Deep Forest: Towards an Alternative to Deep Neural Networks

Faezeh Khabbaz

March 2018

What the paper is about?

- This paper, proposes gcForest, a decision tree ensemble approach with performance highly competitive to deep neural networks in a broad range of tasks.

How did they position it?

- In recent years, deep neural networks have achieved great success in various applications, particularly in tasks involving visual and speech information, leading to the hot wave of deep learning.
- Though deep neural networks are powerful, they have apparent deficiencies.

Inspired by Deep Learning

- It is widely recognized that the representation learning ability is crucial for deep neural networks.
- Representation learning in deep neural networks mostly relies on the layer-by-layer processing of raw features.
- They believe that in order to tackle complicated learning tasks, it is likely that learning models have to go deep.

Cascade Forest Structure

- Each level is an ensemble of decision tree forests, i.e., an ensemble of ensembles.
- Each completely-random tree forest contains 500 completely random trees, generated by randomly selecting a feature for split at each node of the tree, and growing tree until each leaf node contains only the same class of instances.

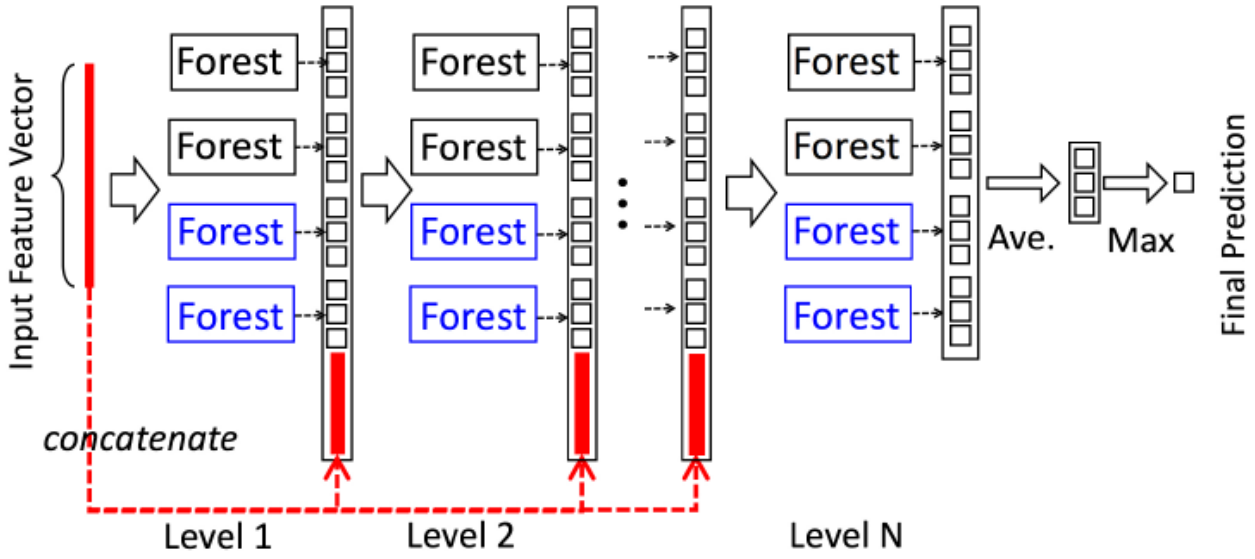


Figure 1: Illustration of the cascade forest structure. Suppose each level of the cascade consists of two random forests (black) and two completely-random tree forests (blue). Suppose there are three classes to predict; thus, each forest will output a three-dimensional class vector, which is then concatenated for re-representation of the original input.

Class vector generation

- Given an instance, each forest will produce an estimate of class distribution, by counting the percentage of different classes of training examples at the leaf node where the concerned instance falls, and then averaging across all trees in the same forest.
- The estimated class distribution forms a class vector, which is then concatenated with the original feature vector to be input to the next level of cascade.
- To reduce the risk of overfitting, class vector produced by each forest is generated by k-fold cross validation.

