Comments and Suggested Corrections

General Comments and Suggestions

The thesis surveys some common neural network models with an emphasis on the Bayesian regularized neural networks. Here are several general comments.

1. Create a list of abbreviations for notations and technical terms used in the draft.
2. Many articles were cited in the survey, and some notations were used with no definition or description which makes some parts of the draft hard to understand.
3. Many different methods in statistics and machine learning fields were surveyed, and some of the descriptions are not accurate. I also suggest summarizing these different methods and algorithms based on reliable sources.
4. I did not see any comparisons between different models in the example (earthquake) using the performance measures defined based on the testing data set. This seems to be the most important part of the example.
5. It seems that you are particularly interested in predicting the frequency at magnitude 9.1. This is a rare event prediction problem. Since not much information is available near $x = 9.1$ and the sample size is also very small, no existing algorithms can do well in this situation. Only very few existing algorithms with special treatments such as penalization, weighting, etc. can handle rare event prediction but with various strong assumptions. The NN models you surveyed are not specifically designed for the rare event prediction.

Specific Comments and Suggestions

1. [page 5]. The reason for using the activation function NN is not correct. First of all, the returned binary output value is due to the default special linear activation function, *Heaviside step function*, with a decision boundary M in the definition. The major reason for using the various activation function in a neural network model is to produce a non-linear decision boundary via non-linear combinations of the weighted inputs.
2. [page 5] The description of the relationship between NN and linear and logistic regression models should be more rigorous. For example, *devoid of a hidden layer or activation function* is technically incorrect since any NN model must have three components: input layer, hidden layers/neurons (along with an activation function), and output layer. The connection between regression models and NN models is determined by the types of activation and the number of hidden layers.
3. [page 6]. *The description of the cost function is inaccurate.* Two major types of cost functions are commonly used in practice: Error-based loss function and information-theoretic loss function. *negative log-likelihood* is the information-theoretic loss (i.e., entropy). the MSE is an error-based loss function.
4. [page 7]. The definition of **MCE** required a constraint: the sum of all category probabilities must be equal to 1.
5. [page 7]. The statement *The efficiency of a neural network is measured and corrected with each iteration based on reducing a cost function.* is unclear. **Gradient descent** is only an optimization algorithm, it will not impact the efficiency of the NN model. The efficiency of an NN model is determined by the architecture of the NN itself.

6. [page 8]. A typo in the definition of C: it should $C = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$. Also, need to tell what is N.

7. [page 9]. The description of *stochastic gradient descent* is not correct. It samples observations randomly (one at a time) to estimate the parameters and update the estimate by other samples in the iterative process.

8. [Page 10] *The kernel, which is much smaller than the image, moves across the image and performs its calculation* - this description of the kernel function is not correct. It is a kind of weighting function that put different weights on the inputs. A discrete kernel is usually represented in the form of a matrix (i.e., a linear mapping function) that gives the weights explicitly in its cells. By the way, the concept of the kernel function originated from statistics.

9. [page 11]. *The measurement of error that is reduced during optimization is known as the training error.* - again, optimization will not reduce the error. The end product of an optimization problem is the estimated values of the unknowns that minimizes the objective (loss) function within a given tolerance error bound.

10. [page 11] Comment: R^2 is rarely used to rate NN models. It is even not reported in NN models with obvious reasons.

11. [page 11] You mentioned *spline* in the section of Rating Model Performance. Do you mean *spline regression* or *spline approximation of an unknown function* or something else?

12. [Page 11] The statement *Because a neural network model has many more parameters to be estimated than its ordinary least squares counterpart,...* is not quite correct. The *least square* is only a general optimization method (it uses numerical algorithms such as gradient descent, Newton, etc., to find the solution to the least square problems).

13. [page 11] The statement *The ensemble of techniques to calibrate the model toward both training data and new data is known as regularization* is also not accurate. A regularization method can be considered as an ensemble method, but not vice versa.

14. [page 12] The statement *If the gradient descent algorithm is used to reduce the following error function of the data* is incorrect. *gradient descent algorithm* is a numerical algorithm just like the Newton–Raphson method that is used to find the solution to the minimization problem. It does not reduce any errors in the model.

15. [page 12] The description of the *delta method* in the section of Quantification of Uncertainty is not quite correct. In fact, it is simply a linear approximation method that converts (approximate) a nonlinear problem to a linear problem to solve.

16. [page 12] The statement *Delta also assumes normality and homogeneity, which limits its use in practice.* is wrong. the delta method is a mathematical approximation as commented in 16, is nothing to do with probability distribution. It can definitely be used to linearize a function of random variables so one can easily find the approximated variance-covariance matrix.

17. [page 13] The description of the bootstrap method is inaccurate. In fact, bootstrap is a simulation method that uses the data of a random sample as a surrogate population to approximate the sampling distribution of a statistic. I suggest to rewrite this section to provide a more accurate description.

18. [page 13] *The first is to generate bootstrapped pairs by sampling with replacement of the original training data and generate estimates from the trained neural network on each iteration of bootstrap pairs* - The description of this first method is not quite clear to me. It will be help to write an algorithm like various algorithms (e.g, MCMC) in the Bayes calculation.

19. [page 13] **General comments on Earthquake example**: The research question should be clearly stated before any analysis. The feature variables and sample size should also be specified. For ease of comparison, the training-testing mechanism should be used in regression models so you can define data-driven metrics to assess the performance of variables models and algorithms.

20. [pages 13 - 14] The linear regression model is inappropriate since the response variable is a (relative) frequency variable. A frequency regression model such as Poisson, negative binomial, etc. The p-values in the output are based on the t-distribution which is defined with the assumption of normal residuals (hence, the normal response). Similarly, polynomial regression is also inappropriate.

21. [page 15] **General comments on MLP**: For MLP, your readers expect information about (1) feature variables in the input layers; (2) the architecture of MLP including hidden layers and activation functions, etc. (3) justifications for the determination of the number of hidden layers, neurons in each layer, and activation(s), (4) same or different activation functions used across hidden layers. I suggest revising this part.

22. [page 15] The visual representation of MLP should include the predicted values and observed values in the test data set for the purpose of comparison. That is why a significant amount of time needs to be put into model training and identification.

23. [page 15 - error table]. In general, testing errors are bigger than the training error. However, your testing errors are consistently smaller than the training error! How to justify this if there is no computational error in your code?

24. [page 19] Statement *A nice feature of Bayesian Statistics is the ability to make sense out of few data points, without the need for a minimum sample size or rules of thumb.* is not correct since the sample size impacts both variance and bias (hence, the power of inference). The benefit of Bayes is its ability to use prior information (sometimes auxiliary information. In fact, the large sample size can reduce the impact of the misspecified prior.

25. [Page 19] Statement *Rather than estimating all unnecessary variables, they are integrated out* is misleading. The common practice is to integrate only nuisance parameters out.

26. [page 20] Some of the notations adopted from references are not predefined. For example, what is k in $Z(\alpha, \beta) = \int d^k w e^{-(\alpha E_W + \beta E_D)}$?

27. [page 22] Statement *A beginner's start for a regression task is to select a normal prior* is inappropriate. A better statement is **A beginner's start for a *linear* regression task is to select a conjugate prior (in normal linear regression, the conjugate prior is normal distribution)**. The reason is that in a general regression model, you may deal with different types of parameters such as shape and scale in which conjugate prior will be some distributions other than the Gaussian (e.g., inverse-gamma). For an experienced analyst, the choice of prior should be based on the prior knowledge of the parameters.

28. [page 23]. No pre-defined notations were used: for example Φ_θ, Θ, $|\Theta|$, etc. All notations in the thesis should be clearly defined before using them.

29. [page 24] **General comments on BRNN with Earthquake Data**: Need to specify the architecture of the NN model. Particularly the information of hidden layers, neurons, activation function (by default, tanh is used in the library) and priors, etc.

30. [page 24] *The model is a 6-layer Bayesian Regularized Neural Network (BRNN) to make a prediction for a magnitude 9.1 earthquake in the Tohoku area* - the help document says that the library only defined NN with only two layers. The subtitle of Figure 15 mentions a 3-layer.

31. [page 24] Why did not use and report any performance measures from testing data? This should be the key information to be reported in any analysis.

32. [page 25] *Figure 16 displays 100 individually trained networks of the same architecture* - what does *individually trained of the same architecture* mean? what samples (observations) were used in these 100 individually trained NN? You also mentioned **prediction curvature**? the curvature of what? The example is about prediction. Need some explanations and justifications.

33. [Page 33] Code for MLP: you did not use the random split. **As a common practice, one should always use RANDOM splits**. Another comment is the use of random seed in the simulations: make sure the seed changes in different random number generation steps. Why not present a visual

representation of the architecture of the underlying NN with 2 hidden layers (also only 2 hidden layers and 5 neurons in each layer)? Tried the cross-validation method to find a better number of hidden layers and neurons?

34. [Page 35] BRNN based on the `brnn` library has only two hidden layers. not 3.