Abstract:

From the time Neural networks are introduced multiple human problems are solved. However, they don't tell about the uncertainty of their estimates. To solve this issue sometimes probabilistic models are attached to them such as the Bayesian model. In this experiment, the Bayesian neural network is applied to air pollutant data set to show its working. The data set is pre-processed after being completely described along with its attributes. The experiment with its model building, parameter settings, and loss curve of training and validation is shown.

Keywords：

Neural network,probabilistic models ,Bayesian neural network

1. Introduction:

In recent times many algorithms were made to solve the problems faced by humans in daily life. Many of them worked perfectly in some scenarios, but still many problems were unsolved as they needed human vision to solve them. Machines were unable to have human brainpower. After getting inspired by brain neurons artificial neurons were made to solve the problem with human-like vision.

The structure of Artificial neural networks imitate human neurons and contain multiple nodes. These neurons are connected with each other and can get input from data to perform simple operations on it and then pass to other neurons. Neural networks work best for complex problems in real-life situation ,they can understand the relationship between input and output quite well like a human brain.

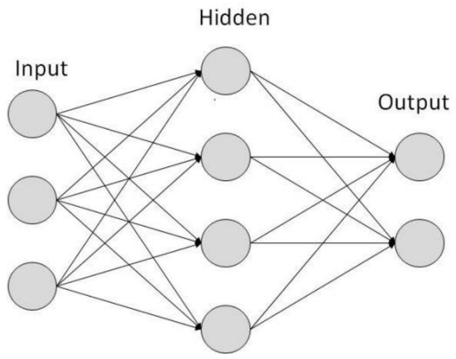


Figure 1:A Simple neural network

There are two paths in the neural network feed-forward and feedback. Feed-forward information travels in one direction and is not allowed to move back while in feedback information travels in both directions forward as well as back. The flow of information is controlled by the weights and the biases of the neural network which are the parameters of it. And during training, we try the optimize these parameter values. This network can be used in many filed which were easy for a human but not for machines for example Aerospace, Military, industry, Medical, speech and Transportation.

After Neural networks introduction, many challenges like object detection, classification and object segmentation in the field of Computer vision, nature language processing and Machine learning have been solved with the help of them. Neural networks are great if you are generating predictions with lots of data. But they don not tell about the uncertainty of their estimates.uncertainty can be very useful for non-liner risk functions. Besides their success,they are unable to think about uncertainty in predictions.To use neural networks more effectively they were combined with a probabilistic model so that they can get more scope in statistical problems this merged model was named as Bayesian neural network[2]

2. Bayesian Neural Networks:

The Bayes models use the Bayes theorem to describe the probability of the hypothesis given any clue. The formula used to calculate the probability is given below:

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

In the above formula:

A, B = Events

P(A\B) = probability of A given B is true

P(B\A) = probability of B given A is true

P(A) = the independent probabilities of A

P(B) = the independent probabilities of B

The models containing both the Neural network as well as the probabilistic model are termed Bayesian Neural Networks. These models contain neural networks that are trained with the help of the Bayesian interface[3]. The main idea behind these models is to combine the strengths of the neural network along with the stochastic model. The statistical models produce probabilistic guarantees on the predictions by generating complete posterior distribution.

Bayesian deep learning models usually form uncertainty estimates by two things one by placing distributions over model weights, second by learning a direct mapping to probabilistic outputs[4].

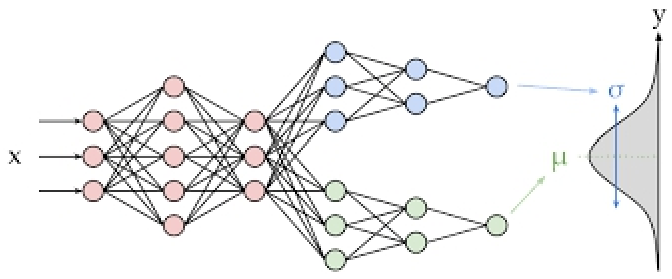


Figure 2: Bayesian Neural Network

Bayesian Networks (BN) are graphical structures that are used to represent the probabilistic relation between random variables. This network thinks about the uncertain domain that's why is important in the statistical domain. Bayesian networks are also called **Belief Networks** or **Bayes Nets**[6].

