# GcForest

# Deep Forest: Towards an Alternative to Deep Neural Networks*

**Zhi-Hua Zhou** and **Ji Feng**
National Key Lab for Novel Software Technology, Nanjing University, Nanjing 210023, China
{zhouzh, fengj}@lamda.nju.edu.cn



## Abstract

In this paper, we propose gcForest, a decision tree ensemble approach with performance highly competitive to deep neural networks in a broad range of tasks. In contrast to deep neural networks which require great effort in hyper-parameter tuning, gcForest is much easier to train; even when it is applied to different data across different domains in our experiments, excellent performance can be achieved by almost same settings of hyper-parameters. The training process of gcForest is efficient, and users can control training cost according to computational resource available. The efficiency may be further enhanced because gcForest is naturally apt to parallel implementation. Furthermore, in contrast to deep neural networks which require large-scale training data, gcForest can work well even when there are only small-scale training data.

ample, even when several authors all use convolutional neural networks [LeCun *et al.*, 1998; Krizhenvsky *et al.*, 2012; Simonyan and Zisserman, 2014], they are actually using different learning models due to the many different options such as the convolutional layer structures. This fact makes not only the training of deep neural networks very tricky, like an art rather than science/engineering, but also theoretical analysis of deep neural networks extremely difficult because of too many interfering factors with almost infinite configurational combinations.

It is widely recognized that the *representation learning* ability is crucial for deep neural networks. It is also noteworthy that, to exploit large training data, the capacity of learning models should be large; this partially explains why the deep neural networks are very complicated, much more complex than ordinary learning models such as support vector machines. We conjecture that if we can endow these properties to some other suitable forms of learning models, we may be able to achieve performance competitive to deep neural networks but with less aforementioned deficiencies.

# Cascade Forest Structure

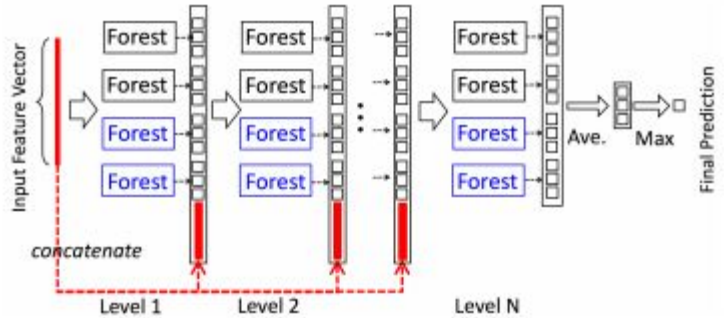


Figure 1: Illustration of the cascade forest structure. Suppose each level of the cascade consists of two random forests (black) and two completely-random tree forests (blue). Suppose there are three classes to predict; thus, each forest will output a three-dimensional class vector, which is then concatenated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

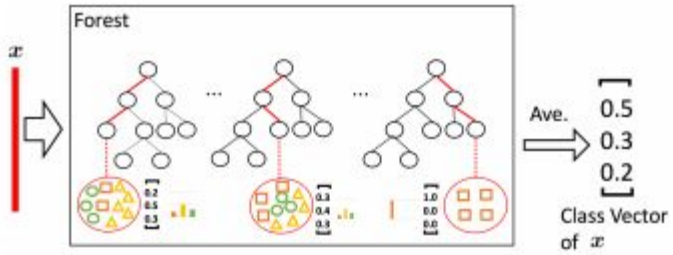


Figure 2: Illustration of class vector generation. Different marks in leaf nodes imply different classes.

