Ensembles of Networks Produced from Neural Architecture Search*

Emily J. Herron[1,2] [0000-0002-7300-8172],
Steven R. Young[2,1] [0000-0003-0591-4330], and
Thomas E. Potok[2] [0000-0001-6687-3435]

[1] Bredesen Center, University of Tennesse-Knoxville, Knoxville, TN
[2] Computational Data Analytics, Oak Ridge National Laboratory, Oak Ridge, TN
herronej@ornl.gov

Abstract. Neural architecture search (NAS) is a popular topic at the intersection of deep learning and high performance computing. NAS focuses on optimizing the architecture of neural networks along with their hyperparameters in order to produce networks with superior performance. Much of the focus has been on how to produce a single best network to solve a machine learning problem, but as NAS methods produce many networks that work very well, this affords the opportunity to ensemble these networks to produce an improved result. Additionally, the diversity of network structures produced by NAS drives a natural bias towards diversity of predictions produced by the individual networks. This results in an improved ensemble over simply creating an ensemble that contains duplicates of the best network architecture retrained to have unique weights.

Keywords: neural architecture search · ensembles · high performance computing

1 Introduction

There has been much work in recent years in developing methods for automatically designing neural networks for various challenges and datasets. This work in neural architecture search (NAS) largely focuses on finding a single best network. However, throughout this process many networks are created and evaluated. This

provides the opportunity to find not just a single network that performs well, but an ensemble of networks that perform well together on problems of interest.

In this work, we will study the results of ensembling networks produced by one such NAS method. The NAS method used is Multi-node Evolutionary Neural Networks for Deep Learning (MENNDL). It produces a variety of deep learning networks that perform well on the given dataset. However, the single network that gives optimal performance may still be limited in its knowledge of the distribution of the data or be over or under-fitted to the training data. Combining the outputs of multiple deep neural network classifiers has been demonstrated as an effective approach that offers significantly better prediction accuracies than that of individual models [11]. Neural network ensembles do so by combining outputs from a finite number of neural networks with different parameters that have been trained on the same data. In this report, we create and evaluate the performances of ensembles of the best performing networks produced by one or more runs of MENNDL. We consider two approaches to creating ensembles from this NAS approach and apply these approaches to two traditional image dataset benchmarks. The key contribution of this work is a study detailing the effects of ensembling networks from an NAS method including:

1. The effect of using ensembles created from multiple instantiations of the NAS method.
2. The effect of the size of the ensemble on performance.
3. The resulting performance measured as accuracy on the problem and the diversity in the ensemble.

2 Background and Related Work

Deep learning is a branch of machine learning based on the concept of learning features from multiple layers of abstraction[14]. In recent years, deep learning models have advanced the state of the art of tasks in fields like image recognition and generation in computer vision; language translation, text classification; and sentiment analysis in natural language processing; and automatic speech identification and generation in speech recognition[3]. Scientific research applications that involve analysis of large volumes of images produced with specialized instruments in particular also rely on the use of these models.

2.1 Neural Architecture Search

The features of deep learning models are controlled by a set of hyperparameters, which in the case of a deep convolutional neural network (CNN), include the number of hidden layers as well as each layer's number of nodes, activation function, and kernel size. The learning capacity of deep neural networks is dependent upon these hyperparameters, which must be selected appropriately to suit a particular dataset. The process of tailoring a deep neural network architecture to a particular data set can be computationally expensive and time consuming even

with the guidance of experts. Furthermore, the features of scientific datasets often differ from that of traditional datasets. Hence, models optimized for traditional datasets may not be well-suited to scientific datasets. Hyperparameters have traditionally been selected either by manual trial and error or grid search. Manual search often requires expert users and involves selecting a set of hyperparameters from a region thought to be best-suited to the data [14]. Grid search, in contrast, finds an optimal solution after evaluating models assembled with each possible combination of hyperparameters. This method is preferred to manual search due to its ease of implementation and tendency to provide a better solution; however, it fails to be efficient in high dimensional feature spaces. If the selection is carried out this way, it can be a time-intensive task owning both to the expansive range of hyperparameters and the evaluation time of each possible network [13, 14]. To overcome the drawbacks of these methods, researchers have suggested other approaches, including evolutionary algorithms.