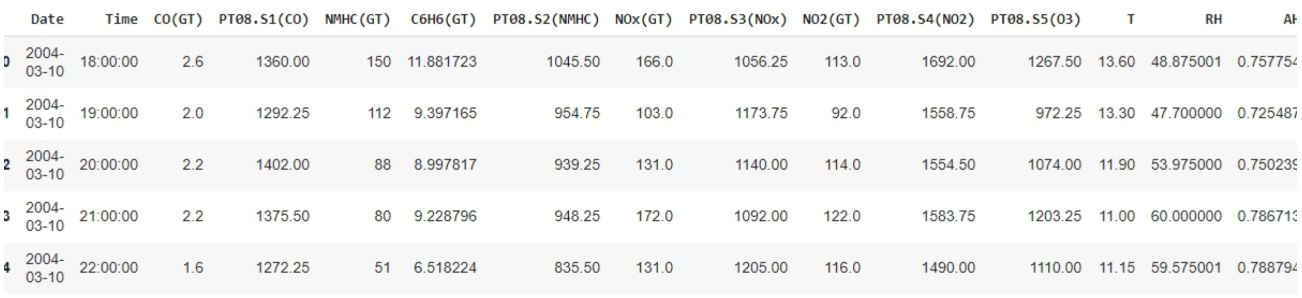
The below experiment shows the implementation of the Bayesian Neural Network with the help of Air pollutant data set from De Vito. In this experiment, the quality of Air from reference analyzer is predicted using our desired Bayesian Neural model. The experiment includes Data pre processing, Model building, model testing along prediction on testing data.

2.1 Data set:

The data set used for this model is the Air Quality data set from De Vito. The data set include 9358 instances of hourly response which is from an array of 5 metal oxide chemical sensors that were embed in a device located in Italy. This data set is from March 2004 to February 2005.
The data set can be downloaded from the given link :
http://archive.ics.uci.edu/ml/datasets/Air+Quality

Attributes Details:

0. Date in format (dd/mm/yyyy)
1. Time in format (hh.mm.ss)
2. Averaged concentration of CO(true hourly) in mg/m^3 (reference analyzer)
3. Hourly averaged sensor response of (tin oxide) PT08.S1 (nominally CO targeted)
4. Averaged overall Non metanic Hydrocarbons concentration (true hourly)in microg/m^3 (reference analyzer)
5. Averaged Benzene concentration(True hourly) in microg/m^3 (reference analyzer)
6. Hourly averaged sensor response of (titania) PT08.S2 (Nominally NMHC Targeted)
7. Averaged NOx concentration(True hourly) in ppb (reference analyzer)
8. Hourly averaged sensor response of (tungsten oxide)PT08.S3(Nominally NOx Targeted)
9. Averaged NO2 concentration(True hourly) in microg/m^3 (reference analyzer)
10. Hourly averaged sensor response of (tungsten oxide) PT08.S4 (Nominally NO2 Targeted)
11. Hourly averaged sensor response of (indium oxide) 1 PT08.S5 (Nominally O3 Targeted)
12. Temperature measured in Â°C
13. Relative Humidity measured (%)
14. AH Absolute Humidity measured

