Expert System using Bayesian Neural Networks

Sathuri Bharath Kumar Goud[1]

[1]Reg No. CS20M011 | cs20m011@iittp.ac.in

22^{nd} Dec 2021

Abstract

An Expert System can be defined as a knowledge-base system that acquires knowledge from its knowledge base of a specific domain and addresses user queries using its inference engine about that specific domain. In the Expert System, we have 3 main parts(knowledge base + inference engine + user interface). It has many applications in various domains. One of the important domains is predictive maintenance where it helps in monitoring the health of machinery and helps us to get quick repair solutions with its knowledge and improve its lifetime. We are using Bayesian Neural Networks(these are Neural networks in which we use probabilities for weights and biases for prediction) as a part of the inference engine of the expert system which helps to overcome the uncertainties from general Neural Networks and gives confident and dynamic predictions. This yields us even much better results and helps in effective monitoring of the machinery both by individuals and organizations.

Here, we present the usage of Bayesian Neural Networks(BNN's) for building Expert Systems. The usage of BNN's overcomes the drawback of the uncertainty of neural networks. It also makes predictions more dynamic.

Midterm MTP Report

Dept of Computer Science and Engineering
Indian Institute of Technology Tirupati

Evaluation Pannel

		Grading	Signature
Supervisor:	Prof Dr. Srinivas Padmanabhuni		
Member:			
Member:			
Member:			
Member:			

Contents

1 Introduction

The Expert System is a very old concept in the field of AI. Many expert systems are built for various domains and applications ranging from the First Developed MYCIN Expert system for medical diagnosis to different machinery repair expert systems, credit card fraud detection systems, bug reporting systems, etc. Even though the applications are different, the underlying concept is the same. Every Expert system has the following components. Knowledge base, Inference Engine, and User(User Interface).

1.1 Expert System

Definition: Expert system can be defined as a knowledge-based system where it acquires knowledge about a particular(specific and narrow) domain with the help of an expert from that domain and uses its inference engine(here it's BNN and not rule-based ones) to address user queries. The user queries the expert system with the user interface provided to the user. It takes this input, processes it with the help of its domain knowledge and its inference engine, and results in the answer/output to the query.

The advantage of this expert system is that the knowledge base can be updated as per need to make the expert system with up-to-date knowledge. This system simulates human(expert) intelligence. It assists an expert in making better decisions but it is not a replacement. The advantage is that these systems are consistent in their predictions with better accuracy and are fast compared to humans.

The main disadvantage of these systems is the Knowledge acquisitions part.

1.2 Neural Networks

Neural Networks: These are the basic building blocks of the Deep learning domain. When compared to traditional learning approaches, we don't need to tell how to learn the data to the neural network. It automatically learns from the given observation(training) data, extracts the patterns, and applies the new data for predictions. One of the main disadvantages is that these are overconfident in their predictions which leads to uncertainty. So to overcome this and to have more dynamic predictions we are using Bayesian Neural networks where we use Bayesian inference in neural networks which gives us confident predictions and with certainty.

1.3 Bayesian Neural Networks(BNN)

BNN: The authors of the paper [8]"Hands-on Bayesian Neural Networks - a Tutorial for Deep Learning Users" defined it as a stochastic ANN trained using Bayesian inference. In simple terms, we use the probability of weights and biases and other parameters instead of singular values to the above terms. In the same paper as above, Stochastic ANN's are defined as ANN's in which stochastic components are used to simulate multiple possible model parameters along with their probability distributions. This can be also considered as a special case of Ensemble learning(multiple different models combined to have the most accurate predictions).

1.4 BAYESIAN NEURAL NETWORKS

1.4.1 Introduction

The Bayesian paradigm which is the base for Bayesian neural networks work on two basic and simple ideas,

- Probability of an event is defined as the belief in the occurrence of that event.
- Prior beliefs(probabilities) influence posterior beliefs(probabilities). This is best explained by the Bayes theorem.

Bayesian Neural Network: BNN = Stochastic Neural Network + Bayesian Inference.

1.4.2 Implementation

Stochastic Neural Network = ANN + stochastic components(using probability distribution p(θ) which is useful to simulate multiple models similar to the process of ensemble learning. BNN's are better at quantifying the most consistent uncertainty for their predictions when compared to ANN's. Without over fitting the data, they are also efficient for small datasets.

- They can be considered as a special case of ensemble learning.