We use an evolutionary optimization approach to NAS in this work known as MENNDL [13]. MENNDL is a GPU-based high performance computing framework that uses an asynchronous steady-state evolutionary algorithm to parallelize the large-scale evaluation of networks on individual nodes, with selection, mutation, and crossover procedures controlled by a master node. This allows for a more efficient search of a high dimensional hyperparameter space than grid search, and improves upon random search by considering previous results [13, 14]. Networks produced by evolution-based optimization frameworks like MENNDL have demonstrated increased accuracy and efficiency compared to those suggested by domain experts [12].

2.2 Neural Network Ensembles

Neural network ensembles have been defined as a collection of neural networks that have been trained on the same task before their results are combined to produce a model with better generalization ability than individual networks. They have been applied to a variety of problems including handwritten digit recognition, scientific image analysis, face recognition, and OCR [15]. The idea behind the use of neural network ensembles is that the success of a deep learning model is predicated upon its ability to learn the distribution of a dataset. However, a single model that performs optimally on a training dataset may be over-fit to the training set and perform poorly on unseen data. Ensembles of networks with different parameters and architectures can reduce this risk since different networks may learn varying aspects of the training set before being combined to produce the desired outputs. The networks are typically combined by taking an average or weighted average of the outputs of each model in order to obtain the final result [2].

Constructing ensembles of networks can be a challenging task. Traditionally, ensemble techniques have relied on networks with randomly generated topologies, weights, or topologies that have learned random subsets of the training data. The intuition behind this this that the networks will be diverse in the sense that they differ in terms of their errors. [7] It has been shown that the

generalization ability of an ensemble is directly dependent upon its average generalization ability (e.g. accuracy) and average diversity of individual networks in the model. Previous work has found that the accuracy of an ensemble model can be improved by constructing and weighting multiple base learners and that the diversity of a model can be enhanced by selecting only learners that are are less correlated in terms of training error. [2] Other studies have concluded that the ideal ensemble is one comprised of accurate networks that make errors on different parts of the input space.[7] A range of solutions have aimed to address the problem of assembling neural network ensembles that balance fitness and diversity. One work demonstrated that large ensembles of neural network models can be summarized with a relatively small number of representative models selected via clustering based on distances between model outputs. This method was demonstrated to, in certain cases, yield better prediction accuracies. [1] Elsewhere, a cluster-based selective algorithm was proposed for building a neural network ensemble based on the idea that more effective ensembles are comprised of networks that are both accurate and diverse. Clustering was used to identify subsets of similar networks before selecting the most accurate network from each cluster to form an ensemble. Experiments showed that this approach outperformed traditional ensemble approaches such as Boosting and Bagging [9]. In another study, an ensemble-based model was implemented by using a genetic algorithm to calculate the weights of individual networks to create a population with high overall accuracy. K-means clustering was then used to select an optimal subset of learners to improve the diversity of the model. This approach was compared to other ensemble techniques including the traditional average, weighted average, and kriging models and demonstrated to outperform each [2]. A different study examined the relationship between the generalization abilities of neural network ensembles and correlations between networks based on correctly and incorrectly classified samples selected at random. It was discovered that, in some instances, selecting a subset of networks was superior to ensembles of all of the individual networks. The authors proposed an approach that uses a genetic algorithm to select an optimal set of neural networks given a set of pretrained networks to serve as an ensemble. They demonstrated that this method worked well compared to a popular ensemble approach and produced ensembles with high generalizing ability with a relatively low computational cost [15]. Another publication introduced a method known as Addemup, which leveraged a diverse population of neural networks generated by a genetic algorithm in creating an ensemble of neural networks. The genetic algorithm used for this purpose was designed to meet an objective function that seeks to maximize the accuracy of the networks while ensuring dissimilarity between members of the population. Ensembles were evaluated during training following an approach that focuses on more difficult examples in order to quickly produce good results. The authors demonstrated that their algorithm yielded significantly better results than uses of the single best network alone, the Bagging ensemble approach, and a similar algorithm with an objective function that only considers validation accuracy. [7].

