k-Winners-Take-All Ensemble Neural Network

Abien Fred Agarap 🖂 and Arnulfo P. Azcarraga

College of Computer Studies De La Salle University 2401 Taft Ave, Malate, Manila, 1004 Metro Manila, Philippines {abien_agarap, arnulfo.azcarraga}@dlsu.edu.ph https://dlsu.edu.ph

Abstract. Ensembling is one approach that improves the performance of a neural network by combining a number of independent neural networks, usually by either averaging or summing up their individual outputs. We modify this ensembling approach by training the sub-networks concurrently instead of independently. This concurrent training of sub-networks leads them to cooperate with each other, and we refer to them as "cooperative ensemble". Meanwhile, the mixture-of-experts approach improves a neural network performance by dividing up a given dataset to its subnetworks. It then uses a gating network that assigns a specialization to each of its sub-networks called "experts". We improve on these aforementioned ways for combining a group of neural networks by using a k-Winners-Take-All (kWTA) activation function, that acts as the combination method for the outputs of each sub-network in the ensemble. We refer to this proposed model as "kWTA ensemble neural networks" (kWTA-ENN). With the kWTA activation function, the losing neurons of the sub-networks are inhibited while the winning neurons are retained. This results in sub-networks having some form of specialization but also sharing knowledge with one another. We compare our approach with the cooperative ensemble and mixture-of-experts, where we used a feedforward neural network with one hidden layer having 100 neurons as the sub-network architecture. Our approach yields a better performance compared to the baseline models, reaching the following test accuracies on benchmark datasets: 98.34% on MNIST, 88.06% on Fashion-MNIST, 91.56% on KMNIST, and 95.97% on WDBC.

Keywords: Theory and algorithms \cdot competitive learning \cdot ensemble learning \cdot mixture-of-experts \cdot neural network models.

1 Introduction and Related Works

We use artificial neural networks in a myriad of automation tasks such as classification, regression, and translation among others. Neural networks would approach these tasks as a function approximation problem, wherein given a dataset of input-output pairs $D = \{(x_i, y_i) | x_i \in \mathcal{X}, y_i \in \mathcal{Y}\}$, their goal is to learn the mapping $\mathcal{F} : \mathcal{X} \mapsto \mathcal{Y}$. They accomplish this by optimizing their parameters θ with some modification mechanism, such as the retro-propagation of output

errors [14]. We then deem the parameters to be optimal if the neural network outputs are as close as possible to the target outputs in the training data, and if it can adequately generalize on previously unseen data. This can be achieved when a network is neither too simple (has a high bias) nor too complex (has a high variance).

Through the years, combining a group of neural networks is among the simplest and most straightforward ways to achieve this feat. The two basic ways to combine neural networks are by ensembling [1, 3, 4, 15], and by using a mixture-of-experts (MoE) [5, 6]. In an ensemble, a group of independent neural networks is trained to learn the entire dataset. Meanwhile in MoE, each network is trained to learn their own and different subsets of the dataset.

In this work, we use a group of neural networks for a classification task on the following benchmark datasets: MNIST [8], Fashion-MNIST [18], Kuzushiji-MNIST (KMNIST) [2], and Wisconsin Diagnostic Breast Cancer (WDBC) [17]. We introduce a variant of ensemble neural networks that uses a k-Winners-Take-All (kWTA) activation function to combine the outputs of its sub-networks instead of using averaging, summation, or voting schemes to combine such outputs. We then compare our approach with an MoE and a modified ensemble network on a classification task on the aforementioned datasets.

1.1 Ensemble of Independent Networks

We usually form an ensemble of networks by independently or sequentially (in the case of boosting) training them, and then by combining their outputs at test time usually by averaging [1] or voting [4]. In this work, we opted to use the averaging scheme for ensembling.

That is, we have a group of neural networks f_1, \ldots, f_M parameterized by $\theta_1, \ldots, \theta_M$, and we compute its final output as,

$$o = \frac{1}{M} \sum_{m=1}^{M} f_m(x; \theta_m) \tag{1}$$

Each sub-network is trained independently to minimize their own loss function, e.g. cross entropy loss for classification, $\ell_{ce}(y, o) = -\sum y \log(o)$. Then Eq. 1 is used to get the model outputs at test time.