  Ensemble learning: it's a technique where we use multiple models instead of a single one and the combination of them will work with much better accuracy.

- Stochastic models are implemented using probabilistic graphical models(PGM's). After this, the initial priors are set in the network using various techniques.

- Various Bayesian inference algorithms are MCMC(Markov chain Monte Carlo) and variational inference, Bayes by back-propagation, learning the priors method.
- In practice, it's hard to implement the exact algorithm for the implementation of BNN's, so it's a general practice to approximate it with various techniques.

  some of them are:

  1. Bayes via Dropout
  2. Bayes via Stochastic Gradient Descent
  3. Bayesian inference on the last n layers of the network.

1.4.3 Conclusion

Even though the overall idea might be easy where we are using probability distributions of weights in the network, the actual algorithmic implementation is difficult.

1.5 EXPERT SYSTEM ARCHITECTURE

An Expert system is an intelligent system that is based on the knowledge base and uses an inference engine to solve problems. The knowledge base would be restricted to a specific and narrow domain and with the help of an inference engine, it solves problems or answers user queries related to that domain. It also has a heuristics engine which helps in better predictions. These machines help the domain expert in taking better decisions. These have a wide range of domain applications.

Expert System = knowledge base + inference engine + user interface(optional)

1.5.1 Knowledge Base

Knowledge base: It's the strength of the expert system as the expert system derives its power from the knowledge base.

Knowledge base = organized collection of facts + heuristics knowledge

The process of building a knowledge base by getting the knowledge base from the domain expert along with various other sources is known as **knowledge engineering.** The way we arrange or organize the knowledge in the knowledge base is defined as **knowledge representation**.

Methods:

- **Frame-based systems:** In this, frames represent the attributes or properties of the objects and relationships among them.
- **Production rules:** the most commonly used method where we represent knowledge in the form of production rules.

  Similar to if-else conditional rules.

1.5.2 Inference Engine

Inference engine: It combines all the knowledge from the knowledge base it has and generates its predictions for the specified input provided with the help of the user interface provided to the user. Here, in this project, we are using Bayesian neural networks as the inference engine of this expert system to get the predictions with confidence and to avoid uncertainty from general neural networks.

1.5.3 User Interface

User Interface: it's an optional one where we can use this to test the expert system by any user by providing them with the interface. Or the testing can be manually done by the developer with the inputs to the system.

1.5.4 Conclusion

The overall stages of an Expert system are:

1. Identifying the domain for the Expert system.
2. Designing the system and developing it.
3. Testing and refining it.
4. Deploying and maintaining the system.

The main limitations of Expert system are the domain expertise and knowledge acquisition.

1.6 ADVANTAGES

The main advantages of Expert systems using BNN's are that

- An Expert System with better performance and confident predictions.
- Useful for experts in decision making and applicable across various other domains.

1.7 LIMITATIONS

The main disadvantages of Expert systems using BNN's are that

- Domain expertise availability.
- Extract and up-gradation of knowledge from expert.
- The high complexity of Bayesian neural networks implementation in practice increases the cost of the project to overcome the uncertainties

2 Existing Works

2.1 Expert System

The papers discussed for the literature review are :

[1]"An Expert System for Automobile Repairs and Maintenance"

In this, the authors have designed an expert system that can detect various automobile problems instead of using a human expert for the same requirement. It's a complete GUI based expert system where a user can check the possible faults with causes and solutions to them and also diagnose the car with possible symptoms, causes and solutions to those problems.

[2]"An Expert System for Fault Diagnosis, Repairing and Maintenance of Electrical Machines"

In this, the authors have developed an expert system software tool that uses an SQL database with different access levels and with the use of suitable weights for the questions to define their priority. For this, a simple user interface is developed which can be used by all to effectively monitor the condition of electrical machines.

[3]"Development of an Expert System for Maintenance and Repair of Masonry Barrages"

In this, the authors focus on building an expert for the domain which has very limited experts so as to carry forward the knowledge to others and also in better maintenance and early monitoring of barrages for repair to improve their life. In this, the expert system consists of only a knowledge base and inference engine.

[4]"An Expert System for fault Diagnostics in Condition-based Maintenance"

In this, the authors discussed condition-based maintenance which helps in monitoring the health of the machine and its various critical components, monitors its performance and help in the early repair of the machine which reduces a great amount of repair cost, logistic cost etc. Data Acquisition, signal processing, condition monitoring, health assessment, prognostics and decision support were the stages involved in this expert system.

