ENSEMBLE OF NARROW DNN CHAINS

Tinghao Xie
Zhejiang University*
vtu@zju.edu.cn

ABSTRACT

Deep neural networks (DNN) have been widely adopted for various real-world applications. With the growth of computational capability, DNNs are growing larger and reaching better performance. Meanwhile, the time for training a DNN could vary from hours to days and even months, and the size of a DNN could vary from MB to GB level, depending on the task. It is also known that state-of-the-art DNNs are usually redundant. Here we aim at reducing large DNN models into smaller pieces by looking back into ensemble methods for binary classiers in multi-class problems, but with one/two-channel narrow DNN chains as the base classifiers. We propose the *Ensemble of Narrow DNN Chains* (*ENDC*) framework: first train such narrow DNN chains that perform well on one-vs-all binary classification tasks, then aggregate them together by voting to predict for the multiclassification task. Our ensemble framework could utilize the abstract interpretability of DNNs, while being 2-4 orders of magnitude smaller than normal DNN and > 6 times smaller than traditional ML models, furthermore compatible with full parallelism in both the training and deployment stage. Our empirical study shows that a narrow DNN chain could learn binary classifications well. Moreover, our experiments on three popular datasets confirm the potential power of ENDC. Compared with traditional ML models, ENDC, with the smallest parameter number, could achieve similar accuracy on MNIST and Fashion-MNIST, and significantly better accuracy on CIFAR-10.

1 Introduction

Deep learning [1, 2, 3, 4, 5, 6, 7, 8] has taken over the dominance in various real-world applications (images, natural languages, speeches, videos, *etc*), even surpassing human's performance. As computational capability grows, these models are also growing deeper [3] and performing better. However, the model's size, training time, and inference time are growing swiftly. Take VGG-16 [9] as an example; training it on ImageNet [10] could take 2-3 GPU weeks, an inference could cost >3s on CPU, and the model size could reach >500MB. On the one hand, for resource-critical scenarios, *e.g.* edge devices, embedded systems, automobiles, such costs of large deep neural networks (DNNs) may not be affordable. On the other hand, [11] points out the redundancy in DNNs. To ease this concern, compact light-weight DNNs [12, 13] are designed, network pruning is adopted for reducing the parameter size, *etc*.

Another motivation of this paper is the independent mechanisms. The spirit is that most physical processes have a modular structure, with each module only interacting sparsely. For example, two cars on the Earth can be modeled separately, even though there is always gravity between a car and the Earth itself. Only when the two cars crash should they be considered jointly. When it comes to learning, it's not surprising that we can correctly recognize different objects without re-learning them all over again. This is generally attributed to the fact that human intelligence is considered to utilize independent mechanisms, which are modular, re-usable and broadly applicable [14]. Deep learning models can fit into the concept of independent mechanisms, *e.g.* Recurrent Independent Mechanisms [15], a deep learning architecture in which multiple recurrent cells operate nearly independently and communicate only sparingly through the bottleneck of attention. While fully connections and dense structures are usually considered better, this line of work goes the contrary by exploring sparse architectures.

In this paper, we look back into ensemble methods for binary classifiers in multi-class problems and propose a sparse aggregation framework, *Ensemble of Narrow DNN Chains* (*ENDC*), as both a countermeasure to DNN's redundancy

[1]Completed as a visiting student at the University of Oxford.