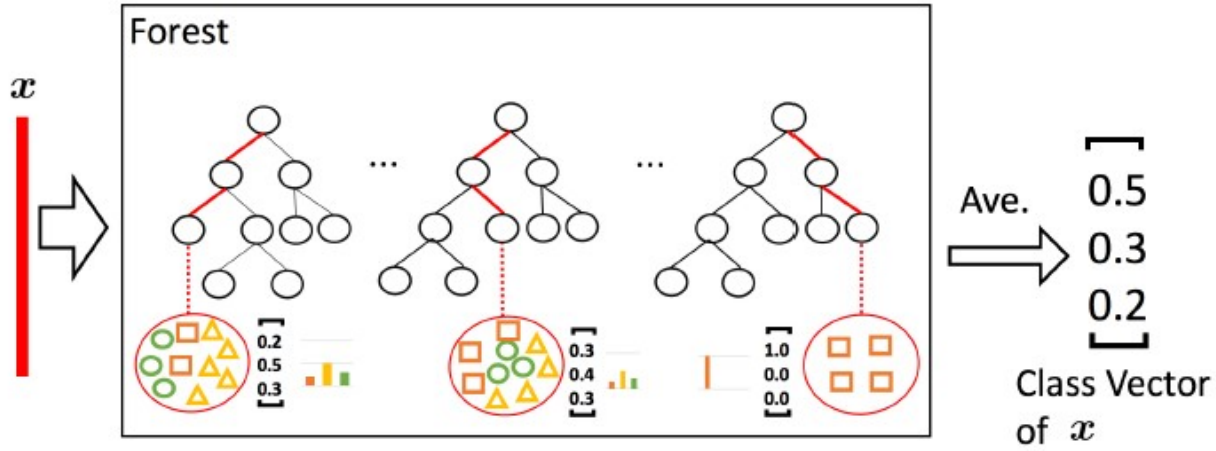


Figure 2: Illustration of class vector generation. Different marks in leaf nodes imply different classes.

Multi-Grained Scanning

- Deep neural networks are powerful in handling feature relationships, e.g., convolutional neural networks are effective on image data where spatial relationships among the raw pixels are critical; recurrent neural networks are effective on sequence data where sequential relationships are critical.
- Sliding windows are used to scan the raw features.