[5]"Machine Learning in Predictive Maintenance towards Sustainable Smart Manufacturing in Industry 4.0"

In this, the authors have discussed Predictive Maintenance and its importance in the present industry 4.0. Various Machine learning techniques have been discussed in this paper which can help in fault detection and diagnosis which helps to have fewer failures and more usage of the machine.

[6]"Development of an expert system for the repair and maintenance of bulldozer's work equipment failure"

In this, the authors have developed an expert system for the repair and Maintenance of bulldozer's equipment, which gives the solutions to the various problems like low or high hydraulic pressure, abnormal noise etc and helps in getting repaired quickly improves its performance and its life-cycle.

2.2 BNN

The papers discussed for the literature review are :

[7]"Hands-on Bayesian Neural Networks - a Tutorial for Deep Learning Users"

In this, the authors describe quantifying the uncertainty of deep learning models with the help of Bayesian inference and how it can be implemented with the help of Bayesian neural networks. They have described in detail all the steps involved in constructing the Bayesian neural networks and their practical implementations.

[8]"Explaining Bayesian Neural Networks"

In this, the authors discussed the limited transparency of BNN's and two perspectives of transparency about BNN's and their proposed way of explaining BNN's will help to get the most effective and insightful explanations. They have used various evaluation procedures and experimented with different datasets.

[9]"A Bayesian Approach to Invariant Deep Neural Networks"

In this, the authors describe learning invariances directly from data with the help of posteriors distributions over various weight sharing schemes. and is outperforming non-invariant architectures. They have experimented with the different forms of the data.

[10]"Efficacy of Bayesian Neural Networks in Active Learning"

In this, the authors discussed active learning and the efficiency of Bayesian neural networks for active learning. They found that BNN's are more efficient than ensemble learning methods in capturing the uncertainty.

In this, they experimented and compared various other techniques along with BNN's and with the use of various other datasets.

[11]"Approximate Bayesian Optimisation for Neural Networks"

In this, the authors discussed the option of exploring with BNN's instead of GP's(Gaussian Processes) for building the model distributions for given functions and how it improved the performance and efficiency of the system. They have experimented and bench-marked the effect of priors on BNN's with HPO Benchmark.

[12]"Structured Dropout Variational Inference for Bayesian Neural Networks"

In this, the authors proposed a new technique called Variational Structured Dropout which is inspired by the Bayesian interpretation of the Dropout regularization technique. In this, the VSD technique introduces an adaptive regularization term with various desired properties which increases the model to make more generalized predictions. Besides this, they have discussed the scalability of their VSD technique and explicit regularization of the Variational Structured Dropout(VSD) technique.

[13]"Bayesian Neural Networks: Essentials"

In this, the author explained the Bayesian neural networks and how they are built and implemented with examples. They concluded that in practice, it's very inefficient to use due to its cost for complexity in implementing for all layers in the network, instead, using a variation of BNN mostly hybrid ones are more suitable in practice. The author described Bayesian inference using Markov Chain Monte Carlo(MCMC) inference and variational inference techniques and the use of Gaussian priors and posteriors besides bayesian priors and posteriors which are used in general.

3 Problem Definition and Preliminary Experiments

3.1 Problem Definition

We are developing an Expert system by using Bayesian Neural Networks as a part of its inference engine. Why use Bayesian Neural Networks when we can do it with Feed-Forward Neural Networks (General ones), It is because normal Neural Networks fit over the given data and becomes overconfident about their predictions. This causes them to be uncertain. To overcome this, we are implementing Bayesian Neural Networks for developing an inference engine for our expert system. These BNN's predicts with certainty and are dynamic and confident over their predictions. They achieve it by using probabilities for weights and biases instead of only integers.

3.2 Dataset

The dataset for this project is taken from UCI Dataset Repository[14] titled " AI4I 2020 Predictive Maintenance Dataset Data Set" provided by Stephan Matzka, School of Engineering - Technology and Life, Hochschule Wirtschaft Berlin, 12459 Berlin, Germany. It contains data of 10000 rows 1 for each data point and 14 columns 1 for each feature. The target variable 'machine failure' indicates whether the machine has failed for that particular machine point in the dataset only if any one of the independent failures has occurred for that machine.