1.2 Mixture of Experts

The Mixture-of-Experts (MoE) model consists of a set of M "expert" neural networks E_1, \ldots, E_m and a "gating" neural network G [5]. The experts are assigned by the gating network to handle their own subset of the entire dataset. We compute the final output of this model using the following equation,

$$o = \sum \arg \max G(x) E_m(x) \tag{2}$$

where G(x) is gating probability output to choose E_m for a given input x. The gating network and the expert networks have their respective set of parameters. Then, we compute the MoE model loss by using the following equation,

$$\mathcal{L}_{MoE}(x,y) = \frac{1}{M} \sum_{m=1}^{M} \left[\frac{1}{n} \sum_{i=1}^{n} \arg\max G(x_i) \cdot \ell_{ce}(y_i, E_m(x_i)) \right]$$
(3)

where ℓ_{ce} is the cross entropy loss, G(x) is the weighting factor to choose E_m .

In this system, each expert learns to specialize on the cases where they perform well, and they are imposed to ignore the cases on which they do not perform well. With this learning paradigm, the experts become a function of a sub-region of the data space, and thus their set of learned weights highly differ from each other as opposed to traditional ensemble models that result to having almost identical weights for their learners.

1.3 Cooperative Ensemble Learning

We refer to the ensemble learning we described in Section 1.1 as traditional ensemble of independent neural networks. However, in our experiments, we trained the ensemble sub-networks concurrently instead of independently or sequentially. In Algorithm 1, we present our modified version of the traditional ensemble, and we call it "cooperative ensemble" (CE) for the rest of this paper.

```
Algorithm 1: Cooperative Ensemble Learning
              : Dataset D = \{(x_i, y_i) | x_i \in \mathbb{R}^d, y_i = 1, \dots, k\}, randomly
    Input
                 initialized networks f_1, \ldots, f_M parameterized by \theta_1, \ldots, \theta_M
    Output: Ensemble of M trained networks f_1, \ldots, f_M
 1 Initialization;
 2 Sample mini-batch B \subset D;
 3
    for t \leftarrow 0 to convergence do
         for m \leftarrow 1 to M do
 4
             \# Forward pass: Compute model outputs for mini-batch
 5
            \hat{y}_{m,1},\ldots,\hat{y}_{m,B}=f_m(x_B)
 6
        end
 7
        o = \frac{1}{M} \sum_{m}^{M} \hat{y}_{m}
# Backward pass: Update the models
 8
 9
        \theta_m^* = \theta_m - \alpha \nabla \ell(y, o)
\mathbf{10}
11 end
```

First, in a training loop, we compute each sub-network output $\hat{y}_{m,B}$ for minibatches of data B (line 6). Then, similar to a traditional ensemble, we compute the output of this model o by averaging over the individual network outputs (line 8). Finally, we optimize the parameters of each sub-network in the ensemble based on the gradients of the loss between the ensemble network outputs o and the target labels y (line 10).

In contrast, a traditional ensemble of independent networks train each subnetwork independently before ensembling, thus not allowing an interaction among

the members of the ensemble and not allowing a chance for each member to contribute to the knowledge of one another.

Cooperative ensemble may have already been used in practice in the real world, but we take note of this variant for it presents itself as a more competitive baseline for our experimental model. This is because cooperative ensemble introduces some form of interaction among the sub-networks during training since there is an information feedback from the combination stage to the sub-network weights, thus giving each sub-network a chance to share their knowledge with one another [9].

The contributions of this study are as follows,

- 1. The conceptual introduction of cooperative ensembling as a modification to the traditional ensemble of independent networks. The cooperative ensemble is a competitive baseline model for our experimental model (see Section 3).
- 2. We introduce an ensemble network that uses a kWTA activation function to combine its sub-network outputs (Section 2). Our approach presents better classification performance on the MNIST, Fashion-MNIST, KMNIST, and Wisconsin Diagnostic Breast Cancer (WDBC) datasets (see Section 3).

2 Competitive Ensemble Learning

We take the cooperative ensembling approach further by introducing a competitive layer as a way to combine the outputs of the sub-networks in the ensemble.

We propose to use a k-Winners-Take-All (kWTA) activation function for a fully connected layer which combines the sub-network outputs in the ensemble, and we call the resulting model "kWTA ensemble neural network" (kWTA-ENN). As per Majani et al. (1989) [10], the kWTA activation function admits $k \geq 1$ winners in a competition among neurons in a hidden layer of a neural network (see Eq. 4 for the kWTA activation function).

$$\phi_k(z)_j = \begin{cases} z_j & z_j \in \{\max_k z\}\\ 0 & z_j \notin \{\max_k z\} \end{cases}$$

$$\tag{4}$$

where z is an activation output, and k is the percentage of winning neurons we want to get. We set k = 0.75 in all our experiments, but it could still be optimized as it is a hyper-parameter. This kWTA activation function that we used is the classical one [10] as we are only inhibiting the losing neurons in the competition while retaining the values of the winning neurons. Due to competition, the winning neurons gain the right to respond to particular subsets of the input data, as per Rumelhart & Zipser (1985) [13].