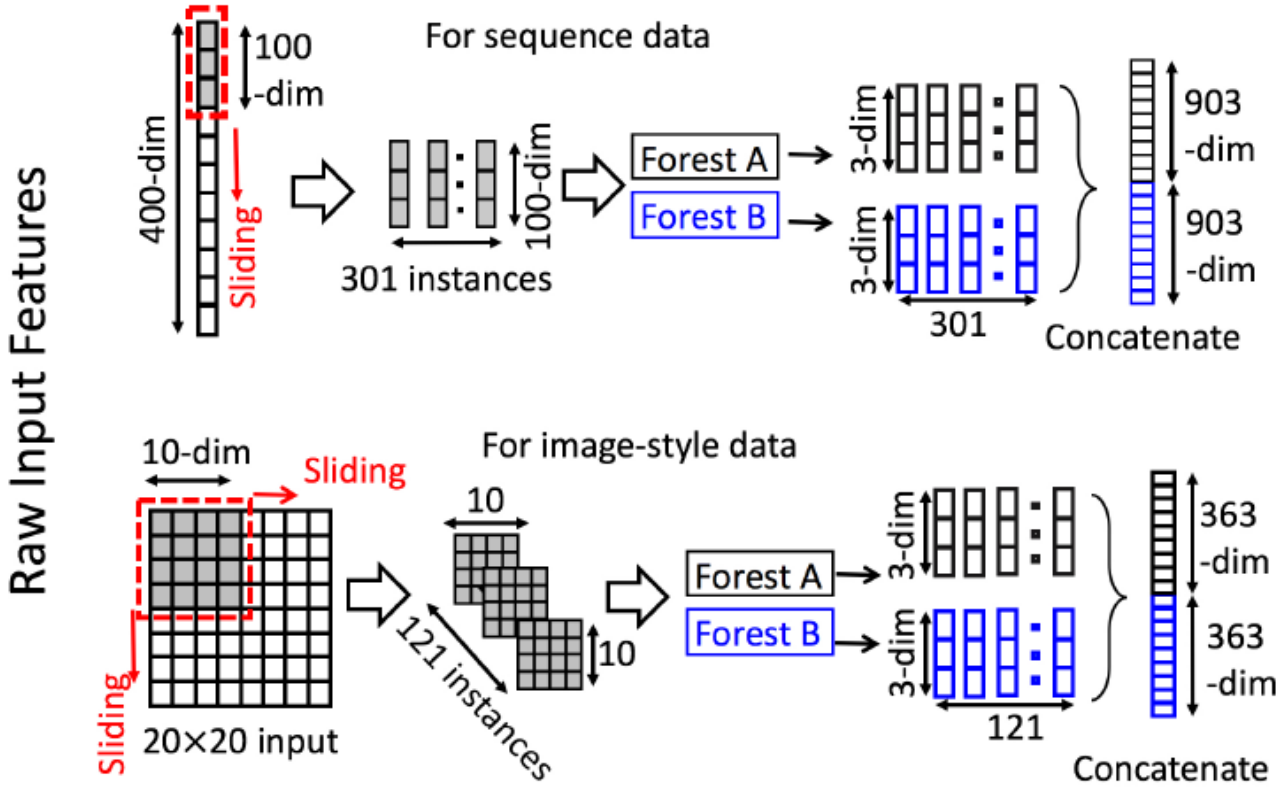
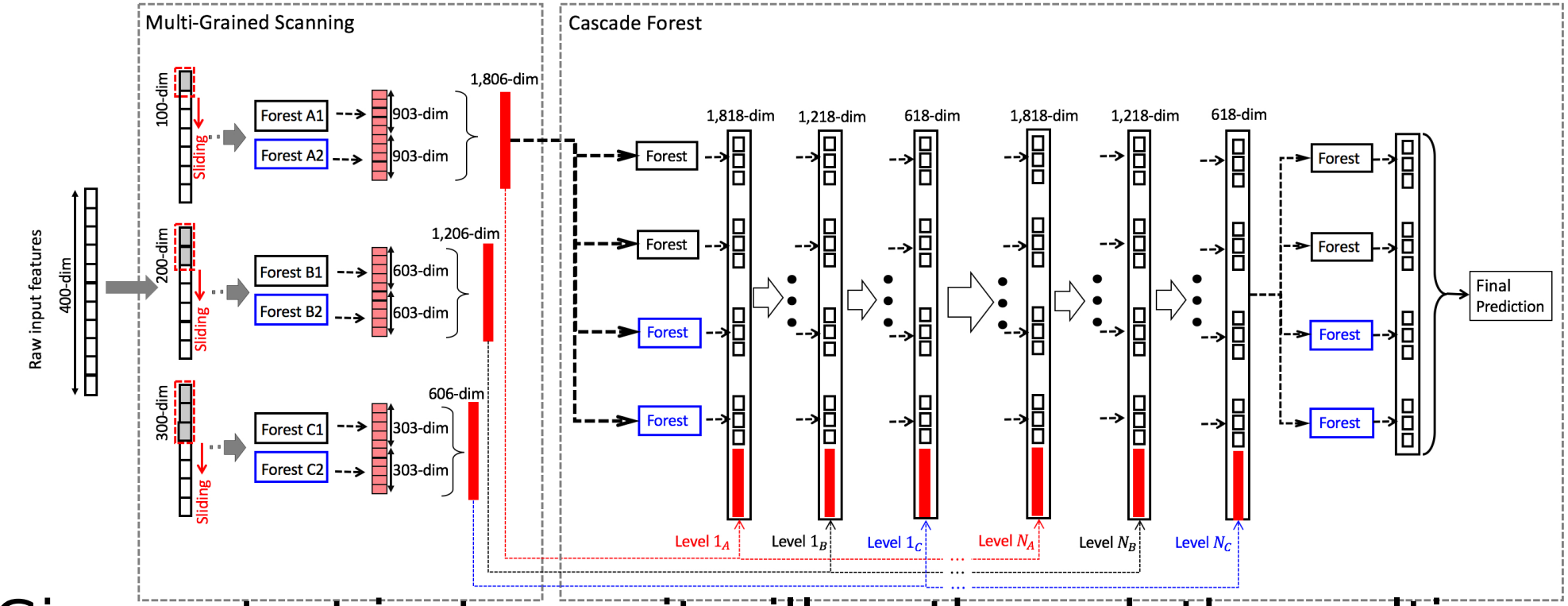


Figure 3: Illustration of feature re-representation using sliding window scanning. Suppose there are three classes, raw features are 400-dim, and sliding window is 100-dim.