4 Implementation

4.1 Data Description

Unique values in each attribute of the dataset:

UDI: 10000, Product ID: 10000, Type: 3, Air temperature [K]: 93, Process temperature [K]: 82, Rotational speed [rpm]: 941, Torque [Nm]:577, Tool wear [min]: 246, Machine failure: 2, TWF: 2, HDF: 2, PWF: 2, OSF: 2, RNF: 2.

datatype: int64

4.2 pre-processing data

Pre-processing the data:
Ordinal Encoding:
df['Type'].unique() : array(['M', 'L', 'H'], dtype=object)

- L, M, H are three types representing low (50% of all products), medium (30%), and high (20%) as product quality variants respectively.
- Converting this categorical data to numerical with class 0, 1, 2 for L, M, H respectively using OrdinalEncoder from the sklearn library.
- One-hot encoding is not suitable for ordinal data so we use an Ordinal encoder.
- This gives categories converted into integers.

- This sorts all the categories present and assigns values to them in alphabetical order.
  0 for H
  1 for L
  2 for M

Getting all numerical data: 'Type', 'Air temperature [K]', 'Process temperature [K]', 'Rotational speed [rpm]', 'Torque [Nm]', 'Tool wear [min]', TWF', 'HDF', 'PWF', 'OSF', 'RNF','Machine failure'

Creating training and evaluation datasets : Using the $train_{t}est_{s}plit$ function, I split the data into training and testing in the ratio of 70% - 30%.

4.3 Various plots

All these are the plots that are plotted during the implementation of the project.

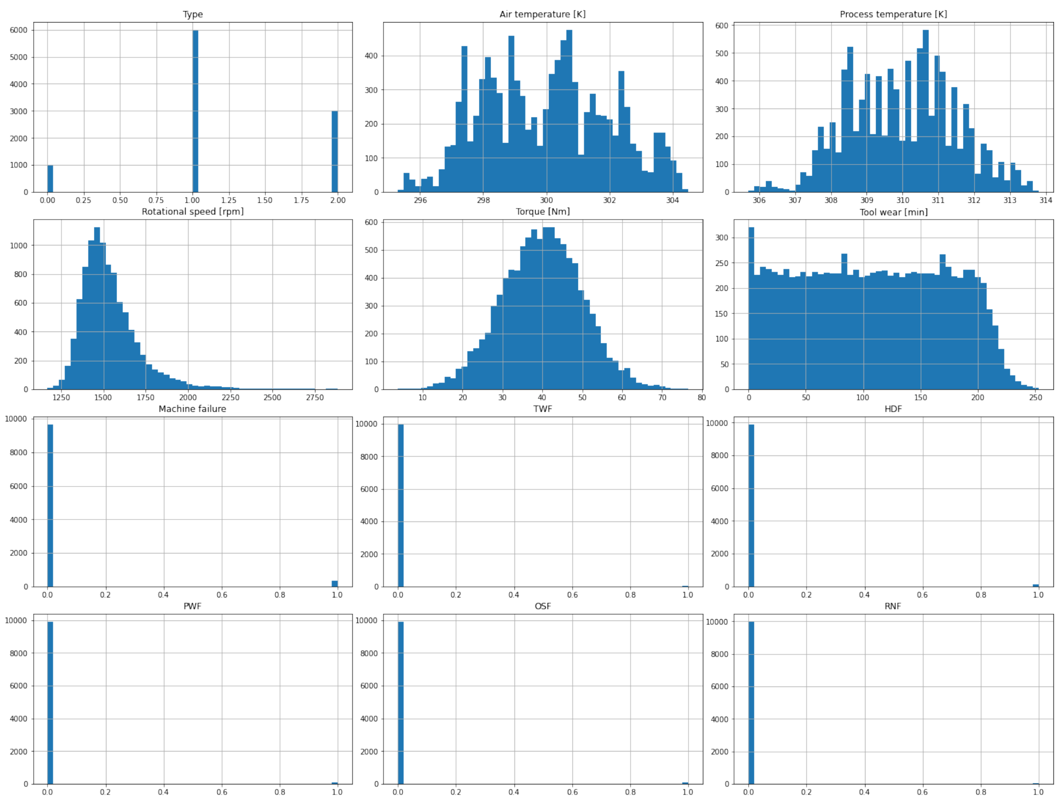


Figure 1: Histograms of All Attributes.