3 Methods

3.1 MENNDL

Multi-node Evolutionary Neural Networks for Deep Learning (MENNDL) is a software framework that implements an evolutionary algorithm for optimizing neural network topology and hyperparameters. More specifically, it can optimize the number of layers, layer type for each layer, and the corresponding layer hyperparameters. MENNDL utilizes an asynchronous approach to evaluate the networks it generates in parallel in order to maximize utilization of available computation resources. Evolutionary algorithms mimic the process of natural selection, treating a population of neural networks as individuals, each with their own 'genes' or set of architectural hyperparameters. The fitnesses of individuals in each generation are evaluated before a selection protocol chooses a subset of individuals in the population who will pass their features on to the next generation of networks, following crossover and mutation. Given proper initialization, parameterization, and a sufficient number of generations, this framework produces high-performing networks by focusing on regions of the parameter space containing individuals selected at each generation, while avoiding searches in the neighborhoods of less-fit individuals [3, 10]. Figure 1 illustrates the architectures of the top networks produced by eight separate runs of MENNDL against the CIFAR-10 image dataset. Note that the architectures of the best performing networks produced by each run are diverse, yet each network performs comparably on the validation sets. The specific details of the evolutionary algorithm implemented by MENNDL are provided in [13].

3.2 Ensembles of MENNDL Generated Networks

We created ensembles of the top networks across one or more runs of MENNDL against two different datasets: MNIST and CIFAR-10. For each dataset, MENNDL was run 24 times on 8 nodes for 6 hours each. For each of these runs, a 'keep best' flag was used in order to automatically select the individual with the highest fitness at each generation. The validation accuracies and networks from each run were saved. Following these runs, four categories of ensembles were assembled from the networks with the highest validation accuracies. The first three ensembles were of the top 2, 4, and 8 networks and the fourth was of 8 separately trained versions of the top network. The networks were selected from each run as well as from pools of 2, 4, and 8 randomly selected runs for the same dataset. Each of the chosen networks were evaluated on the test set, producing softmax outputs that were averaged to obtain the final predictions. The ensembles were repeated 24 times for each combination of ensemble type and selection pool size. Each MENNDL run and ensemble experiment was carried out on on the Summit supercomputer at Oak Ridge National Laboratory. The system has a total of 4608 nodes, each with two IBM POWER9 CPUs and six NVIDIA Volta GPUs.[8] The diversity of each ensemble was measured by averaging the total disagreement between the predicted outputs for each sample, following a

method similar to the one described in [6]. Given two arrays of sample predictions for a test set of length m, p_i and p_j, the average disagreement between the two sets of predictions is calculated using the equation

$$d(p_i, p_j) = \frac{1}{m} \sum_{k=1}^{m} \psi(p_{ik}, p_{jk}) \tag{1}$$

Where the disagreement between two predictions for a sample at index k is given by

$$\psi(p_{ik}, p_{jk}) = \begin{cases} 0, & p_{ik} = p_{jk} \\ 1, & \text{otherwise} \end{cases} \tag{2}$$

The average disagreements between the predictions of an ensemble of size n are then averaged:

$$\frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} d(p_i, p_j) \tag{3}$$

The result is a value that measures the probability that two networks in the ensemble will disagree with one another given a sample in the test set.