The overall procedure of gcForest. Suppose there are three classes to predict, raw features are 400-dim, and three
sizes of sliding windows are used.



Given a test instance, it will go through the multi-grained scanning procedure to get its corresponding transformed feature representation, and then go through the cascade till the last level.

The hyper-parameters of deep neural networks and gcForest

Table 1: Summary of hyper-parameters and default settings. Boldfont highlights hyper-parameters with relatively larger influence; "?" indicates default value unknown, or generally requiring different settings for different tasks.

Deep neural networks (e.g., convolutional neural networks)	gcForest
Type of activation functions: Sigmoid, ReLU, tanh, linear, etc. Architecture configurations: **No. Hidden layers**: ? **No. Nodes in hidden layer**: ? **No. Feature maps**: ? **Kernel size**: ? Optimization configurations: **Learning rate**: ? Dropout: {0.25/0.50} **Momentum**: ? **L1/L2 weight regularization penalty**: ? Weight initialization: Uniform, glorot_normal, glorot_uni, etc. Batch size: {32/64/128}	Type of forests: Completely-random tree forest, random forest, etc. Forest in multi-grained scanning: **No. Forests**: {2} **No. Trees in each forest**: {500} Tree growth: till pure leaf, or reach depth 100 **Sliding window size**: $\{\lfloor d/16 \rfloor, \lfloor d/8 \rfloor, \lfloor d/4 \rfloor\}$ Forest in cascade: **No. Forests**: {8} **No. Trees in each forest**: {500} Tree growth: till pure leaf

Experiments

- They compare gcForest with deep neural networks and several other popular learning algorithms.

Configuration

- The number of cascade levels is automatically determined. If growing a new level does not improve the performance, the growth of the cascade terminates and the estimated number of levels is obtained.
- For deep neural network configurations, They use ReLU for activation function, cross-entropy for loss function, adadelta for optimization, dropout rate 0.25 or 0.5 for hidden layers according to the scale of training data.

Image Categorization

- The MNIST dataset contains 60,000 images of size 28 by 28 for training (and validating), and 10,000 images for testing.

Table 2: Comparison of test accuracy on MNIST

gcForest	**99.26%**
LeNet-5	99.05%
Deep Belief Net	98.75% [Hinton *et al.*, 2006]
SVM (rbf kernel)	98.60%
Random Forest	96.80%

Face Recognition

- The ORL dataset [Samaria and Harter, 1994] contains 400 gray-scale facial images taken from 40 persons.

Table 3: Comparison of test accuracy on ORL

	5 image	7 images	9 images
gcForest	**91.00%**	**96.67%**	**97.50%**
Random Forest	91.00%	93.33%	95.00%
CNN	86.50%	91.67%	95.00%
SVM (rbf kernel)	80.50%	82.50%	85.00%
kNN	76.00%	83.33%	92.50%

Music Classification

- The GTZAN dataset contains 10 genres of music clips, each represented by 100 tracks of 30 seconds long.

Table 4: Comparison of test accuracy on GTZAN

gcForest	**65.67%**
CNN	59.20%
MLP	58.00%
Random Forest	50.33%
Logistic Regression	50.00%
SVM (rbf kernel)	18.33%

Hand Movement Recognition

- The sEMG dataset consists of 1,800 records each belonging to one of six hand movements, i.e., spherical, tip, palmar, lateral, cylindrical and hook.
- In addition to an MLP with input-1,024-512-output structure, they also evaluated a recurrent neural network, LSTM with 128 hidden units and sequence length of 6 (500-dim input vector per second).

Table 5: Comparison of test accuracy on sEMG data

gcForest	**71.30%**
LSTM	45.37%
MLP	38.52%
Random Forest	29.62%
SVM (rbf kernel)	29.62%
Logistic Regression	23.33%

Sentiment Classification

- The IMDB dataset contains 25,000 movie reviews for training and 25,000 for testing.