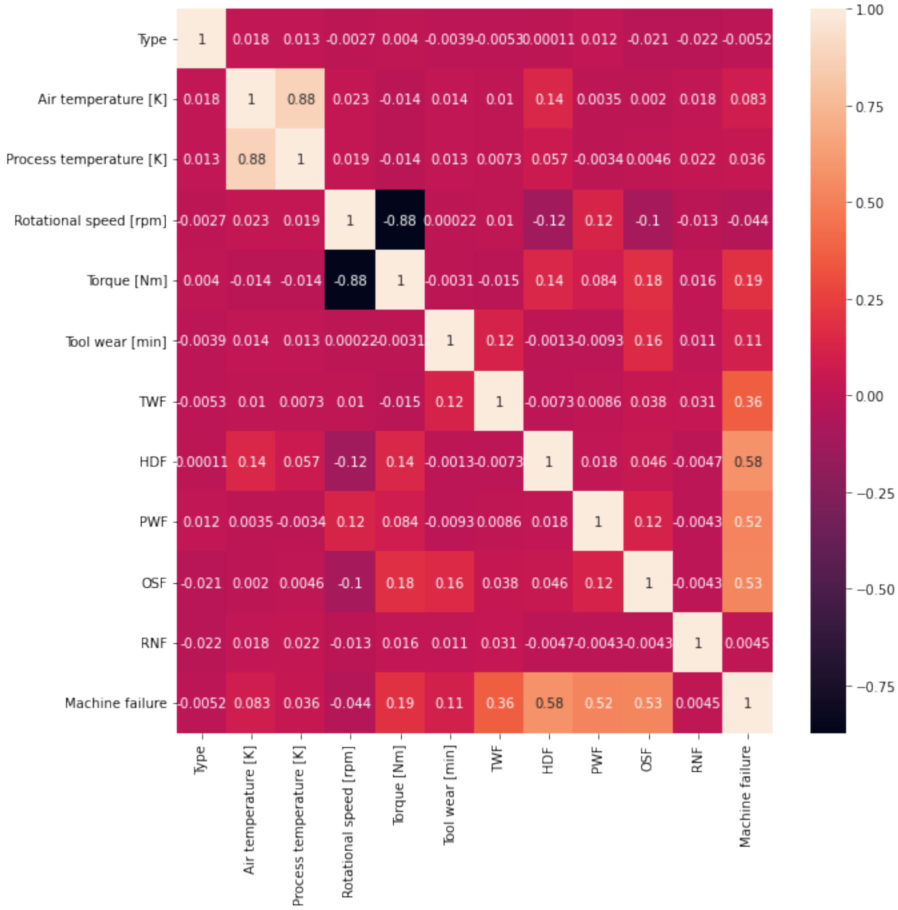
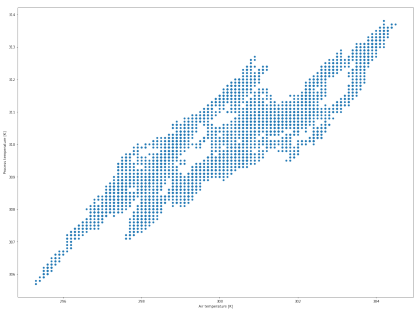
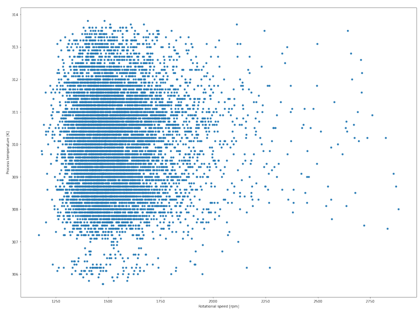