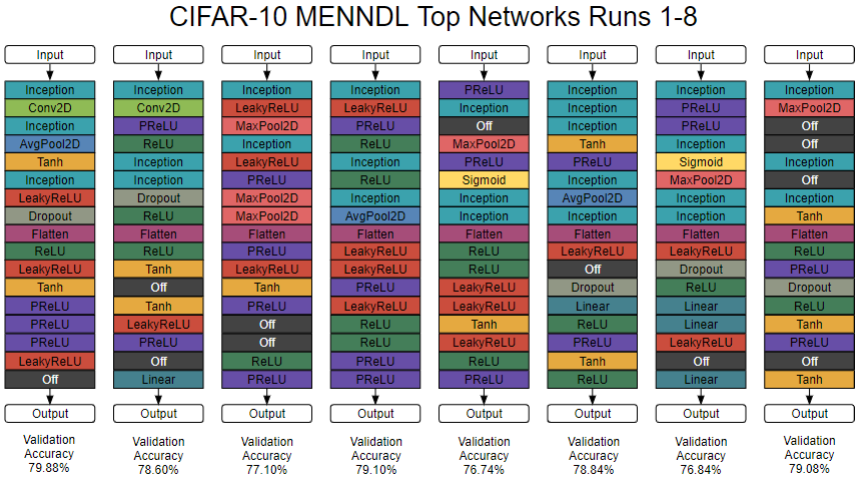

Fig. 1. Top network architectures produced by eight separate runs of MENNDL against the CIFAR-10 training dataset. The validation accuracies are listed for each network.

3.3 Datasets and Experiments

A set of experiments were carried out to compare the accuracies of top individual networks to those of four different ensembles of top performing networks.

Three ensembles were created by selecting the top 2, 4, and 8 networks from a pool of runs. A fourth ensemble was constructed by combining the outputs of 8 separately trained versions of the top network architecture. The top networks for each of the 24 MENNDL runs against the dataset were evaluated against ensembles comprised of networks from 1, 2, 4 or 8 randomly selected MENNDL runs. The top networks and ensembles were selected and evaluated 24 times for per configuration.

The CIFAR-10 [4] and MNIST [5] image datasets were used in these experiments. The CIFAR-10 dataset consists of 60,000 32 by 32 multicolor images, each belonging to one of 10 classes. It is divided into a training set of size 50,000 and test set of size 10,000. The MNIST dataset consists of training and test sets of 60,000 and 10,000 28 by 28 grayscale images of handwritten digits ranging from 0 to 9. Upon initializing each run of MENNDL, the CIFAR-10 samples were normalized with the mean and standard deviation transforms (0.4914 0.4822 0.4465) and (0.2023 0.1994 0.2010) and MNIST with (0.1307) and (0.3081). 10% of the samples in each training set were selected at random and held out as a validation set. Individual networks were trained with a batch size of 64 on the remaining training samples. Afterward, the networks were evaluated on the validation set to obtain the fitnesses for selection. The CIFAR-10 and MNIST test sets were used to obtain the accuracies of each ensemble. No data augmentation or transformation beyond the simple normalization given above was used in this work.

4 Results

The means of the total networks, generations, and maximum fitnesses across 24 runs of MENNDL against each dataset are listed in Table 1. We note that the datasets' mean total networks and generations per run were similiar. However, the average maximum fitness was significantly higher with MNIST than with CIFAR-10. The standard deviation of this statistic was also much lower with the MNIST than CIFAR-10.

The mean test set accuracies for each ensemble and pool size configuration are listed in Tables 2 and 3 and plotted in Figure 2. The ensemble accuracies were generally higher when ensembles were composed of more of the top networks. This trend was consistent in the case of both when going from the top individual network to the ensemble of the top 8 networks. Creating an ensemble of only the top two networks offered accuracy improvements over that of individual networks of as much as ($3.0900 \pm 0.3698\%$) on CIFAR-10 and ($0.1821 \pm 0.0432\%$) on MNIST. This finding is consistent with our expectations and demonstrates that creating ensembles of the top two or more MENNDL runs is an effective means of improving upon the generalizability of the single best-performing network across one or more runs.