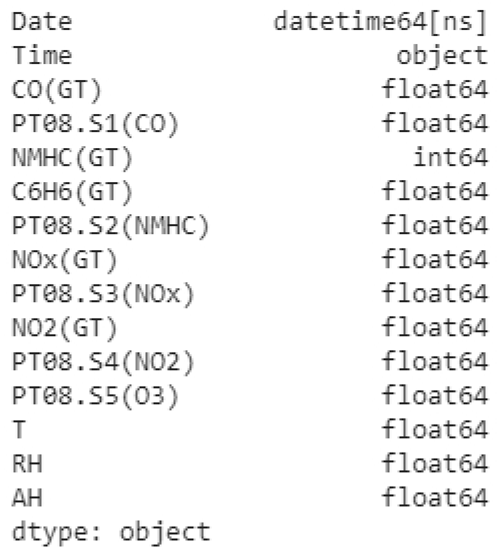


	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O3)	T	RH	AH
0	2004-03-10	18:00:00	2.6	1360.00	150	11.881723	1045.50	166.0	1056.25	113.0	1692.00	1267.50	13.60	48.875001	0.75775
1	2004-03-10	19:00:00	2.0	1292.25	112	9.397165	954.75	103.0	1173.75	92.0	1558.75	972.25	13.30	47.700000	0.725487
2	2004-03-10	20:00:00	2.2	1402.00	88	8.997817	939.25	131.0	1140.00	114.0	1554.50	1074.00	11.90	53.975000	0.75023
3	2004-03-10	21:00:00	2.2	1375.50	80	9.228796	948.25	172.0	1092.00	122.0	1583.75	1203.25	11.00	60.000000	0.78671
4	2004-03-10	22:00:00	1.6	1272.25	51	6.518224	835.50	131.0	1205.00	116.0	1490.00	1110.00	11.15	59.575001	0.78879

Figure 3: Data frame of data set representing first five rows

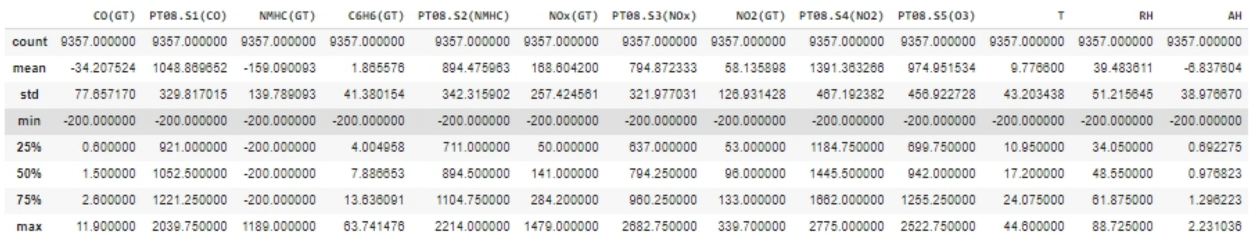
2.2 Pre-Processing:

The Pre-Processing of data started by first exploring the data type of all the attributes. The details of all the attributes types are given below:



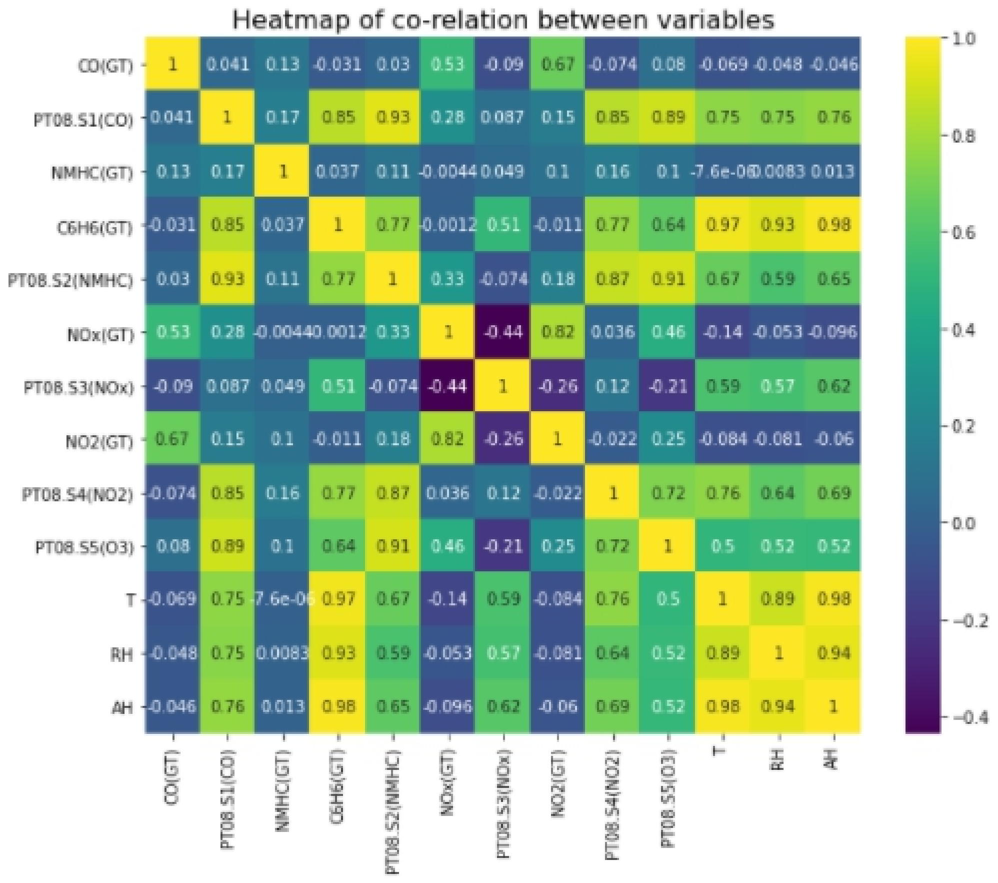
```
Date              datetime64[ns]
Time                      object
CO(GT)                   float64
PT08.S1(CO)              float64
NMHC(GT)                   int64
C6H6(GT)                 float64
PT08.S2(NMHC)            float64
NOx(GT)                  float64
PT08.S3(NOx)             float64
NO2(GT)                  float64
PT08.S4(NO2)             float64
PT08.S5(O3)              float64
T                        float64
RH                       float64
AH                       float64
dtype: object
```

