy is significant. Hence high volatility in the stock black price makes forecasting very challenging and increases model uncertainty, their applicability in stock price prediction remains due to volatility and hence model validity is essential.

References

1. R.M. Neal, Bayesian Learning for Neural Networks, University of Toronto, 1995
2. D.J. MacKay, A practical Bayesian framework for backpropagation networks, Neural Compute
3. T. Bayes, An essay towards solving a problem in the doctrine of chances, Philos. Trans. R. Soc. Lond.
4. Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J.V. Dillon, B. Lakshminarayanan, J. Snoek, Can you trust your model's uncertainty, Evaluating predictive uncertainty under dataset shift, 2019.
5. K. Ciosek, V. Fortuin, R. Tomioka, K. Hofmann, R. Turner, Conservative uncertainty estimation by fitting prior networks, in: International Conference on Learning Representations, 2019.
6. Kendall, Y. Gal, What uncertainties do we need in Bayesian deep learning for computer vision?, in: Proceedings of the 31st International Conference on Neural Information Processing System.

7. Bayesian Neural Networks for Classification Skyler Richard Saucedo

8. MacKay DJC. A Practical Bayesian Framework for Backpropagation Networks. Neural Comput. 1992;4(3):448–472. https://doi.org/10.1162/neco.1992.4.3.448

9. Hinton GE, van Camp D. Keeping the Neural Networks Simple by Minimizing the Description Length of the Weights. In: Proceedings of the Sixth Annual Conference on Computational Learning Theory. Association for Computing Machinery; 1993. p. 5–13.

10. Neal RM. Bayesian Learning via Stochastic Dynamics. In: Advances in Neural Information ProcessingSystems 5. Morgan-Kaufmann; 1993. p. 475–482.

11. Neal RM, et al. MCMC using Hamiltonian dynamics. Handbook of Markov Chain Monte Carlo. 2011; 2(11).

12. Welling M, Teh YW. Bayesian Learning via Stochastic Gradient Langevin Dynamics. In: Proceedings of the 28th International Conference on International Conference on Machine Learning. Omnipress;2011. p. 681–688.

13. Chandra R, Jain K, Deo RV, Cripps S. Langevin-gradient parallel tempering for Bayesian neural learn-ing. Neurocomputing. 2019; 359:315–326. https://doi.org/10.1016/j.neucom.2019.05.082

14. Monteil V, Kwon H, Prado P, Hagelkrüys A, Wimmer RA, Stahl M, et al. Inhibition of SARS-CoV-2 infections in engineered human tissues using clinical-grade soluble human ACE2. Cell. 2020;. https://doi.org/10.1016/j.cell.2020.04.004 PMID: 32333836.

□❍□