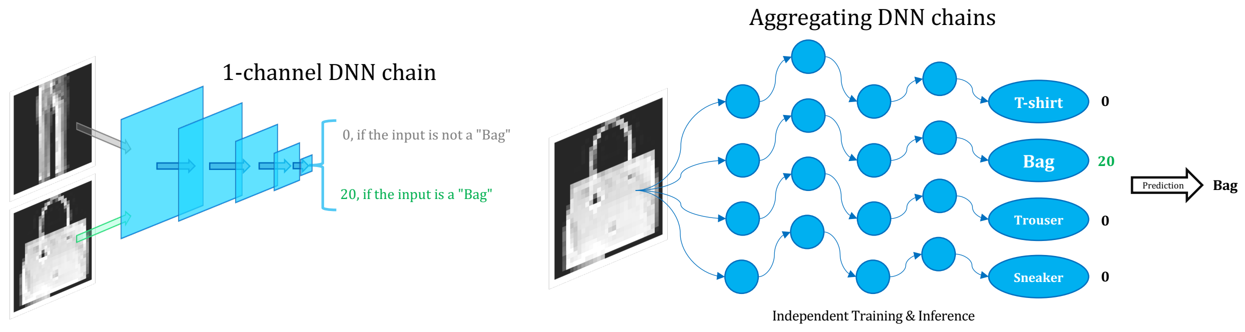

1-channel DNN chain
0, if the input is not a "Bag"
20, if the input is a "Bag"
Aggregating DNN chains
T-shirt
0
Bag
20
Trouser
0
Sneaker
0
Prediction
Bag
Independent Training & Inference

Figure 1: **Ensemble of Narrow DNN Chains (ENDC) framework.** We reduce the multi-classification task into multiple binary ones. As shown in the left, we train 1-channel DNN chains to decide for these one-vs-all binary classification problems, *e.g.* answering the question "is the input in class Bag?"; *i.e.* the narrow DNN chain outputs large activation values (like 20) for "Bag" inputs, but small (like 0) activation values for other inputs. Then we aggregate these DNN chains together by voting, predicting the class with the largest logit (as convention).

and a baseline for independent convolutional architecture. As shown in Figure 1, we reduce the multiclassification problem into several one-vs-all (OVA) [16, 17] binary classification problems, then train narrow DNN chains (with only 1 or 2 channels) as the base binary classifiers, eventually aggregate them by voting. ENDC: 1) utilizes DNN's abstract interpretability; 2) heavily reduces the model's parameter number by abandoning all internal cross-connections; 3) requires less training time and fewer training samples; 4) allows full parallelism and separation during both the training and inference stage.

To summarize, our main contributions include:

- We empirically study the learning ability of narrow DNN chains with convolutional structures on image tasks. Our experiments show that a narrow DNN chain could learn binary classifications well. Specifically, it can recognize a certain target class and discriminate it from other classes, though the classification quality depends on the input size and the exact target class.
- We propose *Ensemble of Narrow DNN Chains* (*ENDC*), a sparse aggregation framework of very narrow DNN chains. Our experiments on three popular datasets show the potential power of sparse structure with convolutions. Compared with traditional ML models, ENDC, with the smallest parameter number, could achieve similar accuracy on MNIST and Fashion-MNIST, and significantly better accuracy on CIFAR-10.
- We provide our insights for the experiment results and point out several promising future work directions. We analyze the binary classification qualities, discuss the superiority of ENDC over state-of-the-art DNNs, and point out future work directions based on sparse architectures as ENDC.

2 Related Work

Narrow DNNs. [18] first studies the nature of super-narrow deep architectures and shows that given enough layers, they can shatter any separable binary datasets. However, they only focus on MLP-fashioned deep architectures and discuss simple artificial datasets, since the work was born in an era when various nowaday-popular computer vision datasets had not been cast yet. [19] points out the depth of a DNN contributes to the expressive power more than the width. Meanwhile, [20] proposes a wide 16-layer residual structure, achieving the same performance compared with a 1000-layer narrower DNN. Since the width, depth, and capacity of DNNs all contribute to state-of-the-art performance on complex multiclassification tasks, most existing research has not focused on very narrow DNNs (with only one or two channels). An approximate trial is network pruning [21, 22], a line of methods that remove useless and redundant neurons from large DNNs and therefore reduce the width of the models. A recent work [23] trains narrow DNN chains to recognize abstract patterns and then utilizes these chains to inject neural backdoors into DNNs. Motivated by previous work, we train narrow DNN chains to recognize real-world objects and empirically study their expressive power on popular datasets.