After checking the data type dimensions of the data were checked and found to be **(9357,15)**. The basic statistics of the data like mean, min, max, std and count were printed to know more about data and are shown below:



	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O3)	T	RH	AH
count	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000
mean	-34.207524	1048.869652	-159.090093	1.865576	894.475963	168.604200	794.872333	58.135898	1391.363266	974.951534	9.776600	39.483611	-6.837604
std	77.657170	329.817015	139.789093	41.380154	342.315902	257.424561	321.977031	126.931428	467.192382	456.922728	43.203438	51.215645	38.976670
min	-200.000000	-200.000000	-200.000000	-200.000000	-200.000000	-200.000000	-200.000000	-200.000000	-200.000000	-200.000000	-200.000000	-200.000000	-200.000000
25%	0.600000	921.000000	-200.000000	4.004958	711.000000	50.000000	637.000000	53.000000	1184.750000	699.750000	10.950000	34.050000	0.692275
50%	1.500000	1052.500000	-200.000000	7.886653	894.500000	141.000000	794.250000	96.000000	1445.500000	942.000000	17.200000	48.550000	0.976823
75%	2.600000	1221.250000	-200.000000	13.636091	1104.750000	284.200000	960.250000	133.000000	1662.000000	1255.250000	24.075000	61.875000	1.296223
max	11.900000	2039.750000	1189.000000	63.741476	2214.000000	1479.000000	2682.750000	339.700000	2775.000000	2522.750000	44.600000	88.725000	2.231036

Then the data was explored by finding the correlation between parameters. The heat-map of correlations between variables is shown below:



This data needs to be pre-processed as it contains missing values along with outliers. So the data is processed by removing rows with missing values from the given data and the outliers were first detected and then removed using an isolation forest. The mean value was scaled at 0 while the variance was scaled to 1.

2.3 Model building:

To compensate for the aleotoric uncertainty that comes from the noise of the output layer the probabilistic layers are attached with dense layers. The first hidden layer contains 10 nodes while the second layer contains total of 14 nodes (4 for mean and 10 for Variance and covariance of the multivariate Gaussian posterior probability distribution in the final layer. The number of parameters in Models are 224.

To compensate for epistemic uncertainty along with aleotoric uncertainty the dense layer is changed with the Flipout layer due to which total parameters become 424 and Bayesian Neural Networks acts as an Ensemble.

2.4 **Training and Testing:**

After setting all the parameters and model settings, the model was trained on training data. The data set was split into training and testing data. In this experiment, 70% of data is used as training data and 30% as testing data. The experiment was run on different epochs and the best value was selected as 50 as shown in the loss graph below in figure 4.
The model was tested after successful training at 50 epochs for the remaining 30 % data. As its a probabilistic model so Monte Garlo Experiment is performed to show predictions. In specific, every prediction of input sample result into different output. That is why the expectation over many predictions has to be calculated.

Monte Garlo Experiments or methods are a broad class of algorithms that depends on repeated random sampling so that the numeric result can be found. The method uses the concept of randomness to solve deterministic problems. These methods are mostly used in a mathematical problem where it is difficult to use other approaches.

3. Experiment:

The experiment was performed with all the mentioned parameter settings and specified model (Bayesian neural network). In this experiment the value of the reference analyzer of **CO, C6H6, NOx, NO2** are predicted using the remaining data.