Table 6: Comparison of test accuracy on IMDB

gcForest	**89.16%**
CNN	89.02% [Kim, 2014]
MLP	88.04%
Logistic Regression	88.62%
SVM (linear kernel)	87.56%
Random Forest	85.32%

Low-Dimensional Data

- They also evaluated gcForest on UCI-datasets with relatively small number of features: LETTER with 16 features and 16,000/4,000 training/test examples, ADULT with 14 features and 32,561/16,281 training/test examples, and YEAST with only 8 features and 1,038/446 training/test examples

Table 7: Comparison of test accuracy on low-dim data

	LETTER	ADULT	YEAST
gcForest	**97.40%**	**86.40%**	**63.45%**
Random Forest	96.50%	85.49%	61.66%
MLP	95.70%	85.25%	55.60%

Influence of Multi-Grained Scanning

- To study the separate contribution of the cascade forest structure and multi-grained scanning, they compared gcForest with cascade forest on MNIST, GTZAN and sEMG datasets.
- It is evident that when there are spacial or sequential feature relationships, the multi-grained scanning p

Table 8: Results of gcForest w/wo multi-grained scanning

	MNIST	GTZAN	sEMG
gcForest	99.26%	65.67%	71.30%
CascadeForest	98.02%	52.33%	48.15%

Running time

- For their experiments, they use a PC with 2 Intel E5 2695 v4 CPUs (18 cores), and the running efficiency of gcForest is good.

Relation to Other Works

- The gcForest is a decision tree ensemble approach. In particular, by using the cascade forest structure, they hope not only to do representation learning, but also to decide a suitable model complexity automatically.
- The multi-grained scanning procedure uses different sizes of sliding windows to examine the data; this is somewhat related to wavelet and other multi-resolution examination procedures.
- The cascade procedure is related to Boosting, which is able to automatically decide the number of learners in ensemble, and particularly, a cascade boosting procedure has achieved great success in object detection tasks.
- Each grade can be regarded as an ensemble of ensembles; in contrast to previous studies such as using Bagging as base learners for Boosting, gcForest uses the ensembles in the same grade together for feature re-representation.
- Passing the output of one grade of learners as input to another grade of learners is related to stacking.
- As a tree-based approach, gcForest could be potentially easier for theoretical analysis than deep neural networks

More Experiments - Appendix

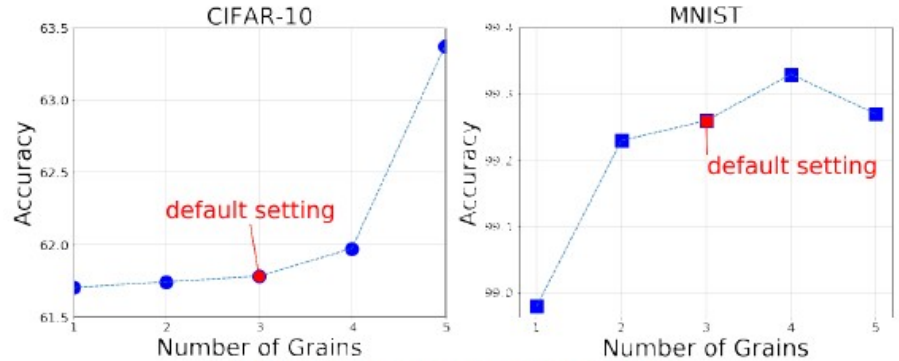
- The goal of our experiments in main body of the paper is to show that gcForest is applicable to various tasks with almost same hyper-parameter settings; this is an apparent advantage in contrast to deep neural networks that are quite sensitive to hyper-parameter settings.
- They tried the CIFAR-10 dataset which contains 50,000 colored 32 by 32 images of 10 classes for training and 10,000 imag

Table 1: Comparison of test accuracy on CIFAR-10.