Figure 2: Heatmap representing Correlation Matrix

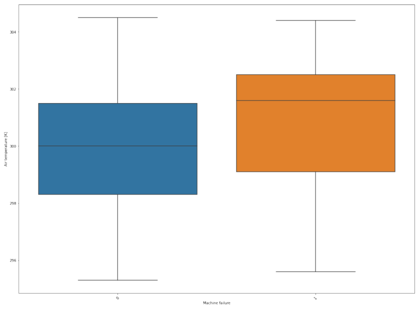


(a) Scatter Plot of process and Air temperatures

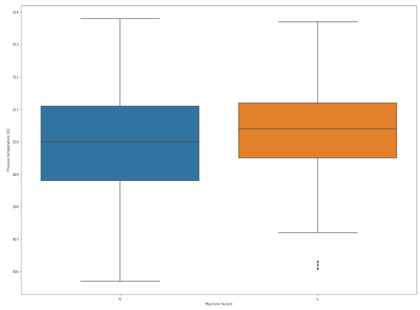


(b) Scatter Plot of process and Rotational Speed

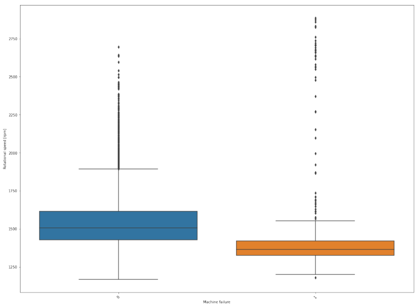
Figure 3: Scatter Plots



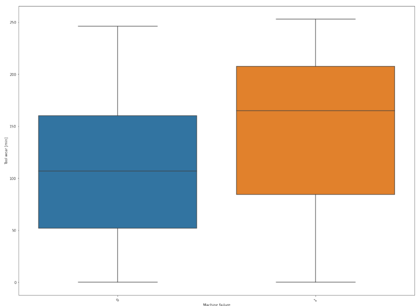
(a) Box plot of Air temperature and Machine Failure.



(b) Box plot of Process temperature and Machine Failure.

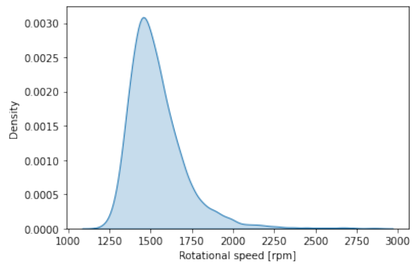


(c) Box plot of Rotational Speed and Machine Failure.

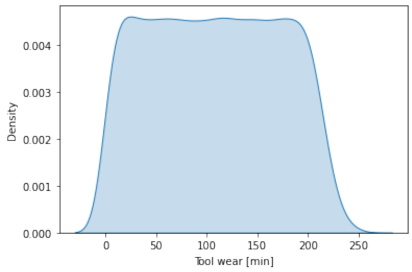


(d) Box plot of Tool wear and Machine Failure.

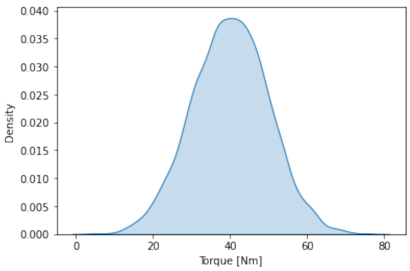
Figure 4: Box Plots



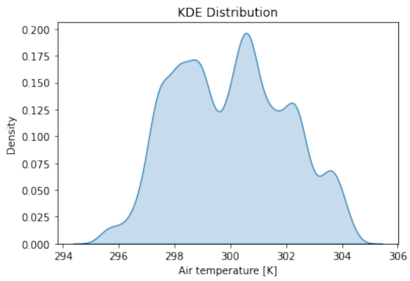
(a) KDE plot of Rotation Speed



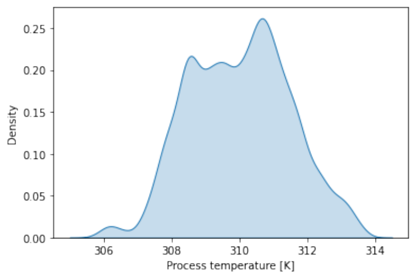
(b) KDE plot of Tool Wear.



(c) KDE plot of Torque.



(d) KDE plot of Air Temperature.



(e) KDE plot of Process Temperature.

Figure 5: KDE Plots

4.4 Implementing a Neural Network

Standard Neural Network:

No of epochs: 50
Learning rate: 0.001
Test : Accuracy: 0.9690
Test loss: 0.1385
Layers: Dense size = 5 ; parameters = 65
Dense size = 3 ; parameters =18
Dense size = 1 ; parameters =4
train accuracy: 0.9649, loss: 0.1522 after 50 epochs
test accuracy: 0.9690, loss: 0.1385
Model: "sequential"

Layer (type)	Output Shape	Param
dense (Dense)	(None, 5)	65
dense_1 (Dense)	(None, 3)	18
dense_2 (Dense)	(None, 1)	4

4.5 Implementing a BNN

Bayesian Neural Network (BNN):

No of epochs: 50
Learning rate: 0.001
Test : Accuracy: 0.9686
Test loss: 0.450
Test accuracy: 0.9686 after 50 epochs and test loss: 0.450
Model: "sequential_2"

Layer (type)	Output Shape	Param
dense_flipout (DenseFlipout)	(None, 16)	368
dense_flipout_1 (DenseFlipout)	(None, 6)	198
dense_flipout_2 (DenseFlipout)	(None, 3)	39

4.5.1 Packages used

Tensorflow-probability: TensorFlow Probability (TFP) is an open-source Python library that is built on the TensorFlow framework that makes it easy to combine probabilistic models and deep learning on modern hardware (TPU, GPU). Since TFP is built on TensorFlow, we can build, fit, and deploy a model using a single language throughout the lifecycle of the model.