The CIFAR-10 ensembles also tended to achieve higher overall test set accuracies when larger pools of runs were used. However, this trend was not the case with the MNIST ensembles. This is likely because the average best individual

network fitnesses of the MENNDL runs against the MNIST dataset had considerably lower standard deviations than that of the runs against the CIFAR-10 dataset. In other words, the top network accuracies from the CIFAR-10 dataset varied more than those from MNIST. Hence, selecting the top overall networks from larger pools of MENNDL runs against this dataset would more likely result in top networks with higher generalization ability than top networks chosen from a smaller pool or single run. Additionally, as the misclassification rate was much smaller for the best MNIST networks, there is little room to add functionally diverse networks to the ensemble while still maintaining high classification rates.

The mean accuracies and diversities of ensembles of the top 8 networks and the top network trained 8 separate times are listed in Tables 4 and 5. These results reveal that ensembles of the top 8 networks yielded diversities that were consistently higher than the ensembles of 8 separately trained versions of the top network. The ensemble diversities' tendency to decrease as larger pools of runs were used was likely an artifact of the larger pools of runs' increased likelihood of having access to top networks with better generalizability, resulting in outputs that were less likely to differ from one another.

Table 1. CIFAR-10 and MNIST mean total networks, generations, and fitness of best network across 24 runs of MENNDL.

Statistic	Dataset	
	CIFAR-10	MNIST
Total Networks	607.63±86.35	589.63±73.71
Generations	13.54±1.76	13.08±1.61
Best Network Fitness	78.47±1.26	99.33±0.10

Table 2. MNIST mean top network and ensemble test set accuracies for run pool sizes of 1, 2, 4, and 8. Note that the ensembles of the top 8 networks from run pool sizes of 2 and 4 achieved the highest mean accuracies out of all configurations.

MENNDL Runs	Ensemble Method				
	Top Network	Top 2 Networks	Top 4 Networks	Top 8 Networks	Top Network 8x
1	99.4067±0.1225	99.2471±0.1225	99.4929±0.0658	99.4929±0.0624	99.4092±0.1226
2	99.2554±0.1129	99.4375±0.0697	99.4742±0.0815	**99.5487±0.0550**	99.4471±0.0897
4	99.2858±0.0953	99.3954±0.0816	99.5029±0.0443	99.5125±0.0673	99.4629±0.0666
8	99.2629±0.1154	99.4117±0.0860	99.4646±0.0587	99.5229±0.0501	99.4038±0.0933

5 Conclusion and Future Work

We have presented a study demonstrating that creating ensembles of multiple different networks from a NAS method produces a better result than simply

Table 3. CIFAR-10 mean top network and ensemble test set accuracies for run pool sizes of 1, 2, 4, and 8. Note that the ensemble of the top 8 networks from a run pool size of 8 achieved the highest mean accuracy out of all configurations.

MENNDL Runs	Ensemble Method				
	Top Network	Top 2 Networks	Top 4 Networks	Top 8 Networks	Top Network 8x
1	77.9025±1.5848	80.9925±1.2150	82.5629±1.1345	83.0067±0.9954	82.7583±1.5473
2	78.3483±1.1599	80.8808±1.6867	83.0500±0.8213	83.5075±0.7859	83.4538±1.2226
4	79.9271±1.5532	81.6767±1.2697	83.5146±0.8869	**83.9796±0.6361**	83.1325±1.0810
8	79.7904±1.3920	81.7825±1.7717	83.6996±0.7334	84.3708±0.6521	84.0350±1.0589

Table 4. MNIST mean diversities and accuracies for ensembles of top 8 and of top network trained 8 separate times selected from run pools of size 1, 2, 4, and 8.

MENNDL Runs	Ensemble Method			
	Top 8 Networks		Top Network 8x	
	Diversity	Accuracy	Diversity	Accuracy
1	0.0077±0.0007	99.4929±0.0624	0.0058±0.0016	99.4092±0.1226
2	0.0071±0.0005	99.5487±0.0550	0.0052±0.0013	99.4471±0.0897
4	0.0068±0.0008	99.5125±0.0673	0.0057±0.0014	99.4629±0.0666
8	0.0065±0.0007	99.5229±0.0496	0.0061±0.0009	99.4038±0.0933

Table 5. CIFAR-10 mean diversities and accuracies for ensembles of top 8 and of top network trained 8 separate times selected from run pools of size 1, 2, 4, and 8.