Ensemble of Binary Classifiers. End-to-end DNN framework has shown its strength for multiclassification tasks, relying on its large capacity and dense internal connections. But for complex multiclassification tasks, a more intuitive and practical option is to divide the multiclassification problem into several binary classification problems, which are easier to solve. The most common strategies for decomposing multiclassification problems are "one-vs-one" (OVO) [24] and "one-vs-all" (OVA) [16, 17]: 1) OVO divides the problem into binary problems between pairs of classes, each binary classifier learns to discriminate between each pair of classes, and their outputs are aggregated to predict; 2) OVA divides a K-classification problem into K binary problems, each binary classifier learns to recognize one class and discriminate it from the others, and their outputs are aggregated to predict. [25] provides an overview of OVO and OVA, concluding that OVO is generally better than OVA. Ensemble methods either adopt traditional ML models [25, 26, 27] or neural networks ([16, 28]) as the base classifiers. In this work, we follow the OVA decomposition, since in many multiclassification problems where classes are mostly independent (***e.g.*** "trouser" and "bag"), it's more intuitive (even for humans) to distinguish one against all the others, but not between a pair. And as DNN is a nice weapon to distinguish by these abstract features, we adopt narrow DNN chains as binary base classifiers.

3 Method

3.1 Overview

In this chapter, we describe our ensemble framework ENDC in two steps (as shown in Figure 1):

1. Training several narrow DNN chains as the OVA base binary classifiers (3.2)
2. Aggregating these binary classifiers by voting (3.3)

Before that, we formally introduce our base classifiers, narrow DNN chains. A narrow DNN chain is yet another deep neural network, but with a much smaller width and a scalar output. Specifically:

Definition 1. *Narrow DNN Chain*

A narrow DNN chain with a small width W and a depth L is a general feedforward neural network $F(x;\mathbf{w}_i)$ with x as its input, parameterized by $\mathbf{w}$, and outputs a scalar. The narrow DNN chain has L fully connected or convolutional layers. Each layer of the narrow DNN chain is described by $\mathcal{V}_i = \{v_i^{(1)}, v_i^{(2)}, \dots, v_i^{(n_i)}\}$ where each v_i^j denotes a neuron (for fully connected layers) or channel (for convolutional layers) and

$$n_L = 1, n_i \le W, \forall i \in \{1, 2, \dots, L-1\}$$

For each neuron v, its input is denoted as $\mathcal{I}_v$ and output as $\mathcal{O}_v$, and they could be either scalar (for fully connected layers) or vector (for convolutional layers). Each neuron's output is defined as $\mathcal{O}_v = \mathcal{I}_v$ or $\mathcal{O}_v = \sigma(\mathcal{I}_v)$, where σ could be any non-linear activation function. The forwarding process between layers is defined as:

$$\mathcal{I}_v = \begin{cases} \sum_{u\in\mathcal{V}_{i-1}} w_{uv}\mathcal{O}_u & V_i \text{ is a fully connected layer} \\ \sum_{u\in\mathcal{V}_{i-1}} w_{uv}\circ\mathcal{O}_u & V_i \text{ is a convolutional layer} \end{cases}$$

where w_{uv} is the connection edge from u to v, and $\circ$ denotes the convolution operator. Concretely, w_{uv} is a linear weight if v is in a fully connected layer, and is a convolutional kernel if v is in a convolutional layer. Other details are similar to the generally accepted definition for DNNs and thus omitted.

3.2 Narrow DNN Chains as ENDC Base Binary Classifiers

As mentioned earlier, we follow the OVA style and decompose the K-classification problem into K binary classification problems. Each binary classifier is assigned a target class $\mathcal{C}_i (i = 1, 2, \dots, K)$, and should distinguish class $\mathcal{C}_i$ from all other classes $\overline{\mathcal{C}}_i = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_K\} \backslash \{\mathcal{C}_i\}$. We adopt narrow DNN chains as base classifiers, and therefore we expect them to output differently for their target class and other classes, respectively. Assuming the data distribution of each class $\mathcal{C}_i$ is $\mathcal{D}_i$, its complement classes $\overline{\mathcal{C}}_i$'s data distribution is $\overline{\mathcal{D}}_i$:

Definition 2. *ENDC Base Binary Classifiers*

Given a K-classification problem, the ENDC base binary classifiers are K narrow DNN chains, each assigned to a target class $\mathcal{C}_i (i = 1, 2, \dots, K)$. Each narrow DNN chain $F_i(x;\mathbf{w}_i)$ with its target class $\mathcal{C}_i$ should satisfy the following:

- $F_i(x) \approx a \gg 0, \forall x \in supp(\mathcal{D}_i)$
- $F_i(x) \approx 0, \forall x \in supp(\overline{\mathcal{D}}_i)$

Table 1: **Results Overview of ENDC.** We test ENDC on three datasets (MNIST, Fashion-MNIST, CIFAR-10) and report the overall accuracy here. We use 1-channel VGG as the base classifier for MNIST and Fashion-MNIST tasks, and 2-channel VGG for CIFAR-10 task. "# Param" column reports the number of parameter in our ENDC framework.

Dataset	Accuracy (%)	Arch	# Param
MNIST	93.40	1-channel	1300
Fashion-MNIST	80.39	1-channel	1300
CIFAR-10	47.72	2-channel	4930

In a simple sentence, each base binary classifier is a narrow DNN chain that fires large activations (*e.g.* 20) when seeing inputs from its target class and stays inactive (*i.e.* 0) when seeing inputs from other classes.

Similar to conventional deep learning models training, we may train the narrow DNN chains following the objective:

$$\min_{\mathbf{w}_i} \mathop{\mathbb{E}}_{x\sim\mathcal{D}_i}\Big(F_i(x;\mathbf{w}_i)-a\Big)^2+\lambda_i \mathop{\mathbb{E}}_{x\sim\overline{\mathcal{D}}_i}\Big(F_i(x;\mathbf{w}_i)-0\Big)^2,\ \forall i\in\{1,2,\dots,K\} \tag{1}$$

where λ_i controls the trade-off between the two sub-objectives.

3.3 Aggregation by Voting

The voting strategy (also called maximum confidence strategy, binary voting, or Max-Wins rule [29]) is the simplest aggregation method to put up all the binary classifiers. The prediction class is taken from the classifier with the largest output:

$$\text{Prediction} = \mathop{\arg\max}_{i=1,2,\dots,K} F_i(x)$$

As long as each trained base binary classifier could behave as Definition 2, we may expect the entire ENDC framework to predict correctly for the multiclassification problem.

4 Experiments

In this section, we conduct experiments to: 1) empirically study narrow DNN chains' expressive power 2) demonstrate the feasibility of ENDC by comparison with normal DNNs and traditional ML models 2) provide insights into our results.

4.1 Setup

Datasets. We test ENDC on three popular datasets, MNIST [30], Fashion-MNIST [31], and CIFAR-10 [32]. MNIST is composed of 1-channel 28x28 inputs of handwritten digits from 0 to 9; Fashion-MNIST is a similar dataset but with 1-channel 28x28 inputs of fashion objects like trousers and bags; CIFAR-10 is a more complex dataset where inputs are 3-channel 32x32 images of cars, birds, *etc*.

Models. We modify a traditional convolutional architecture, VGG [4], as our base classifiers. Specifically, we only adopt VGG's convolutional (feature extraction) part with a scalar output, limiting its width to one channel (for MNIST and Fashion-MNIST tasks) or two channels (for CIFAR-10 tasks). We train each narrow DNN chain independently following Eq (1), and decide whether to stop training on a validation set divided from the full train set.

Metrics. We record the overall and classwise accuracy for each task. We compare the accuracy and model parameter number of ENDC with normal DNNs and traditional ML models, respectively. We briefly discuss the training and inferencing stage compared with normal DNNs.

4.2 Results

Table 1 shows the overview of our results and Table 2 shows the classwise accuracy. Tables 3, 4, 5 list out the comparison of ENDC, traditional ML (Support Vector Classifier or SVC, Logistic Regression or LR), and normal DNNs (LeNet [1], VGG [4]) for the three datasets respectively.

Table 2: **Accuracy Per Class of ENDC.**

Dataset	#0 (%)	#1 (%)	#2 (%)	#3 (%)	#4 (%)	#5 (%)	#6 (%)	#7 (%)	#8 (%)	#9 (%)
MNIST	97.04	97.53	96.51	88.91	95.52	92.38	90.29	94.55	88.71	91.67
Fashion-MNIST	80.60	92.90	77.60	77.60	75.50	92.30	40.70	81.30	90.00	95.50
CIFAR-10	48.90	55.70	43.50	31.80	41.00	45.40	61.90	42.00	49.90	57.10

Table 3: **Comparison on MNIST.** "LR" for Logistic Regression, "SVC" for Support Vector Classifier. Results are referenced from internet, see citations for sources.

Method	Accuracy (%)	# Param
ENDC (ours)	**93.4**	**1.3K**
LR	91.7 [33]	7.7K+
SVC	97.8 [33]	7.7K+
Normal DNN (LeNet)	99.3 [34]	0.41M

As shown, when compared with traditional ML models, ENDC always has fewer parameters by more than 6 times (the parameter number of each binary classifier is even smaller than the input size). ENDC achieves similar accuracy on simpler tasks, MNIST and Fashion-MNIST. Thanks to CNN's abstract interpretability, ENDC performs significantly better than carefully designed traditional ML models on CIFAR-10. Still, ENDC is weaker than state-of-the-art DNNs (especially in the CIFAR-10 task), but we will discuss ENDC's superiority over normal DNNs on model size, convenience at training and inferencing. Moreover, we are looking forward to further work making ENDC-like ensemble/modular framework stronger, even surpassing nowadays state-of-the-art but redundant DNNs.

We also demonstrate all 3x10 binary base classifiers' activation distribution histograms, receiver operating characteristic curves (ROC), and area under the curve (AUC) on the test set in Figure 2, 3 and 4, and discuss them in the next section.

4.3 Discussions

To how much could the narrow DNN chains learn the binary classification task? See Figure 2, 3 and 4. We can say the narrow DNN chains learn MNIST OVA tasks well (all AUC > 0.96), and there are clearly large gaps between the two classes. For the Fashion-MNIST task, the binary classifiers for most classes are good (AUC > 0.92). But the binary classifier for class 6 is significantly worse (see Figure 3g and 3q, AUC = 0.83), and ENDC's accuracy for class 6 is equivalently bad (40.70%, see Table 2). And for CIFAR-10, the narrow chains are performing even worse (AUC > 0.76), but still much better than random guesses (AUC = 0.5). To sum up, though only having an extremely small capacity and a parameter number even less than the input entries, a narrow chain can learn to recognize a certain target class, and discriminate it between the rest of the classes.

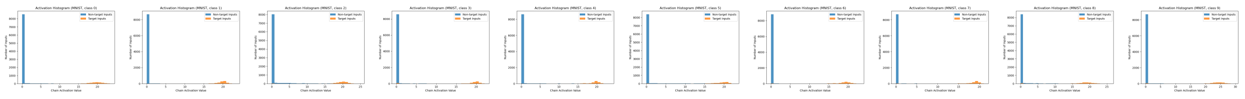
What are the inadequacies of narrow DNN chains as base classifiers? First of all, in these datasets, different classes are not completely independent. For example in Fashion-MNIST, class 6 is "Shirt" which consists of various types of upper outer garments, and they share close relationships with other classes like "T-shirt", "Pullover", and "Coat" (which are even difficult for humans to distinguish). And in CIFAR-10 task, the animal classes "bird" (class 2), "cat" (class 3), "deer" (class 4), and "horse" (class 7) share similar features. And completely independent voting chains are not enough to notice these shared features, therefore leading to worse AUC (compared with others classes that are more independent). Secondly, we use narrow architecture, which may not have enough capacity to capture all

Table 4: **Comparison on Fashion-MNIST.** "LR" for Logistic Regression, "SVC" for Support Vector Classifier. Results are referenced from internet, see citations for sources.

Method	Accuracy (%)	# Param
ENDC (ours)	**80.4**	**1.3K**
LR	84.2 [33]	7.7K+
SVC	89.7 [33]	7.7K+
Normal DNN (VGG-16)	93.5 [35]	26M

Table 5: **Comparison on CIFAR-10.** "LR" for Logistic Regression, "SVC (PCA)" for Support Vector Classifier with Principal Component Analysis to reduce input dimensions. Results are referenced from internet, see citations for sources.

Method	Accuracy (%)	# Param
ENDC (ours)	**47.7**	**4.8K**
LR	39.9 [36]	30.0K+
SVC (PCA)	40.2 [37]	0.44M+
Normal DNN (VGG-16-BN)	93.9 [38]	15M


(a) M (0) (b) M (1) (c) M (2) (d) M (3) (e) M (4) (f) M (5) (g) M (6) (h) M (7) (i) M (8) (j) M (9)

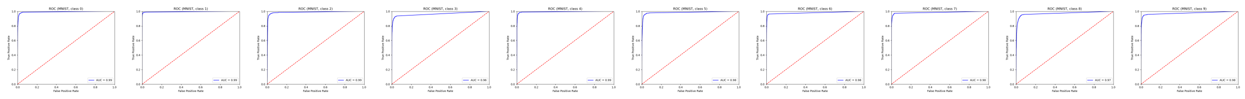
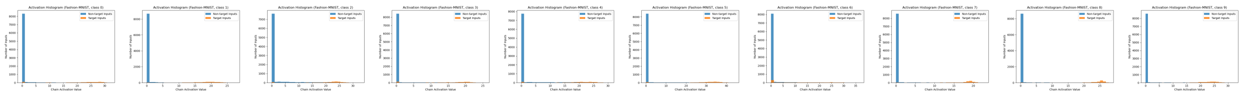

(k) M (0) (l) M (1) (m) M (2) (n) M (3) (o) M (4) (p) M (5) (q) M (6) (r) M (7) (s) M (8) (t) M (9)

Figure 2: **Activation Distribution Histograms and ROC Curves of Narrow DNN Chains (MNIST).**


(a) F (0) (b) F (1) (c) F (2) (d) F (3) (e) F (4) (f) F (5) (g) F (6) (h) F (7) (i) F (8) (j) F (9)

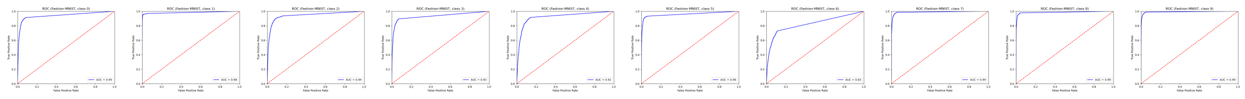
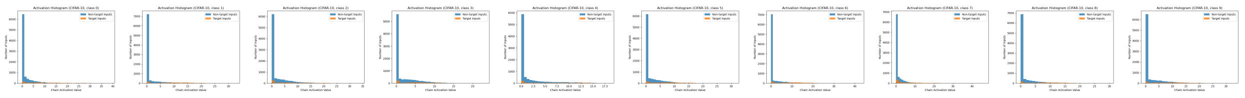

(k) F (0) (l) F (1) (m) F (2) (n) F (3) (o) F (4) (p) F (5) (q) F (6) (r) F (7) (s) F (8) (t) F (9)

Figure 3: **Activation Distribution Histograms and ROC Curves of Narrow DNN Chains (Fashion-MNIST).**


(a) C (0) (b) C (1) (c) C (2) (d) C (3) (e) C (4) (f) C (5) (g) C (6) (h) C (7) (i) C (8) (j) C (9)

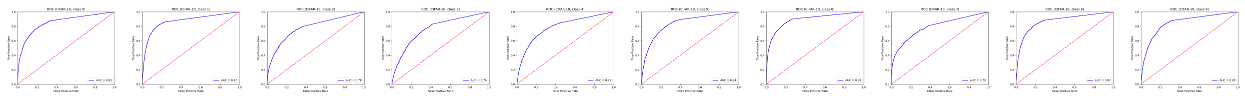

(k) C (0) (l) C (1) (m) C (2) (n) C (3) (o) C (4) (p) C (5) (q) C (6) (r) C (7) (s) C (8) (t) C (9)

Figure 4: **Activation Distribution Histograms and ROC Curves of Narrow DNN Chains (CIFAR-10).**

features that help distinguishment. This is obvious and our experimental attempts prove this – when we increase the chain's width, generally we could train better classifiers. But we leave this scalability factor to future work, as well as the best design for narrow chain architectures.

What's the superiority of ENDC over normal DNNs? A huge advantage of ensemble frameworks is that both the training and inference could be arbitrarily distributed w.r.t. each class. Furthermore, there are much fewer parameters (by 2 to 4 orders of magnitude), so the optimization could converge much quicker. In our experiments, we only train each narrow DNN chain within <5 epochs. Likewise is the inference stage, where ENDC's good separability even allows the framework to be efficiently deployed on CPU via multiple threads or processes.

How is ENDC meaningful and what are the potential future directions? 1) ENDC demonstrates the power of ensembling extremely small DNNs (each chain's parameters are fewer than the input entries), showing that such deep models with tiny capacity could make up a satisfying multiclassifier. 2) Through the activation histograms and ROC curves, we can see that the difficulty to learn different classes could vary a lot. Obviously, uniformly using an identical module for each label is not making much sense. This naturally leads to the question: *How to explicitly design interactive modules according to the characteristic of each class and relationships among certain classes? (e.g. the classifiers of different types of clothes should be allowed to share information via certain communication, while the classifiers for shoes and clothes should be completely disconnected.)* What we hope to see is not a redundantly large model, but a carefully designed one making use of the classification problem itself, the Occam's razor, and the independent mechanism. For instance, the attention mechanism [39] may be a potential solution to select appropriate interactions (and keep them sparse). 3) As discussed earlier, other DNN chain architectures should be considered to (and definitely will) improve the results, so are the training techniques (narrow networks could be more difficult to train up). For example, we tried to retain the batch normalization [8] layers, and in most cases, the chains became much worse on generalization. But with carefully set regularizer and hyperparameters, the BatchNorm layers could (have chances) help improve the quality. Another example is that different loss functions (*e.g.* different λ_i in Eq (1)) could make the training process harder or easier, also directly deciding how the activation value distributes. Also, we use the complete training set to train each chain. Whether the imbalance of training data (one versus all) would affect the training process and classification quality should be noticed. All in all, it's still unclear what layers, operators, optimization techniques, loss functions, and datasets, could benefit the binary classifiers.

5 Conclusions

In this work, we empirically study the learning ability and expressive power of very narrow DNN chains. Based on these narrow DNN chains, we propose the *Ensemble of Narrow DNN Chains* (*ENDC*) framework, which decomposes a multiclassification problem into binary ones, uses narrow DNNs as the base binary classifiers and decides the prediction by voting. By experiments and comparisons to state-of-the-art traditional ML models, we show that ENDC, with the smallest parameter number, could achieve similar accuracy on MNIST and Fashion-MNIST, and significantly better accuracy on CIFAR-10. Furthermore, we discuss the superiority of ENDC over state-of-the-art DNNs and point out future work directions based on sparse architectures as ENDC.

References

[1] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[4] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[8] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[9] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[10] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[11] Misha Denil, Babak Shakibi, Laurent Dinh, Marc'Aurelio Ranzato, and Nando De Freitas. Predicting parameters in deep learning. *arXiv preprint arXiv:1306.0543*, 2013.

[12] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[13] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[14] Giambattista Parascandolo, Niki Kilbertus, Mateo Rojas-Carulla, and Bernhard Schölkopf. Learning independent causal mechanisms. In *International Conference on Machine Learning*, pages 4036–4044. PMLR, 2018.

[15] Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. Recurrent independent mechanisms. *arXiv preprint arXiv:1909.10893*, 2019.

[16] Rangachari Anand, Kishan Mehrotra, Chilukuri K Mohan, and Sanjay Ranka. Efficient classification for multiclass problems using modular neural networks. *IEEE Transactions on Neural Networks*, 6(1):117–124, 1995.

[17] Peter Clark and Robin Boswell. Rule induction with cn2: Some recent improvements. In *European Working Session on Learning*, pages 151–163. Springer, 1991.

[18] Lech Szymanski and Brendan McCane. Deep, super-narrow neural network is a universal classifier. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2012.

[19] Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. The expressive power of neural networks: A view from the width. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6232–6240, 2017.

[20] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[21] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[22] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. *arXiv preprint arXiv:1611.06440*, 2016.

[23] Xiangyu Qi, Tinghao Xie, Ruizhe Pan, Jifeng Zhu, Yong Yang, and Kai Bu. Towards practical deployment-stage backdoor attack on deep neural networks. *arXiv preprint arXiv:2111.12965*, 2021.

[24] Stefan Knerr, Léon Personnaz, and Gérard Dreyfus. Single-layer learning revisited: a stepwise procedure for building and training a neural network. In *Neurocomputing*, pages 41–50. Springer, 1990.

[25] Mikel Galar, Alberto Fernández, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera. An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. *Pattern Recognition*, 44(8):1761–1776, 2011.

[26] Venkataraman Santhanam, Vlad I Morariu, David Harwood, and Larry S Davis. A non-parametric approach to extending generic binary classifiers for multi-classification. *Pattern Recognition*, 58:149–158, 2016.

[27] King-Shy Goh, Edward Chang, and Kwang-Ting Cheng. Svm binary classifier ensembles for image classification. In *Proceedings of the Tenth International Conference on Information and Knowledge Management*, CIKM '01, page 395402, New York, NY, USA, 2001. Association for Computing Machinery.

[28] Il-Seok Oh and Ching Y Suen. A class-modular feedforward neural network for handwriting recognition. *pattern recognition*, 35(1):229–244, 2002.

[29] Jerome H Friedman. Another approach to polychotomous classification. *Technical Report, Statistics Department, Stanford University*, 1996.

[30] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[31] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[32] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[33] zalandoresearch. Fashion-mnist benchmarks. `http://fashion-mnist.s3-website.eu-central-1.amazonaws.com`.

[34] zzzxxxttt. Pytorch simple classification baselines. `https://github.com/zzzxxxttt/pytorch_simple_classification_baselines`.

[35] zalandoresearch. Fashion-mnist. `https://github.com/zalandoresearch/fashion-mnist`.

[36] cdamore. Logistic regression mnist cifar10. `https://github.com/cdamore/logistic_regression_MNIST_CIFAR10`.

[37] mok232. Cifar 10 image classification. `https://github.com/mok232/CIFAR-10-Image-Classification`.

[38] chengyangfu. Pytorch vgg cifar-10. `https://github.com/chengyangfu/pytorch-vgg-cifar10`.

[39] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.