We have seen the training algorithm for our cooperative ensemble in Algorithm 1, wherein we train the sub-networks concurrently instead of independently or sequentially. We incorporate the same manner of training in kWTA-ENN, and we lay down our proposed training algorithm in Algorithm 2.

Our model first computes the sub-network outputs $f_m(x_B)$ for each minibatch of data B (line 6) but as opposed to cooperative ensemble, we do not

5

Algorithm 2: k-Winners-Take-All Ensemble Network

: Dataset $D = \{(x_i, y_i) | x_i \in \mathbb{R}^d, y_i = 1, \dots, k\}$, randomly initialized Input networks f_1, \ldots, f_M parameterized by $\theta_1, \ldots, \theta_M$ **Output**: Ensemble of M trained networks f_1, \ldots, f_M 1 Initialization; **2** Sample mini-batch $B \subset D$; **3** for $t \leftarrow 0$ to convergence do for $m \leftarrow 1$ to M do 4 # Forward pass: Compute model outputs for mini-batch 5 6 $\hat{y}_{m,1},\ldots,\hat{y}_{m,B}=f_m(x_B)$ 7 end $\hat{Y} = \hat{y}_{1,B}, \dots, \hat{y}_{M,B}$ 8 9 $z = \theta_z \hat{Y} + b_z$ $o = \phi_k(z)$ 10 # Backward pass: Update the models 11 $\mathbf{12}$ $\theta_m^* = \theta_m - \alpha \nabla \ell(y, o)$ 13 end

use a simple averaging of the sub-network outputs. Instead, we concatenate the sub-network outputs \hat{Y} (line 8) and use it as an input to a fully connected layer (line 9). We then pass the fully connected layer output z to the kWTA activation function (line 10). Finally, we update our ensemble based on the gradients of the loss between the kWTA-ENN outputs o and the target labels y (line 12).

To further probe the effect of the kWTA activation function in the combination of sub-network outputs, we add a competition delay parameter d. We define this delay parameter as the number of initial training epochs where the kWTA activation function is not yet used on the fully connected layer output that combines the sub-network outputs. We set d = 0; 3; 5; 7.

3 Experiments

To demonstrate the performance gains using our approach, we used four benchmark datasets for evaluation: MNIST [8], Fashion-MNIST [18], KMNIST [2], and WDBC [17]. We ran each model ten times, and we report the average, best, and standard deviation of test accuracies for each of our model. Then, we ran a Kruskal-Wallis H test on the test accuracy results from ten runs of the baseline and experimental models.

3.1 Datasets Description

We evaluate and compare our baseline and experimental models on three benchmark image datasets and one benchmark diagnostic dataset. We list the dataset statistics in Table 1.

Table 1: Dataset statistics.										
Dataset	# Samples	Input Dimension	# Classes							
MNIST	70,000	784	10							
Fashion-MNIST	70,000	784	10							
KMNIST	70,000	784	10							
WDBC	569	30	2							

Table 1: Dataset statistics

All the *MNIST datasets consist of 60,000 training examples and 10,00 test examples each – all in grayscale with 28×28 resolution. We flattened each image pixel matrix to a 784-dimensional vector.

- MNIST. MNIST is a handwritten digit classification dataset [8].
- Fashion-MNIST. Fashion-MNIST is said to be a more challenging alternative to MNIST that consists of fashion articles from Zalando [18].
- KMNIST. Kuzushiji-MNIST (KMNIST) is another alternative to the MNIST dataset. Each of its classes represent one character representing each of the 10 rows of Hiragana [2].
- WDBC. The WDBC dataset is a binary classification dataset where its 30-dimensional features were computed from a digitized image of a fine needle aspirate of a breast mass [17]. It consists of 569 samples where 212 samples are malignant and 357 samples are benign. We randomly over-sampled the minority class in the dataset to account for its imbalanced class frequency distribution, thus increasing the number of samples to 714. We then splitted this dataset to 70% training set and 30% test set.

We randomly picked 10% of the training samples for each of the dataset to serve as the validation dataset.

3.2 Experimental Setup

The code implementations for both our baseline and experimental models are found in https://gitlab.com/afagarap/kwta-ensemble.