# Multi-Grained Scanning

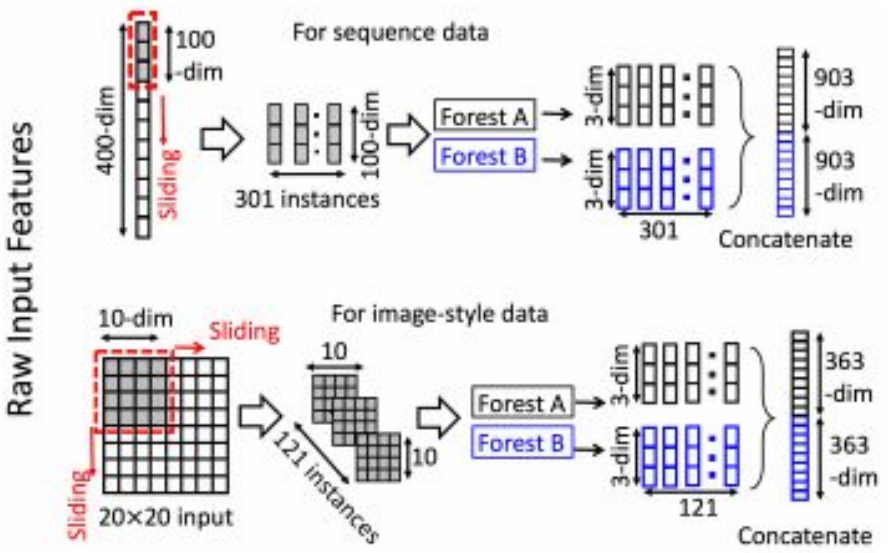


Figure 3: Illustration of feature re-representation using sliding window scanning. Suppose there are three classes, raw features are 400-dim, and sliding window is 100-dim.

# Overall Procedure and hyper-Parameters

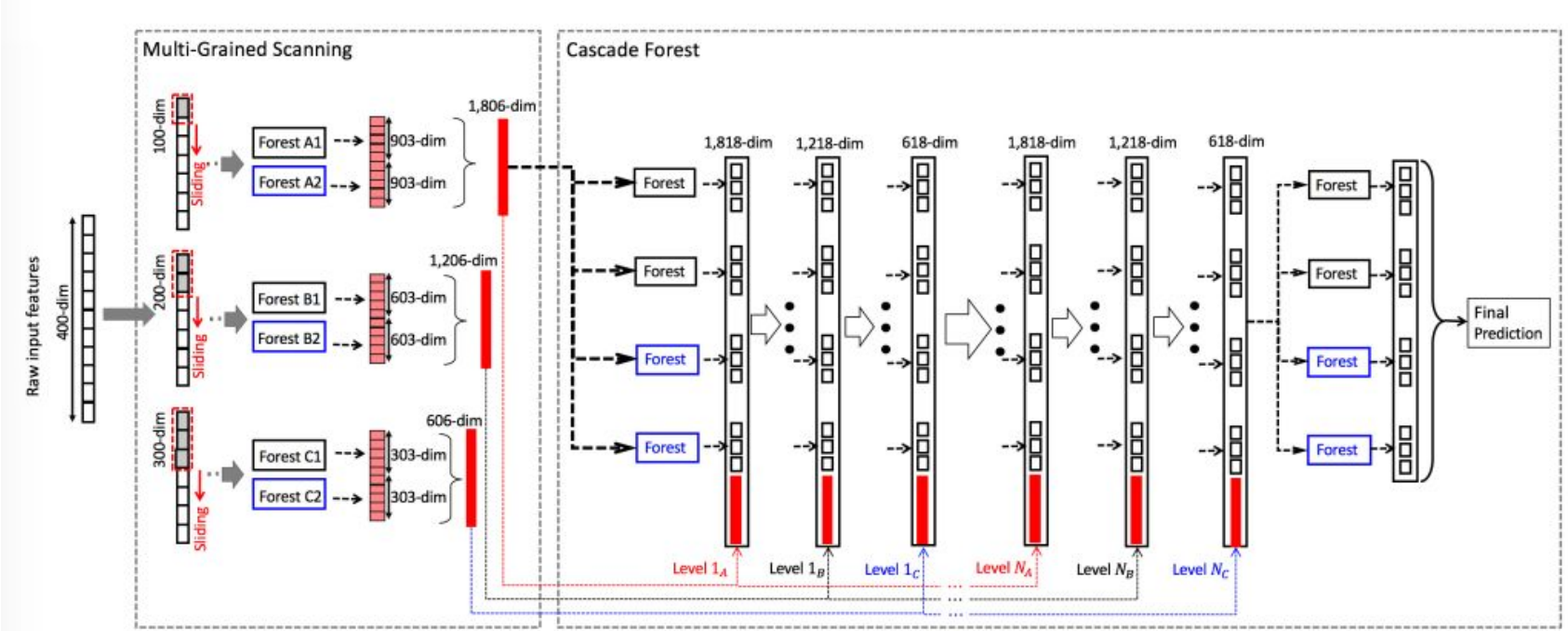


Figure 4: The overall procedure of gcForest. Suppose there are three classes to predict, raw features are 400-dim, and three sizes of sliding windows are used.

# compare with DNN

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

# Results

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% |

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% |

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% |

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% |

# 讨论、争议

## 如何评价周志华教授新提出的 Deep Forest 模型，它会取代当前火热的深度学习 DNN 吗？

目前该模型的paper已经在arXiv上可阅读： Towards An Alternative to Deep Neural Networks. 模型的框...显示全部 ∨

**1. Idea**

不用多说，通过stacking weak learner来提升模型性能的想法已经非常常见。然而这样一个idea一定不会work，原罪就在于tree based model无法进行fine-tune。这会带来两个非常严重的问题：

A. 无法进行end to end learning，这会极大程度上影响到模型最终的结果。会使depth完全发挥不出威力。

B. 无法进行feature transfer，CNN最最最重要成功的秘诀就在于可以在ImageNet这样海量的数据上进行pretrain，然后再把学习到的(approximately) general feature 通过finetune transfer到其他任务中去。如果BP都没法做，这个就完全无从谈起了。

换句话说，如果这样的Idea能在大数据上work，那么其实相当于否定了BP存在的意义。以后大家都一层一层加就是了嘛。当然这样的好事是不存在的。。。

如何把tree based model可以做到e2e训练，以及拓展开来如何更高效地把更高效的base learner引入神经网络这又是另外一个故事。我们在这方面也做了一些尝试，但是所有的这一切的基础都是可以bp。

**2. Method**

这里使用了一个stacking random forest的方法来实现上述idea。从创新性上而言，在Kaggle比赛中不同种类模型之间的Stacking和Concatenation早就不是什么新想法。所以从实现层面上而言，仍旧乏善可陈。而且中间层只是把最终分类的结果concat在一起，都不是把每个tree生成的leaf对应的feature。个人觉得这样会严重影响stack的性能，好歹之前FB还是把GBDT的每个leaf当成一个feature才去train的logisitic regression。

**3. Experiment**

这是本文槽点最多的部分。看到还在跑MNIST和ORL就知道这完全是做ML的人的玩具。不熟悉别的应用，单就Vision而言，这个结果比ELM其实还没说服力，而且我相信会有更简单的模型同样可以达到类似的结果。比如随便开开脑洞train个SVM中间加个non-linear transformation什么的。现在在CVPR大家都知道Cifar都不能算是有信服力的数据集，更何况MNIST和ORL这种。

另外，比较的网络结构也都是拍拍脑袋想出来的，唯一比了一个经典结构LeNet5还没比过。。不过基于上述第一点的看法，我完全不相信这样的算法能在稍微大一些的数据集上取得还不错的结果。

# 关于gcForest这篇论文 我们请教了周志华教授以下几个问题

本文作者：宗仁 2017-03-06 19:09

> 导语：今天下午，雷锋网参加了由中国人工智能学会（CAAI）主办，南京大学周志华教授担任学术负责人的人工智能前沿讲习班 - 机器学习前沿。

**1. 周老师那篇arXiv的文章还是在强调“深度”，并没有完全否定深度学习？您只是提出了另一种做深的方法？使得很多参数更稳定鲁棒，也是要依靠大数据。请问这么想对不对？**

主要思想是，现在大家谈到深度学习就觉得它就等于深度神经网络。我们认为解决复杂问题把模型变深可能是有必要的，但是深度学习应该不只是深度神经网络，还可以有其他形式，与神经网络相比，其他形式也许有更好的性质。

**2. 昨天田渊栋老师在知乎上提到一点， Multi-Grained Scanning这部分非常像1D和2D Convolution。“另外实验还只是在小规模数据集上做的，期待CIFAR甚至是ImageNet的结果。深度学习这里也有一直在提但是一直效果不怎么好的Layer-by-Layer训练的思路，如果这个思路能在大数据集上做好，那确实是大突破了。”**

**您如何看待田老师说的“还只是在小规模数据集上做的，期待CIFAR甚至是ImageNet的结果”这个说法的？**

我们更关心的是这件事可以怎么去做。以前大家不知道这个事情可以这样去做，我们现在把这种可能性展示出来，这是基础研究所追求的。 新东西出来的时候，性能不是很重要，因为很多方面都没有优化。

性能本身并不是我们现在很关心的事情，因为改进的空间很大，可以有各种途径去提高，这都是后面的事情了。 在大数据上做需要更大的资源、更多人力、物力，这是我们暂时所不具备的。工业界会更关注性能，我们也希望看看更大的资源来了之后，能做到什么样程度。

**3. 周老师您在CNCC2016上，指出机器学习的近期目标是需要利用更多的计算资源及更多的数据。您新近提出的gcForest如何实现这一点？**

要做大数据、复杂问题，肯定要用更强大的计算资源。现在有一些计算架构是非常适合这样的模型的。有了更大的资源之后，我们也很好奇它到底能做到什么程度。