3.1 Experimental Analysis:

Some of the states are given below from the experiment

Loss Curve:
The loss curve for testing and training data is shown below :

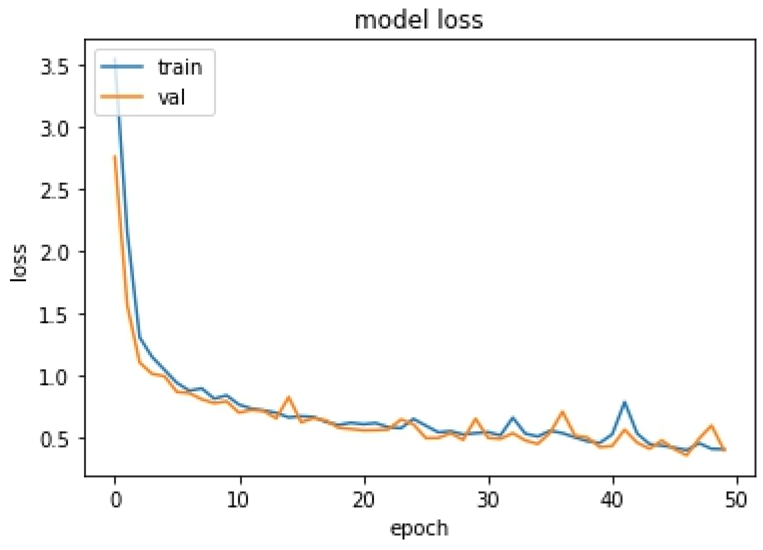


Figure 4:Learning curve with training and validation loss

From the above graph of training and validation loss, we can see that there is a sharp decrease from 0 to 6 epochs and then loss starts to reduce gradually after passing 30 epochs the fluctuations start to increase but the loss starts to converge nearly at 50 epochs.

At the testing stage the mean and standard deviation of output was calculated

```
print(Y_pred_m)
```

```
[[-0.23501783  0.10036119 -0.75275215 -0.67349012]
 [ 0.34551202  0.42157623  0.69059524  0.28847312]
 [ 0.36616508  0.02886915 -0.23361534  0.03676619]
 ...
 [ 0.02028204  0.08881977 -0.38781873 -0.25058595]
 [ 0.54191342  0.27953013  0.86106182  1.04117188]
 [ 0.02294657  0.05773167 -0.56421321 -0.44778901]]
```

```
print(Y_pred_s)
```

```
[[0.73046874 0.02757556 0.63145414 0.77617644]
 [0.22330039 0.06643206 0.42344558 0.32133847]
 [0.73705997 0.02250191 0.34778324 0.42506272]
 ...
 [0.4439324  0.02878268 0.51388681 0.60425109]
 [0.30389321 0.05899884 0.50908076 0.44051859]
 [1.47389827 0.03807916 0.89309665 1.09108708]]
```

4 Conclusion:

In Air quality of reference analyzer is being predicted using Bayesian neural network that performs better as compare to simple approaches. From the above discussion, we can see that the Bayesian neural network is a combination of a Bayesian model with the neural network so that the power of both can be integrated for useful purposes especially in the statistical domain.

It is quite clear from the above experiment that the Bayesian approach is better than the deterministic one because of the addition of uncertainty information. But along with advantages, it has some disadvantages and one of the main ones is its computational cost. Recent researches are focused on overcoming this limitation.

5 Reference:

[1] https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_neural_networks.htm

[2] https://www.kdnuggets.com/2017/11/bayesian-networks-understanding-effects-variables.html

[3] Laurent Valentin Jospin, Wray Buntine, Farid Boussaid, Hamid Laga, and Mohammed Bennamoun. 2020. Hands-on Bayesian Neural Networks - a Tutorial for Deep Learning Users. ACM Comput. Surv. 1, 1 (July 2020)

[4] https://www.sas.com/en_us/insights/analytics/neural-networks.html

[5] David J. C. MacKay. A practical Bayesian framework for backpropagation networks. Neural Computation, 4(3):448–472, 1992.

[6] John Mitros and Brian Mac Namee. On the validity of Bayesian neural networks for uncertainty estimation. In AICS, 2019.