MENNDL Runs	Ensemble Method			
	Top 8 Networks		Top Network 8x	
	Diversity	Accuracy	Diversity	Accuracy
1	0.2118±0.0199	83.0067±0.9954	0.1798±0.0200	82.7583±1.5473
2	0.1984±0.0146	83.5075±0.7859	0.1801±0.0148	83.4538±1.2226
4	0.1943±0.0110	83.9796±0.6361	0.1676±0.0135	83.1325±1.0810
8	0.1794±0.0131	84.3708±0.6521	0.1594±0.0118	84.0350±1.0589

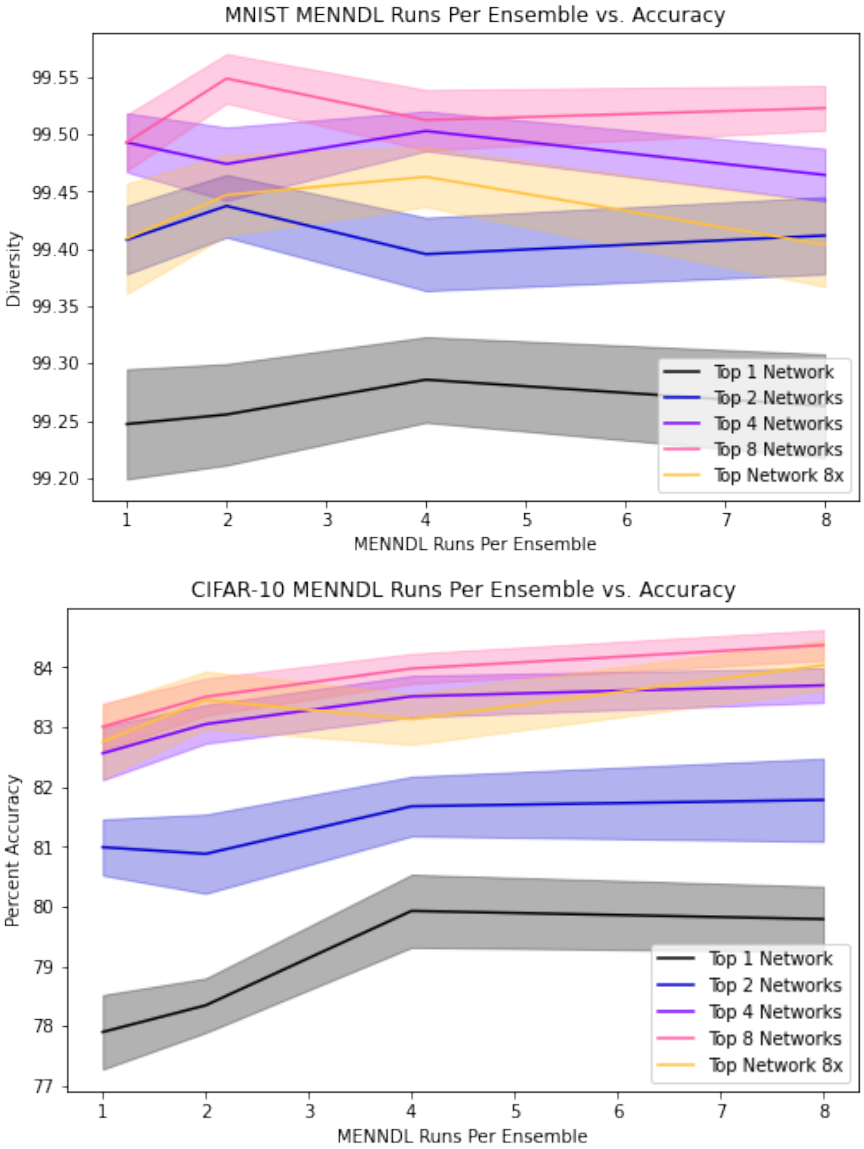

Fig. 2. MNIST and CIFAR-10 MENNDL run pool size vs. mean accuracy for the top network, ensembles of the top 2, 4, and 8 networks, and an ensemble of 8 separately trained versions of the top network.