5 Timeline and Execution Plan

The overall timeline and the execution of the project.

3rd Semester Endterm: Developing a Bayesian Neural Network.

4th Semester Midterm: Different variations of BNN's developed for Expert System.

4th Semester Final evaluation: Full Project and Final Report to be submitted by the end of the 4th Semester. (Mtech 2nd year)

6 Individual Contributions

Only 1 student(me) in my project.

7 Conclusion

Expert system has many advantages compared to its limitations. So to overcome its limitations, we can use various practical implementation techniques of Bayesian neural networks as a part of its inference engine which approximates them yet gives better results.

8 Acknowledgements

I am thankful to my guide Dr. Srinivas Padmanabhuni for his continuous support for the project.

I have taken the dataset for this project from UCI Dataset Repository [14] titled " AI4I 2020 Predictive Maintenance Dataset Data Set" provided by Stephan Matzka, School of Engineering - Technology and Life, Hochschule fÃ¼r Technik und Wirtschaft Berlin, 12459 Berlin, Germany.

References

[1] A. A. Adekunle, P. P. Ikubanni, and O. O. Agboola, "An expert system for automobile repairs and maintenance," *Mindanao Journal of Science and Technology*, vol. 16, no. 1, 2018.

[2] V. Kontargyri, C. Tsirekis, S. Kabanaros, A. Sakarellos, A. Moronis, N. Kolliopoulos, and S. Kaminaris, "An expert system for fault diagnosis, repairing and maintenance of electrical machines," in *Proceeding of*

the 6 th WSEAS International Conference on Applications of Electrical Engineering. Citeseer, 2007, pp. 27–29.

[3] H. Abd-Elhamid, A. Javadi, A. Negm, A. Elalfi, and T. Owais, "Development of an expert system for maintenance and repair of masonry barrages," *Intelligent Computing in Engineering–ICE08*, pp. 212–221, 2011.

[4] E. A. Simeón, A. J. Álvares, and R. R. Gudwin, "An expert system for fault diagnostics in condition based maintenance," in *ABCM Symp. Ser. Mechatron*, vol. 4, 2010, pp. 304–313.

[5] Z. M. Çınar, A. Abdussalam Nuhu, Q. Zeeshan, O. Korhan, M. Asmael, and B. Safaei, "Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0," *Sustainability*, vol. 12, no. 19, p. 8211, 2020.

[6] O. B. Akinluli, V. A. Balogun, and T. M. Azeez, "Development of an expert system for the repair and maintenance of bulldozer's work equipment failure," *International Journal of Scientific & Engineering Research*, vol. 6, no. 6, pp. 1–14, 2015.

[7] L. V. Jospin, W. Buntine, F. Boussaid, H. Laga, and M. Bennamoun, "Hands-on bayesian neural networks – a tutorial for deep learning users," 2020.

[8] K. Bykov, M. M.-C. Höhne, A. Creosteanu, K.-R. Müller, F. Klauschen, S. Nakajima, and M. Kloft, "Explaining bayesian neural networks," *arXiv preprint arXiv:2108.10346*, 2021.

[9] N. Mourdoukoutas, M. Federici, G. Pantalos, M. van der Wilk, and V. Fortuin, "A bayesian approach to invariant deep neural networks," *arXiv preprint arXiv:2107.09301*, 2021.

[10] V. Rakesh and S. Jain, "Efficacy of bayesian neural networks in active learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2601–2609.

[11] N. Hassen and I. Rish, "Approximate bayesian optimisation for neural networks," *arXiv preprint arXiv:2108.12461*, 2021.

[12] S. Nguyen, D. Nguyen, K. Nguyen, N. Ho, K. Than, and H. Bui, "Structured dropout variational inference for bayesian neural networks," 2021.

[13] D. T. Chang, "Bayesian neural networks: Essentials," *arXiv preprint arXiv:2106.13594*, 2021.

[14] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml