2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA) | 978-1-6654-4337-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICMLA52953.2021.00172

Theory for Deep Learning Regression Ensembles with Application to Raman Spectroscopy Analysis

1st Wenjing Li
Mathematical Department
Worcester Polytechnic Institute
Worcester, United States
wli5@wpi.edu

2nd Randy C. Paffenroth
Mathematical Department
Worcester Polytechnic Institute
Worcester, United States
rcpaffenroth@wpi.edu

3rd Michael T. Timko
Chemical Engineering Department
Worcester Polytechnic Institute
Worcester, United States
mttimko@wpi.edu

4th Matthew P. Rando
Chemical Engineering Department
Worcester Polytechnic Institute
Worcester, United States
mprando@wpi.edu

5th Avery B. Brown
Chemical Engineering Department
Worcester Polytechnic Institute
Worcester, United States
abbrown@wpi.edu

6th N. Aaron Deskins
Chemical Engineering Department
Worcester Polytechnic Institute
Worcester, United States
nadeskins@wpi.edu

***Abstract*—Regression ensembles consisting of a collection of base regression models are often used to improve the estimation/prediction performance of a single regression model. It has been shown that the individual accuracy of the base models and the ensemble diversity are the two key factors affecting the performance of an ensemble. In this paper, we derive a theory for regression ensembles that illustrates the subtle trade-off between individual accuracy and ensemble diversity from the perspective of statistical correlations. Then, inspired by our derived theory, we further propose a novel loss function and a training algorithm for deep learning regression ensembles. We then demonstrate the advantage of our training approach over standard regression ensemble methods including random forest and gradient boosting regressors with both benchmark regression problems and chemical sensor problems involving analysis of Raman spectroscopy. Our key contribution is that our loss function and training algorithm is able to manage diversity explicitly in an ensemble, rather than merely allowing diversity to occur by happenstance.**

***Keywords*—theory, application, accuracy, diversity, correlation, algorithm, deep learning regression ensemble.**

I. Introduction

Ensemble learning is a process by which a collection of base learners are strategically generated and combined into one composite learner [1]. The primary goal of ensemble algorithms is to improve the performance (classification, prediction etc.) of single models. Ensemble algorithms have been applied to a variety of machine learning domains including text mining [2], recommender systems [3], and many others. Regression ensemble algorithms take the combined predictions from constituent regression models as the ensemble prediction, and a (simple/weighted) average is a popular combining strategy [4]. Bagging [5], boosting [6] (including well-known Gradient Boosting [7] and Adaboost [6]), and stacking [8] are classic examples of ensemble algorithms.

One classic result that justifies the effectiveness of regression ensemble methods is the Ambiguity Decomposition [9], in which the authors proved that at a single data point the quadratic error of the ensemble estimator is guaranteed to be less than or equal to the average quadratic error of the component estimators. Therefore, Ambiguity Decomposition encourages regression ensembles as it tells us that taking the combination of several predictors would be better on average than selecting one predictor at random.

It has been shown that the individual accuracy of component learners and the diversity of the learners in the ensemble are the two key contributing factors to their performance [1]. Herein, building upon work on classification ensembles in the recent paper [10] [11], we develop a theory for understanding the subtle trade-off between individual accuracy and ensemble diversity in regression ensembles. Then, inspired by the theory for regression ensembles, we propose an algorithm for training regression neural network ensembles that can explicitly create diversity in ensembles, and manage the balance between individual accuracy and ensemble diversity.

Note that even though the theory we develop here for regression ensembles is an extension of the work on classification ensembles proposed in [10] [11], our work in this paper has application to a wide range of real-world problems. On one hand, in addition to the effectiveness on standard machine learning data sets (e.g. the UCI Parkinson's Telemonitoring data we analyze in Section IV), our proposed training algorithm for regression ensembles is also capable of handling difficult real-world chemical sensor problems as demonstrated by the quantitative analysis on Raman spectroscopy data detailed in Section V. On the other hand, in some sense regression problems are more complicated than classification problems, due to the variety of choices of training losses. In this paper, we compare a wide range of loss functions used when training regression ensembles. More importantly, we show that our ensemble training idea is generally applicable in that a classic loss function (e.g. L2 loss) can be combined with our proposed novel loss to generate a new loss which works well on regression ensembles.

In summary, we make the following novel contributions:

- We prove two theorems that bound the individual accuracy and diversity of regression ensembles in terms of the statistical correlations among the ground truth and the component learners.
- We propose a novel loss function and a training algorithm for regression neural network ensembles, following the inspirations of the theorems we develop.
- We show the effectiveness of our ensemble training approach with applications on standard machine learning data sets and challenging chemical sensor problems, and

978-1-6654-4337-1/21/$31.00 ©2021 IEEE
DOI 10.1109/ICMLA52953.2021.00172

demonstrate the advantage of our approach over standard regression ensemble methods.

II. Theory for Regression Ensembles

Authors of the recent paper [10] [11] have derived the theory for classification ensembles regarding the accuracy-diversity trade-off. In this paper, inspired by their work, we will investigate the meaning of diversity in regression context and develop theory that can help improve the performance of regression ensembles.

A. Diversity for Regression Ensembles

1) The Ambiguity Measure: In literature, the key point of diversity in classification ensembles has been introduced as "avoiding coincident errors" [12]. For regression ensembles, the idea is similar, that is, to make the errors made by the component regressors diverse, so it is natural to take quadratic error as a diversity measure. Following this idea, Krogh and Vedelsby proposed a well-known diversity measure for regression context called "ambiguity" [9]. In particular, the ambiguity is first defined for a single regression learner, then the ensemble ambiguity measure is obtained by averaging the single measures for all ensemble members.

2) Ambiguity Decomposition: Along with the ambiguity measure, Krogh and Vedelsby also proposed the well-known "Ambiguity Decomposition" [9] showing that at a single data point the quadratic error of the ensemble estimator is guaranteed to be less than or equal to the averaged quadratic errors of the component estimators, which illustrates the importance of having the right balance between diversity and individual accuracy in ensembles. The equation for "Ambiguity Decomposition" can be expressed as

$$(f_{ens} - d)^2 = \sum_i \omega_i (f_i - d)^2 - \sum_i \omega_i (f_i - f_{ens})^2, \quad (1)$$

where d is the true value, f_i are the component estimators, f_{ens} is the convex combination of the component estimators, that is, $f_{ens} = \sum_i \omega_i f_i$, and $\sum_i \omega_i = 1$.

The Ambiguity Decomposition equation in (1) tells us that the squared error of the ensemble estimator can be decomposed into the difference between two terms. The first term is the weighted average squared error of the individual estimators $\sum_i \omega_i (f_i - d)^2$, and the second term is the "Ambiguity term" $\sum_i \omega_i (f_i - f_{ens})^2$ which measures the amount of variability among the ensemble members for a particular input instance [13]. Therefore, the larger the Ambiguity term, the more variability (diversity) among the individual estimators. Since the Ambiguity term is always non-negative, the squared error of the ensemble estimator is guaranteed to be less than or equal to the weighted average squared error of the individuals, i.e.

$$(f_{ens} - d)^2 \leqslant \sum_i \omega_i (f_i - d)^2. \quad (2)$$

The inequality in (2) indicates that taking the combination of several estimators would be better on average, than randomly selecting one of the estimators. In addition, the larger the Ambiguity term, the larger the ensemble error reduction [13].

Based upon (1), we can see that in order to decrease the squared error of the ensemble estimator, we will need to decrease the weighted average squared error of the individuals, and increase the ambiguity (diversity) among the individuals. However, when the ambiguity term increases, the weighted average squared error of the individuals will also increase. Therefore, to lower the ensemble error, it is vital to have the right balance between diversity (the Ambiguity term) and individual accuracy (the weighted average squared error term).

B. Accuracy-Diversity Trade-off in Regression Ensembles from the perspective of Statistical Correlations

Given that the Ambiguity Decomposition has revealed the importance of having the right balance between individual accuracy and diversity in regression ensembles, we will now examine the mathematical relationship between them.

For regression ensembles of size N, we take the averaged learner-learner correlations $r_{LL}^{(ave)}$ as the measure for the ensemble diversity, where $r_{LL}^{(ave)}$ is defined as

$$r_{LL}^{(ave)} = \frac{1}{N(N-1)/2} \sum_{i=1}^{N} \sum_{j>i}^{N} r_{L_i, L_j}, \quad (3)$$

where r_{L_i, L_j} is the pairwise Pearson correlation coefficient [14] between the estimations of learners L_i and L_j.

In particular, we take the averaged truth-learner correlations $r_{TL}^{(ave)}$ as the measure for the overall individual accuracy of the regression ensemble, where $r_{TL}^{(ave)}$ is defined as

$$r_{TL}^{(ave)} = \frac{1}{N} \sum_{i=1}^{N} r_{T, L_i}, \quad (4)$$

where r_{T, L_i} is the Pearson correlation coefficient between the values of ground truth T and the estimations of learner L_i.

Note that the lower the $r_{LL}^{(ave)}$, the more diverse the regression ensemble, as the estimations from individual learners need to be negatively correlated in order to form a diverse ensemble [15]. And the higher the $r_{TL}^{(ave)}$, the higher the overall accuracy of the individual learners in the regression ensemble, as the estimations from individual learners need to be similar to the ground truth in order to be accurate.

Then as an extension of the theorems for classification ensembles derived in [10] [11], we prove the following theorems (Theorem 1 & 2) to illustrate the mathematical relationship between individual accuracy and diversity for regression ensembles.

Theorem 1. *For a regression ensemble with N learners we have that*

$$-\frac{1}{N-1} \leqslant r_{LL}^{(ave)} \leqslant 1, \quad (5)$$

where $r_{LL}^{(ave)}$ is the averaged learner-learner correlations as defined in (3)*. (Proof in the Appendix.)*

Theorem 2. *For a regression ensemble with N learners we have that*

$$-\sqrt{\frac{(N-1) \cdot r_{LL}^{(ave)} + 1}{N}} \leqslant r_{TL}^{(ave)} \leqslant \sqrt{\frac{(N-1) \cdot r_{LL}^{(ave)} + 1}{N}} \quad (6)$$

where $r_{LL}^{(ave)}$ is the averaged learner-learner correlations as defined in (3)*, and $r_{TL}^{(ave)}$ is the averaged truth-learner correlations as defined in* (4)*. (Proof in the Appendix.)*

By combining the above two theorems, we can visualize the relationship between $r_{LL}^{(ave)}$ and $r_{TL}^{(ave)}$ in regression ensembles.

Fig. 1 shows an example of this relationship for ensembles of size 3, and regression ensembles of other different sizes share a similar pattern. In particular, any given regression ensemble with its computed pair $(r_{LL}^{(ave)}, r_{TL}^{(ave)})$ will fall within the parabola-shaped closed region formed by the red curve (the upper bound provided by Theorem 1 & 2), the blue curve (the lower bound provided by Theorem 1 & 2) and the vertical black line $x = 1$ (the genuine upper bound for Pearson correlation coefficient).

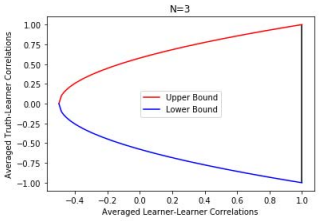

Fig. 1. A theoretical plot (averaged truth-learner correlations versus averaged learner-learner correlations) for regression ensembles of size 3. Ensembles of different sizes have similar patterns. Any regression ensemble will fall within the closed region bounded by the red and blue curve.

The pattern from Fig. 1 suggests that there is always a subtle trade-off between individual accuracies and diversity for regression ensembles. In fact, having both superb diverse and superb accurate individual estimators is impossible based on Theorem 1 & 2, but one can try to balance the two terms to optimize the performance of the ensemble estimator. Such an idea has inspired us to develop a new training algorithm for regression neural network ensembles, which will be detailed in Section III.

III. Loss Function and Training Algorithm for Regression Neural Network Ensembles

A. Loss Function

One advantage of neural network ensembles over other traditional ensembles (e.g. random forests) is that we can explicitly control the trade-off between base learner accuracy and diversity in the loss function we use to train the networks through the process of backpropagation [16]. Herein, inspired by the recent paper [10], we propose a novel loss function that can be used to train regression neural network ensembles, which is a linear combination of the averaged truth-learner correlations $r_{TL}^{(ave)}$ and averaged learner-learner correlations $r_{LL}^{(ave)}$ as shown below

$$Loss = -r_{TL}^{(ave)} + \lambda \cdot r_{LL}^{(ave)}, \tag{7}$$

where $r_{TL}^{(ave)}$ and $r_{LL}^{(ave)}$ is taken as a measure for the overall individual accuracy and diversity of regression neural network ensembles, respectively, and λ is a cost parameter for introducing diversity into an ensemble, which typically takes a value between 0 and 1.

One advantage of our novel loss function (7) proposed for regression neural network ensembles is that we take both individual accuracy and diversity in ensembles into consideration from the perspective of statistical correlations, while standard loss functions, like Cross Entropy [17], only aims to minimize the errors made by the individual learners. More importantly, the parameter λ can control the desired diversity level in the ensemble. In particular, the larger the value of λ, the more diverse the expected ensemble, as we emphasize more on the diversity side with larger λ's.

B. Algorithm

Building upon the novel loss function (7), we now propose the following training algorithm for regression neural network ensembles

Algorithm. Training for regression neural network ensembles.

Input: $X \in \mathcal{R}^{n \times q}$, $Y \in \mathcal{R}^{n \times 1}$, where X is the feature matrix, Y is the vector of true values, n is the number of instances, and q is the number of features.

for epoch in range(num(epochs)):
    optimizer.zero_grad( )
    O=[[ ] for j in range (ensemble_size)]
    for j in range(ensemble_size):
        $O[j]$=nets$[j](X)$
    $\hat{r}_{TL} = 0, \hat{r}_{LL} = 0$
    for j in range(ensemble_size):
        $\hat{r}_{TL} += Corr(Y, O[j])$
        for i in range(ensemble_size):
            if $i < j$:
                $\hat{r}_{LL} += Corr(O[i], O[j])$
    $\hat{r}_{TL}^{(ave)}$ = $\hat{r}_{TL}$/ensemble_size
    $\hat{r}_{LL}^{(ave)}$ = $\hat{r}_{LL}$/(ensemble_size*(ensemble_size -1)/2)
    loss = $-\hat{r}_{TL}^{(ave)} + \lambda \cdot \hat{r}_{LL}^{(ave)}$
    loss.backward( )
    optimizer.step( )

In particular, the major steps of the above training algorithm are as follows

Step 1. Obtain all the corresponding outputs O produced from each of the regression neural networks in the ensemble of size N, where each column of O represents the output produced by the corresponding neural network.

Step 2. Compute all the pairwise Pearson correlations between each column of the network outputs O and the ground truth Y. Take the average of these pairwise correlations as a measure of the averaged truth-learner correlations $\hat{r}_{TL}^{(ave)}$ in the regression neural network ensemble.

Step 3. Similarly compute all the pairwise Pearson correlations between the network outputs O. Take the average of these pairwise correlations as a measure of the averaged learner-learner correlations $\hat{r}_{LL}^{(ave)}$ in the regression neural network ensemble.

Step 4. The regression neural network ensemble now can be trained with the loss function

$$\widehat{Loss} = -\hat{r}_{TL}^{(ave)} + \lambda \cdot \hat{r}_{LL}^{(ave)}, \tag{8}$$

where $\hat{r}_{TL}^{(ave)}$ (calculated in Step 2) is a measure for the overall individual accuracy, and $\hat{r}_{LL}^{(ave)}$ (calculated in Step 3) is a measure for the diversity.

We will demonstrate the effectiveness and advantages of our proposed training algorithm compared to standard regression ensemble methods (e.g. gradient boosting and random forest regressor), by applying our algorithm to both benchmark and real-world regression problems (which will be detailed in Section IV and Section V, respectively).

IV. Predictions on Parkinson's Telemonitoring Data

Herein we will present our regression analysis on a benchmark UCI data set, Parkinson's Telemonitoring data used in

paper [18], and the goal is to predict the clinician's Parkinson's disease symptom score on the UPDRS scale (i.e. "motor_UPDRS" and "total_UPDRS") from other 16 biomedical voice measures.

1) Train-Validation-Test Split: The entire data set is first randomly split into a train and a test (with proportion 5 to 1) set. Then the train set is randomly split again into 5 equally sized folds to perform 5-fold cross validation. Regression ensemble models are trained and validated first, and then applied to the test set.

2) Regression Ensemble Models: Herein we apply four regression ensemble models for comparison purposes, and the details of the models are tabulated in Table I, where gradient boosting regressor and random forest regressor are standard regression ensemble models, while the theoretical neural network ensemble and practical neural network ensemble are ensemble models trained based upon our earlier proposed algorithm. Note that we used L2 loss criterion when training all the base regressors in the four ensemble models, where the L2 loss criterion is defined as

$$Loss_{L_2} = \sum_i (y_i - \hat{y}_i)^2. \tag{9}$$

TABLE I
REGRESSION ENSEMBLE MODELS APPLIED ON PARKINSON'S TELEMONITORING DATA.

Model	Description
Gradient Boosting	Gradient boosting regressor with 100 estimators (i.e. boosting stages) to perform, and L2 loss criterion to measure the quality of a split.
Random Forest	Random forest regressor with 100 estimators (i.e. decision trees) in the forest, and L2 loss criterion to measure the quality of a split.
Our Theoretical Neural Network Ensemble	An ensemble of five fully connected neural networks (pre-trained using L2 loss criterion with 9 hidden layers) trained using our proposed loss function $Loss = -r_{TL}^{(ave)} + \lambda \cdot r_{LL}^{(ave)}$.
Our Practical Neural Network Ensemble	An ensemble of five fully connected neural networks (pre-trained using L2 loss criterion with 9 hidden layers) trained using the loss function $Loss = Loss_{L_2} + \lambda \cdot r_{LL}^{(ave)}$.

3) Evaluation Metrics: We take two error measures as the metrics for evaluating the performance (at both cross-validation and testing stage) of the four regression ensemble models: root mean squared error (RMSE) and mean absolute error (MAE), where RMSE is defined as

$$RMSE = \sqrt{\frac{1}{n}\sum_i^n (y_i - \hat{y}_i)^2}, \tag{10}$$

and MAE is defined as

$$MAE = \frac{1}{n}\sum_i^n |y_i - \hat{y}_i|. \tag{11}$$

4) Results for Predicting "total_UPDRS": The testing errors (RMSE and MAE) of predicting "total_UPDRS" using the four regression models are tabulated in Table II. Note that at both cross-validation and testing stage, our theoretical neural network ensemble and practical neural network ensemble (both trained following the idea of our proposed algorithm) achieve the lower RMSE and MAE compared to standard gradient boosting and random forest regressor. In particular, for RMSE, our practical neural network ensemble gives a testing error of 8.260, which is 1% lower than the error of random forest (8.337) and 10% lower than the error of gradient boosting (9.136). While for MAE, our theoretical neural network ensemble gives a testing error of 6.298, which is 2% lower than the error of random forest (6.394) and 15% lower than the error of gradient boosting (7.381).

5) Results for Predicting "motor_UPDRS": The testing errors (RMSE and MAE) of predicting "motor_UPDRS" using the four regression models are tabulated in Table III. Note that our theoretical neural network ensemble and practical neural network ensemble achieve the lower RMSE and MAE compared to standard gradient boosting and random forest regressor, at both the cross-validation and the testing stage. In particular, for RMSE, our theoretical neural network ensemble gives a testing error of 6.399, which is 1% lower than the error of random forest (6.475) and 9% lower than the error of gradient boosting (7.012). While for MAE, our practical neural network ensemble gives a testing error of 4.850, which is 5% lower than the error of random forest (5.092) and 16% lower than the error of gradient boosting (5.799).

6) Remarks: Based upon the results obtained for predicting "total_UPDRS" and "motor_UPDRS" displayed in Table II and III, we can conclude that our theoretical neural network ensemble and practical neural network ensemble (both trained following the idea of our proposed algorithm) provide better performance (at both the cross-validation and the testing stage) than standard random forest and gradient boosting regressor when analyzing the Parkinson's Telemonitoring data.

V. REGRESSION ANALYSIS ON NOISY RAMAN SPECTROSCOPY DATA

To further demonstrate the effectiveness of our proposed algorithm on regression neural network ensembles, we will present our analysis on Raman spectra [19] simulated for 72 molecules. Raman spectroscopy is useful for many chemical sensor applications. The challenge is to connect spectral features, which are present as vibrational bands in the spectra, to molecular structure. Herein our goal is to predict important chemical features (for example, the numbers of double bonds (DB), double bond equivalents (DBE) and hydrogen atoms, etc.) of these molecules merely based on information gathered from their Raman spectra. Moreover, different levels of noise, such as Gaussian noise and other feature broadening, are added to the original spectroscopy data to evaluate the accuracy of a variety of regression ensemble models (including our neural network ensembles) in the presence of different levels of noise.

A. Feature Engineering

The Raman spectroscopy data of each molecule we have consists of two variables: frequency and peak. Fig. 2 shows two examples of Raman spectra for molecules "Anthracene" and "Tetradecahydroanthracene". In particular, for each of the two molecules, the peak variable is plotted against the frequency variable to visualize the Raman spectrum.

As the frequencies of the vibrational bands of the 72 Raman spectra are not consistent from one spectrum to the next (as shown by the two examples in Fig. 2), to configure the data in a feature matrix amenable to machine learning models, we need to conduct a process of feature engineering in the following steps:

Step 1. Obtain the global minimal frequency and global

TABLE II
RMSE (& MAE) FOR PREDICTING TOTAL_UPDRS OF PARKINSON'S TELEMONITORING DATA.

	Gradient Boosting		Random Forest		Our Theoretical Neural Network Ensemble		Our Practical Neural Network Ensemble	
1_{st} Fold Validation	9.248	(7.474)	8.681	(6.698)	8.684	(6.644)	8.644	(6.694)
2_{nd} Fold Validation	9.064	(7.292)	8.462	(6.556)	7.565	(5.849)	7.751	(6.023)
3_{rd} Fold Validation	9.075	(7.308)	8.324	(6.402)	7.886	(6.025)	7.983	(6.174)
4_{th} Fold Validation	9.157	(7.307)	8.444	(6.560)	7.862	(6.081)	7.975	(6.168)
5_{th} Fold Validation	9.220	(7.335)	8.540	(6.524)	7.889	(5.969)	7.958	(6.035)
Validation Average	9.153	(7.343)	8.490	(6.548)	**7.977**	(**6.114**)	8.062	(6.219)
Testing	9.136	(7.381)	8.337	(6.394)	8.279	(**6.298**)	**8.260**	(6.308)

TABLE III
RMSE (& MAE) FOR PREDICTING MOTOR_UPDRS OF PARKINSON'S TELEMONITORING DATA.

	Gradient Boosting		Random Forest		Our Theoretical Neural Network Ensemble		Our Practical Neural Network Ensemble	
1_{st} Fold Validation	7.101	(5.868)	6.638	(5.305)	5.926	(4.592)	5.978	(4.691)
2_{nd} Fold Validation	6.911	(5.731)	6.396	(5.073)	6.488	(5.142)	6.397	(5.083)
3_{rd} Fold Validation	7.066	(5.806)	6.426	(5.061)	5.891	(4.547)	5.947	(4.651)
4_{th} Fold Validation	6.945	(5.675)	6.556	(5.208)	5.919	(4.670)	5.968	(4.727)
5_{th} Fold Validation	6.995	(5.698)	6.492	(5.112)	5.849	(4.634)	5.924	(4.701)
Validation Average	7.004	(5.756)	6.502	(5.152)	**6.015**	(**4.717**)	6.043	(4.771)
Testing	7.012	(5.799)	6.475	(5.092)	**6.399**	(4.949)	6.400	(**4.850**)

maximal frequency for all applicable spectra, and divide the global frequency interval into a set of sub-intervals with equal width w (e.g. $w = 5$).

Step 2. For each spectrum, compute the averaged peak values within each of the sub-intervals obtained in Step 1, so that a new spectrum is produced.

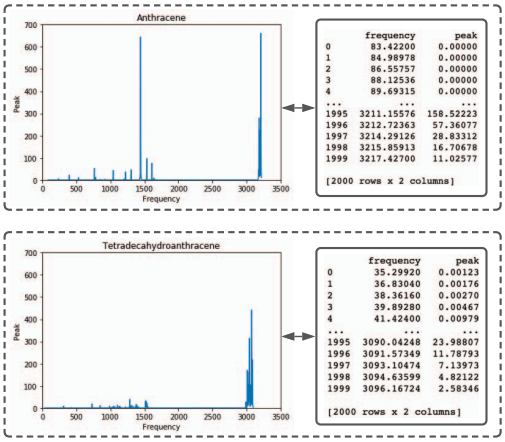

Fig. 2. Examples of Raman spectroscopy data. The upper and lower diagram shows the Raman spectra of molecules "Anthracene" and "Tetradecahydroanthracene", respectively.

Step 3. Add up the peak values of all the new spectra obtained in Step 2, and detect the frequency locations where peaks/spikes occur on the summed spectrum.

Step 4. Obtain new frequency sub-intervals based upon where peaks/spikes occur as suggested by Step 3. Note that these new sub-intervals would be of different widths to capture the variation of the spectra at different frequencies.

Step 5. For each spectrum, compute the averaged peak values within each of the new frequency sub-intervals obtained in Step 4, and take those as the features to be used for machine learning modeling.

Performing the feature engineering process requires care in that the final frequency sub-intervals should be obtained based on the training molecules only, rather than on all (training and testing) molecules, otherwise the information in the testing molecules would leak into the training stage, which would cause "data snooping".

B. Adding Noise to Original Raman Spectroscopy Data

The simulated Raman spectra are "noise-free", that is they contain only information pertaining to the Raman-permitted vibrations in the molecule under consideration. In practice, Raman spectra contain noise, arising from several well known sources. Noise complicates interpretation of Raman spectra, since attributing features to noise or to the underlying molecular structure can be ambiguous. Obtaining accurate chemical information using rapid measurements or low-cost equipment (both of which increase noise compared with what would be possible in a time-average analysis using an expensive instrument) is an important challenge for many industrial applications.

To compare the robustness of a variety of regression ensemble models (including our neural network ensembles) to noisy data, herein we add four different levels of random Gaussian noise ξ to the original spectroscopy data, where $\xi \sim Gaussian(\mu = 0,\ \sigma \in [25, 50, 75, 100])$. Fig. 3 shows an example for the molecule "1,2,8,8a-tetrahydronaphthalene", where the starting subplot is the original Raman spectrum, and then the four levels of noise ($\sigma \in [25, 50, 75, 100]$) are added to the original one, respectively. We can see that the spectra become more and more noisy as we increase the value of σ, and it is almost formed by pure noise when $\sigma = 100$.

Note that the feature engineering steps (introduced in Section V-A) for noisy spectra are performed after Gaussian noise is added to the original spectra.

C. Regression Analysis

1) Dependent Variables: The five feature matrices obtained after feature engineering corresponding to the original and noisy Raman spectra ($\sigma \in [25, 50, 75, 100]$) all have 7200 rows each. In particular, random Gaussian noise (say, $\sigma = 25$) is added repeatedly for 100 times to each of the 72 molecules,

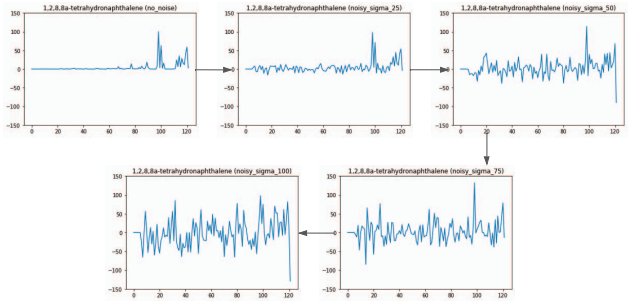

Fig. 3. Examples of Raman spectroscopy data with different levels of Gaussian noise added. The starting subplot is the original spectrum, and then four different levels of noise ($\sigma \in [25, 50, 75, 100]$) are added to the original one, respectively.

forming the noisy feature matrix with 7200 rows. While each of the 72 original spectra is replicated 100 times to form the original feature matrix with 7200 rows. These five feature matrices will be used as the dependent data to get inference for the independent variables.

2) Independent Variables: There are three independent/target variables of interest here: the number of double bonds (DB), double bond equivalents (DBE) and hydrogen atoms, which are all important chemical characteristics for molecules.

3) Train-Validation-Test Split: For both the original data and noisy data, the entire data set is first randomly split into a train and a test (5:1) set. Then the train set is randomly split again into a sub-train and a validation (4:1) set (note that such random split is performed 5 times to ensure robust results). Regression ensemble models are trained on the sub-train set (the training stage), validated on the validation set (the cross-validation stage), and then applied to the test set (the generalization stage).

4) Regression Ensemble Models: Herein we apply four regression ensemble models for comparison purposes, and the details of the models are tabulated in Table IV, where gradient boosting regressor and random forest regressor are standard regression ensemble models, while the theoretical neural network ensemble and practical neural network ensemble are ensemble models trained based upon our earlier proposed algorithm. Note that we used both L1 and L2 loss criterion for comparison purposes when training the base regressors in the four ensemble models, where the L1 loss criterion is defined as

$$Loss_{L_1} = \sum_i |y_i - \hat{y}_i|,$$

and the L2 loss criterion is defined in (9).

5) Evaluation Metrics: Herein we take two error measures as the metrics for evaluating the performance (at both the cross-validation and the testing stage) of the four regression ensemble models: root mean squared error (RMSE), as defined in (10), and mean absolute error (MAE), as defined in (11).

6) Results for the Number of Double Bonds (DB): Fig. 4 visualizes the trend of errors made by the four ensemble models (introduced in Table IV) for predicting DB over increasing noise levels, where the upper two subplots show RMSE and MAE when base learners are trained using L2 loss, and the lower two subplots show RMSE and MAE when base learners are trained using L1 loss. For all four subplots, the dashed curves correspond to the averaged validation errors

TABLE IV
REGRESSION ENSEMBLE MODELS APPLIED ON ORIGINAL AND NOISY RAMAN SPECTROSCOPY DATA.

Model	Description
Gradient Boosting	Gradient boosting regressor with 100 estimators (i.e. boosting stages) to perform, and L1 (or L2) loss criterion to measure the quality of a split.
Random Forest	Random forest regressor with 100 estimators (i.e. decision trees) in the forest, and L1 (or L2) loss criterion to measure the quality of a split.
Our Theoretical Neural Network Ensemble	An ensemble of three fully connected neural networks (pre-trained using L1 (or L2) loss criterion with 5 hidden layers) trained using our proposed loss function $Loss = -r_{TL}^{(ave)} + \lambda \cdot r_{LL}^{(ave)}$.
Our Practical Neural Network Ensemble	An ensemble of three fully connected neural networks (pre-trained using L1 (or L2) loss criterion with 5 hidden layers) trained using the loss function $Loss = Loss_{L_1}(or\ Loss_{L_2}) + \lambda \cdot r_{LL}^{(ave)}$.

obtained from 5 random runs of cross-validation, while the solid curves correspond to the testing errors.

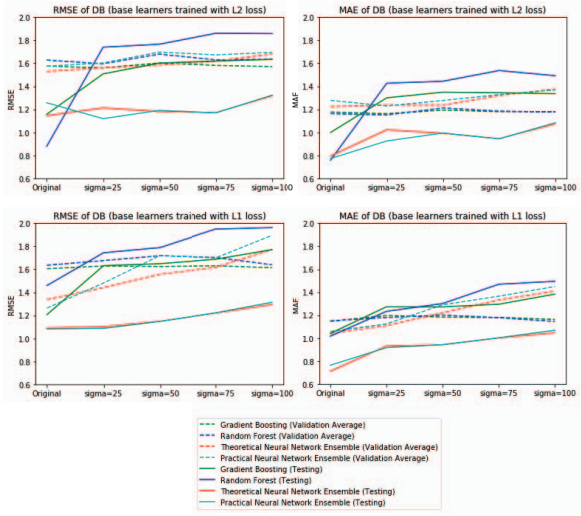

Fig. 4. A visualization for the errors (RMSE and MAE) made by the four ensemble models introduced in Table IV for predicting **DB** over increasing noise levels (original and $\sigma \in [25, 50, 75, 100]$) in the data, when the base learners in all ensembles are trained with L2 and L1 loss criterion, respectively. The dashed curves correspond to averaged validation errors over 5 random runs of cross-validation, while the solid curves correspond to testing errors. The green, blue, red and cyan curves represent results for gradient boosting, random forest, our theoretical neural network ensemble and our practical neural network ensemble, respectively.

A general trend we can see from the four subplots is that both RMSE and MAE increase for all ensemble models as the data become more and more noisy, which is reasonable because the true pattern in the data is more difficult to be detected under a higher level of noise. However, compared to gradient boosting and random forest, our theoretical and practical neural network ensembles trained following the idea of our algorithm show a rather robust pattern when handling noisy data, especially for the testing errors in the upper two subplots. On closer examination of each subplot in Fig. 4, our theoretical and practical neural network ensemble turns out

to generalize better to the testing data than gradient boosting and random forest by showing a much lower testing RMSE and MAE whenever a L1 or L2 loss criterion is applied. In particular, when $\sigma = 75$ (the second highest level of noise considered here), refer to the lower left subplot, the testing RMSE of our theoretical and practical neural network ensemble are both around 1.17, which cuts the error of random forest (which is around 1.86) by 37%.

7) Results for the Number of Double Bond Equivalents (DBE): Fig. 5 visualizes the trend of errors made by the four ensemble models for predicting DBE over increasing noise levels. As we can see from the four subplots, our theoretical and practical neural network ensemble turns out to generalize better to the testing data than gradient boosting and random forest by showing a much lower testing RMSE and MAE whenever a L1 or L2 loss criterion is applied. In particular, when $\sigma = 75$ (the second highest level of noise considered here), refer to the lower left subplot, the testing RMSE of our theoretical and practical neural network ensemble are both around 1.83, which cuts the error of random forest (which is around 3.13) by 42%.

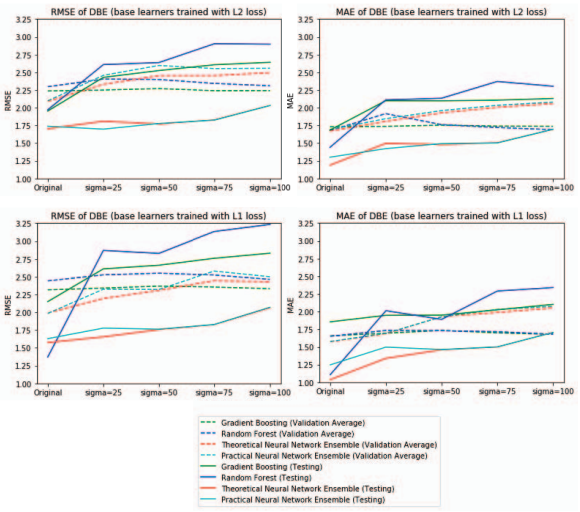

Fig. 5. A visualization for the errors (RMSE and MAE) made by the four ensemble models introduced in Table IV for predicting **DBE** over increasing noise levels (original and $\sigma \in [25, 50, 75, 100]$) in the data, when the base learners in all ensembles are trained with L2 and L1 loss criterion, respectively. The dashed curves correspond to averaged validation errors over 5 random runs of cross-validation, while the solid curves correspond to testing errors. The colors of the curves are the same as those in Fig. 4.

8) Results for the Number of Hydrogen Atoms: Fig. 6 visualizes the trend of errors made by the four ensemble models for predicting hydrogen atoms over increasing noise levels. As we can see from the four subplots, our theoretical and practical neural network ensemble turns out to generalize better to the testing data than gradient boosting and random forest by showing a much lower testing RMSE and MAE whenever a L1 or L2 loss criterion is applied. In particular, when $\sigma = 50$, refer to the lower left subplot, the testing RMSE of our theoretical and practical neural network ensemble are both around 2.13, which cuts the error of gradient boosting (which is around 2.94) by 28%.

9) Remarks: Based upon the regression analysis on the original and noisy Raman spectroscopy data for predicting the number of DB, DBE and hydrogen atoms, we can conclude that compared to the standard regression ensemble models gradient boosting and random forest, our theoretical and practical neural network ensemble trained following the idea of our proposed algorithm are rather robust to noise, and can also generalize better even when noisy data is presented.

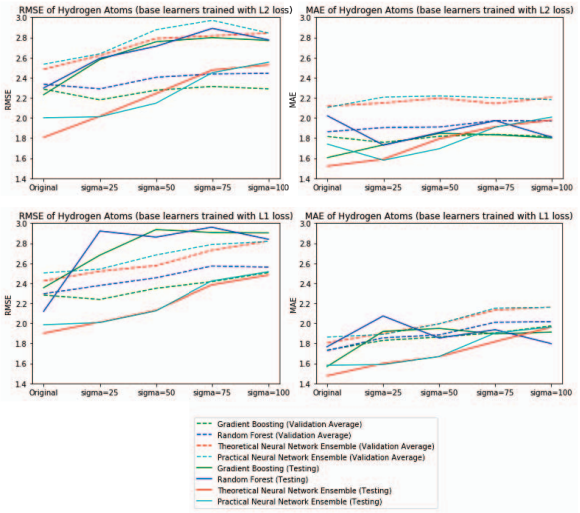

Fig. 6. A visualization for the errors (RMSE and MAE) made by the four ensemble models introduced in Table IV for predicting **hydrogen atoms** over increasing noise levels (original and $\sigma \in [25, 50, 75, 100]$) in the data, when the base learners in all ensembles are trained with L2 and L1 loss criterion, respectively. The dashed curves correspond to averaged validation errors over 5 random runs of cross-validation, while the solid curves correspond to testing errors. The colors of the curves are the same as those in Fig. 4 and Fig. 5.

VI. Conclusions

In this paper, we have derived two theorems that offer a rigorous understanding of the trade-off between individual accuracy of component learners and ensemble diversity in regression ensembles. Inspired by our derived theorems, we then proposed a new training algorithm for deep regression neural network ensembles that can explicitly encourage ensemble diversity. This algorithm is demonstrated to be generally effective, compared to benchmark ensemble models such as random forest and gradient boosting, by analysis on both standard machine learning data sets and real-world chemical applications including Raman spectroscopy analysis.

VII. Acknowledgements

This research was performed using computational resources supported by the Academic & Research Computing group at Worcester Polytechnic Institute (WPI). We would also like to thank the WPI Transformative Research and Innovation, Accelerating Discovery (TRIAD) grant program and National Science Foundation (ENG/1554283) for supporting this work.

Appendix

Proof of theorem 1. — First, we will prove the lower bound. Consider the outputs of the N learners in the regression ensemble, standardize the outputs so that each output can be considered as a random variable L_i with unit variance, i.e. $Var(L_i) = 1, i = 1, 2, \cdots, N$. Based on one statistical property of variance, we have

$$Var\Big(\sum_{i=1}^{N} L_i\Big) = \sum_{i=1}^{N} Var(L_i) + \sum_{i=1}^{N}\sum_{j\neq i}^{N} Cov(L_i, L_j) \qquad (12)$$

Since $Var(L_i) = 1,\ i = 1, 2, \cdots, N$, the covariance between two random variables equals the correlation, i.e. $Cov(L_i, L_j) = Corr(L_i, L_j) = r_{L_i,L_j}$. We also have $r_{L_i,L_j} = r_{L_j,L_i}$, therefore we can rewrite (12) as

$$\begin{aligned} Var\Big(\sum_{i=1}^{N} L_i\Big) &= N + \sum_{i=1}^{N}\sum_{j\neq i}^{N} r_{L_i,L_j} \\ &= N + 2\sum_{i=1}^{N}\sum_{j>i}^{N} r_{L_i,L_j} \\ &= N + 2\cdot\frac{N(N-1)}{2} r_{LL}^{(ave)} \end{aligned} \tag{13}$$

Since $Var\Big(\sum_{i=1}^{N} L_i\Big) \geqslant 0$, we can get from (13) that

$$r_{LL}^{(ave)} \geqslant -\frac{1}{N-1}. \tag{14}$$

As for the upper bound, since $-1 \leqslant r_{L_i,L_j} \leqslant 1$, we have

$$\begin{aligned} r_{LL}^{(ave)} &= \frac{1}{N(N-1)/2}\sum_{i=1}^{N}\sum_{j>i}^{N} r_{L_i,L_j} \\ &\leqslant \frac{1}{N(N-1)/2}\sum_{i=1}^{N}\sum_{j>i}^{N} 1 \\ &= \frac{1}{N(N-1)/2}\cdot\frac{N(N-1)}{2} = 1. \end{aligned} \tag{15}$$

In conclusion, for regression ensembles of size N, the averaged learner-learner correlations $r_{LL}^{(ave)}$ satisfies $-\frac{1}{N-1} \leqslant r_{LL}^{(ave)} \leqslant 1$. ∎

Proof of theorem 2. — Consider the prediction outputs of the N learners in a regression ensemble, standardize the outputs so that each output can be considered as a random variable L_i with unit variance, i.e. $Var(L_i) = 1, i = 1, 2, \cdots, N$. Similarly standardize the values of the ground truth T such that $Var(T) = 1$.

Define the sum of the outputs of the N learners as: $S = \sum_{i=1}^{N} L_i$, then we have

$$\begin{aligned} Cov(S,T) &= Cov\Big(\sum_{i=1}^{N} L_i, T\Big) = \sum_{i=1}^{N} Cov(L_i, T) \\ &= \sum_{i=1}^{N} Corr(L_i, T) = N\cdot r_{TL}^{(ave)}, \end{aligned} \tag{16}$$

where $Cov(.)$ is the covariance, $Corr(.)$ is the correlation. Note that $Cov(L_i, T) = Corr(L_i, T)$ in (16) because L_i's and T are all unit vectors.

Based on one statistical property of variance, we have

$$Var\Big(\sum_{i=1}^{N} L_i\Big) = \sum_{i=1}^{N} Var(L_i) + \sum_{i=1}^{N}\sum_{j\neq i}^{N} Cov(L_i, L_j). \tag{17}$$

Since $Var(L_i) = 1,\ i = 1, 2, \cdots, N$, the covariance between two random variables equals the correlation, i.e. $Cov(L_i, L_j) = Corr(L_i, L_j) = r_{L_i,L_j}$. We also have $r_{L_i,L_j} = r_{L_j,L_i}$, therefore we can rewrite (17) as

$$\begin{aligned} Var\Big(\sum_{i=1}^{N} L_i\Big) &= N + \sum_{i=1}^{N}\sum_{j\neq i}^{N} r_{L_i,L_j} \\ &= N + 2\sum_{i=1}^{N}\sum_{j>i}^{N} r_{L_i,L_j} \\ &= N + 2\cdot\frac{N(N-1)}{2} r_{LL}^{(ave)} \end{aligned} \tag{18}$$

Finally based on the Cauchy-Schwarz inequality, we get:

$$\begin{aligned} (Cov(S,T))^2 &\leqslant Var(S)\cdot Var(T) \\ &= Var(S)\cdot 1 \\ &= Var\Big(\sum_{i=1}^{N} L_i\Big) \\ &= N + N(N-1) r_{LL}^{(ave)} \end{aligned} \tag{19}$$

Combining (16) and (19),

$$(N\cdot r_{TL}^{(ave)})^2 \leqslant N + N(N-1)\cdot r_{LL}^{(ave)}. \tag{20}$$

Therefore, after simplification, we get

$$-\sqrt{\frac{(N-1)\cdot r_{LL}^{(ave)} + 1}{N}} \leqslant r_{TL}^{(ave)} \leqslant \sqrt{\frac{(N-1)\cdot r_{LL}^{(ave)} + 1}{N}} \tag{21}$$

∎

REFERENCES

[1] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*. Springer, 2000, pp. 1–15.

[2] T. P. Williams and J. Gong, "Predicting construction cost overruns using text mining, numerical data and ensemble classifiers," *Automation in Construction*, vol. 43, pp. 23–29, 2014.

[3] X. Amatriain, A. Jaimes, N. Oliver, and J. M. Pujol, "Data mining methods for recommender systems," in *Recommender systems handbook*. Springer, 2011, pp. 39–71.

[4] J. Mendes-Moreira, C. Soares, A. M. Jorge, and J. F. D. Sousa, "Ensemble approaches for regression: A survey," *Acm computing surveys (csur)*, vol. 45, no. 1, pp. 1–40, 2012.

[5] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[6] Y. Freund, R. E. Schapire *et al.*, "Experiments with a new boosting algorithm," in *icml*, vol. 96. Citeseer, 1996, pp. 148–156.

[7] J. H. Friedman, "Stochastic gradient boosting," *Computational statistics & data analysis*, vol. 38, no. 4, pp. 367–378, 2002.

[8] D. H. Wolpert, "Stacked generalization," *Neural networks*, vol. 5, no. 2, pp. 241–259, 1992.

[9] A. Krogh and J. Vedelsby, "Validation, and active learning," *Advances in neural information processing systems 7*, vol. 7, p. 231, 1995.

[10] W. Li and R. Paffenroth, "Optimal ensembles for deep learning classification: Theory and practice," in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. IEEE, 2019, pp. 1658–1665.

[11] W. Li, R. C. Paffenroth, and D. Berthiaume, "Neural Network Ensembles: Theory, Training, and the Importance of Explicit Diversity," *arXiv e-prints*, Sep. 2021, arXiv:2109.14117 [cs.LG].

[12] L. I. Kuncheva, *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, 2004.

[13] G. Brown, J. Wyatt, R. Harris, and X. Yao, "Diversity creation methods: a survey and categorisation," *Information fusion*, vol. 6, no. 1, pp. 5–20, 2005.

[14] J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," in *Noise reduction in speech processing*. Springer, 2009, pp. 1–4.

[15] Y. Liu and X. Yao, "Ensemble learning via negative correlation," *Neural networks*, vol. 12, no. 10, pp. 1399–1404, 1999.

[16] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural networks for perception*. Elsevier, 1992, pp. 65–93.

[17] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes, "Ensemble selection from libraries of models," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 18.

[18] A. Tsanas, M. Little, P. McSharry, and L. Ramig, "Accurate telemonitoring of parkinson's disease progression by non-invasive speech tests," *Nature Precedings*, pp. 1–1, 2009.

[19] D. A. Long, "Raman spectroscopy," *New York*, pp. 1–12, 1977.