using the best network produced by the NAS, even if we use multiple copies of that best network retrained several times. This demonstrates that the increased diversity of network structure in the ensemble produces increased diversity in predictions of the networks leading to improved ensemble performance. These results open the door to several promising directions of future work. As we have demonstrated the diversity of network structures improves performance, we will look to explicitly leverage this by evolving ensembles of networks within a NAS approach instead of simply creating an ensemble as a post-process, thus allowing the NAS to explicitly identify networks that complement each other.

6 Acknowledgements

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Robinson Pino, program manager, under contract number DE-AC05-00OR22725.

This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725.

References

1. Bakker, B., Heskes, T.: Clustering ensembles of neural network models. Neural networks **16**(2), 261–269 (2003)
2. Chatterjee, S., Bandopadhyay, S., Machuca, D.: Ore grade prediction using a genetic algorithm and clustering based ensemble neural network model. Mathematical Geosciences **42**(3), 309–326 (2010)
3. Coletti, M., Lunga, D., Berres, A., Sanyal, J., Rose, A.: Ramifications of evolving misbehaving convolutional neural network kernel and batch sizes. In: 2018 IEEE/ACM Machine Learning in HPC Environments (MLHPC). pp. 106–113 (2018)
4. Krizhevsky, A., et al.: Learning multiple layers of features from tiny images (2009)
5. LeCun, Y., Cortes, C., Burges, C.: Mnist handwritten digit database. ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist **2** (2010)
6. Melville, P., Mooney, R.J.: Creating diverse ensemble classifiers (2003)
7. Opitz, D.W., Shavlik, J.W.: Actively searching for an effective neural network ensemble. Connection Science **8**(3-4), 337–354 (1996)
8. Patton, R.M., Johnston, J.T., Young, S.R., Schuman, C.D., Potok, T.E., Rose, D.C., Lim, S., Chae, J., Hou, L., Abousamra, S., Samaras, D., Saltz, J.: Exascale deep learning to accelerate cancer research. In: 2019 IEEE International Conference on Big Data (Big Data). pp. 1488–1496 (2019)
9. Qiang, F., Shang-Xu, H., Sheng-Ying, Z.: Clustering-based selective neural network ensemble. Journal of Zhejiang University-Science A **6**(5), 387–392 (2005)
10. Real, E., Moore, S., Selle, A., Saxena, S., Suematsu, Y.L., Tan, J., Le, Q., Kurakin, A.: Large-scale evolution of image classifiers (2017)
11. Sharkey, A.J.: Combining artificial neural nets: ensemble and modular multi-net systems. Springer Science & Business Media (2012)

12. Young, S.R., Devineni, P., Parsa, M., Johnston, J.T., Kay, B., Patton, R.M., Schuman, C.D., Rose, D.C., Potok, T.E.: Evolving energy efficient convolutional neural networks. In: 2019 IEEE International Conference on Big Data (Big Data). pp. 4479–4485. IEEE (2019)
13. Young, S.R., Rose, D.C., Johnston, T., Heller, W.T., Karnowski, T.P., Potok, T.E., Patton, R.M., Perdue, G., Miller, J.: Evolving deep networks using hpc. In: Proceedings of the Machine Learning on HPC Environments, pp. 1–7 (2017)
14. Young, S.R., Rose, D.C., Karnowski, T.P., Lim, S.H., Patton, R.M.: Optimizing deep learning hyper-parameters through an evolutionary algorithm. In: Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments. MLHPC '15, Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2834892.2834896, https://doi.org/10.1145/2834892.2834896
15. Zhou, Z.H., Wu, J.X., Jiang, Y., Chen, S.F.: Genetic algorithm based selective neural network ensemble. In: Proceedings of the 17th international joint conference on Artificial intelligence-Volume 2. pp. 797–802 (2001)