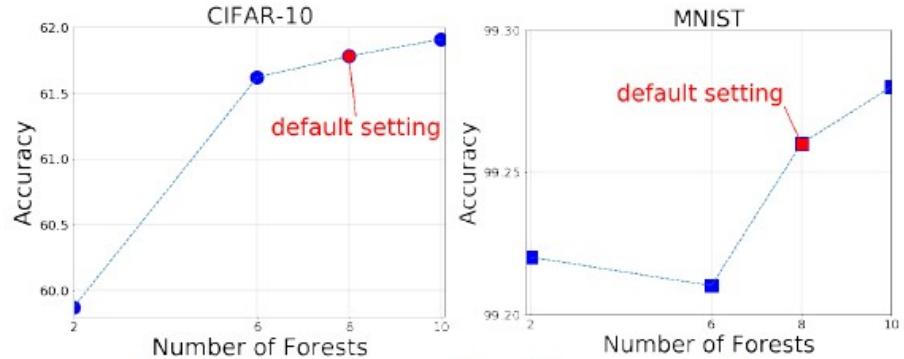
ResNet	93.57% [He *et al.*, 2016]
AlexNet	83.00% [Krizhenvsky *et al.*, 2012]
gcForest(gbdt)	**69.00%**
gcForest(5grains)	**63.37%**
Deep Belief Net	62.20% [Krizhevsky, 2009]
gcForest(default)	**61.78%**
Random Forest	50.17%
MLP	42.20% [Ba and Caruana, 2014]
Logistic Regression	37.32%
SVM (linear kernel)	16.32%

Now Comparing to more advanced DNN

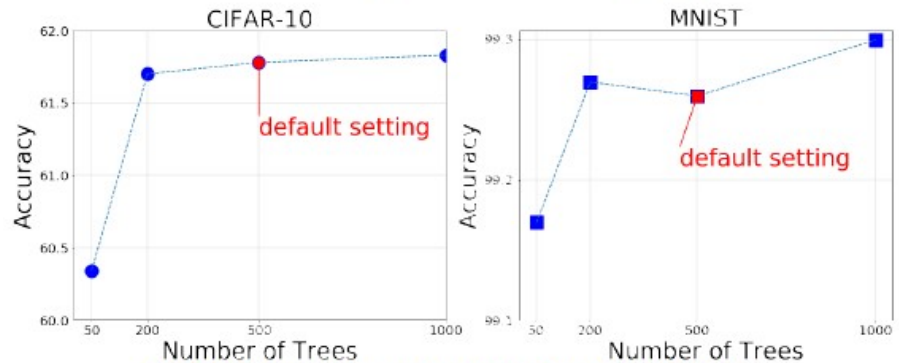
- The gcForest with default setting, i.e., gcForest(default), is inferior to state-of-the-art DNNs; however, it is already the best among non-DNN approaches.
- It could not be ignored that DNNs have been investigated for many years by huge crowd of researchers/engineers, and image tasks are killer applications of DNNs.
- “Due to limitation of computational resource, we have not tried larger models with more grains, forests and trees, although our preliminary results suggest that larger models might tend to offer better performances.” Note that computational facilities are crucial for enabling the training of larger models; e.g., GPUs for DNNs.
- There is plenty of room for improvement with distributed computing implementations.



(a) With increasing number of grains.



(b) With increasing number of forests per grade.



(c) With increasing number of trees per forest.

Figure 1: Test accuracy of gcForest with increasing number of grains/forests/trees. Red color highlights the performance achieved by default setting.

Official GitHub Page: A python 2.7 implementation of gcForest

- https://github.com/kingfengji/gcForest

Some Critics from Public

- Forests are easier to analyze than SVMs but given the rapidly growing immense body of literature doing theoretical analysis on deep learning, it is hard to see "difficulty in analyzing" DL is a valid problem to overcome.
- Deep forest are an interesting and promising idea, it’s just that I don’t think that they are an alternative to deep neural network on “typical neural network problems” (or at least, we don’t know at this point because it hasn’t been explored).
- Deep neural networks still perform quite well even when the number of labels are in the low thousands, provided that sufficient regularization (e.g., Dropout) is used to prevent overfitting.
- MNIST is a great sanity checker for machine learning but it is a poor dataset for deep learning. Choosing mostly small ‘older’ datasets, a CNN would easily overfit. Papers should show results for ‘deeper’ datasets such as CIFAR10 and Imagenet.
- Nobody who understands how neural networks work would say that this approach is capable of the kind of feature learning a neural network can do.