Hardware and Software Configuration We used a laptop computer with an Intel Core i5-6300HQ CPU with Nvidia GTX 960M GPU for training all our models. Then, we used the following arbitrarily chosen 10 random seeds for reproducibility: 42, 1234, 73, 1024, 86400, 31415, 2718, 30, 22, and 17. All our models were implemented in PyTorch 1.8.1 [11] with some additional dependencies listed in the released source code.

Training Details For all our models, we used a feed-forward neural network with one hidden layer having 100 neurons as the sub-network, and then we vary the number of sub-networks per model from 2 to 5. The sub-network weights were initialized with Kaiming uniform initializer [7].

7

Table 2: Classification results on the benchmark datasets (bold values represent the best results) in terms of average, best, and standard deviation of test accuracies (in %). Our kWTA-ENN achieves better test accuracies than our baseline models with statistical significance. * denotes at p < 0.05, ns denotes not significant.

MNIST							Fashion-MNIST										
#	\mathbf{nets}	A	kWTA-ENN			M-E CI	CE		(kWTA-ENN			M-E	CE		
		Acc	d = 0	d = 3	d = 5	d = 7	MOE	CE	# n	nets	Acc	d = 0	d = 3	d = 5	d = 7	NIOE	CE
	2	AVG	98.16	98.18	98.18	98.18	96.43	97.90		2	AVG	87.53	87.54	87.54	87.54	86.59	87.84
		MAX	98.28	98.28	98.28	98.28	96.66	97.96			MAX	87.78	87.70	87.70	87.70	87.54	88.00
		STD	0.08	0.08	0.08	0.08	0.25	0.05			STD	0.16	0.12	0.12	0.12	0.40	0.11
		* $H = 41.51, \ p = 7.39 \times 10^{-8}$								* $H = 36.75, \ p = 6.72 \times 10^{-7}$							
	3	AVG 98.24 98.26 98.26 98.26 94.67 97.62								AVG	87.73	87.81	87.81	87.81	85.54	87.69	
		MAX	98.36	98.39	98.39	98.39	96.33	97.71		3	MAX	88.01	88.10	88.10	88.10	87.15	87.86
		STD	0.06	0.08	0.08	0.08	0.99	0.05			STD	0.18	0.15	0.15	0.15	0.58	0.09
		* $H = 41.19, \ p = 8.61 \times 10^{-8}$									* $H = 28.32, \ p = 3.16 \times 10^{-5}$						
	4	AVG	98.30	98.27	98.27	98.27	92.349	97.33		4	AVG	87.88	87.93	87.93	87.93	84.47	87.40
		MAX	98.43	98.39	98.39	98.39	95.02	97.39			MAX	88.22	88.15	88.15	88.15	86.69	87.54
		STD	0.07	0.08	0.08	0.08	1.30	0.05			STD	0.14	0.15	0.15	0.15	1.20	0.09
			* H	= 41.60	p = 7	7.11×1	0^{-8}					* $H = 42.04, \ p = 5.78 \times 10^{-8}$					
	5	AVG	98.33	98.34	98.34	98.34	90.63	97.02			AVG	87.99	88.06	88.06	88.06	82.89	87.15
		MAX	98.52	98.42	98.42	98.42	91.94	97.13		5	MAX	88.22	88.27	88.27	88.27	85.80	87.27
		STD	0.08	0.05	0.05	0.05	1.25	0.06		0	STD	0.15	0.15	0.15	0.15	2.18	0.05
		* $H = 41.58, \ p = 7.17 \times 10^{-8}$									* $H = 42.26, \ p = 5.22 \times 10^{-8}$						
KMNIST																	
				KMN	IST								WD	BC			
#	nets	Acc		KMN kWT	IST A-ENN	ſ	MoE	CE		note	1.00		WD kWTA	BC -ENN		MoF	CF
#	nets	Acc	d = 0	$\begin{array}{c} \text{KMN} \\ \text{kWTA} \\ \text{d} = 3 \end{array}$	$\begin{array}{l} \text{IST} \\ \text{A-ENN} \\ \text{d} = 5 \end{array}$	d = 7	MoE	CE	#	nets	Acc	d = 0	$\frac{WD}{kWTA}$ $d = 3$	BC -ENN d = 5	d = 7	MoE	CE
#	nets	Acc AVG	d = 0 90.64	KMN kWTA d = 3 90.53	IST -ENN d = 5 90.53	d = 7 90.53	MoE 85.23	CE 89.94	#	nets	Acc AVG	d = 0 95.43	WD kWTA d = 3 95.36	BC -ENN d = 5 95.36	d = 7 95.36	MoE 94.49	CE 95.79
#	nets	Acc AVG MAX	d = 0 90.64 91.11	KMN kWT d = 3 90.53 90.74	IST d = 5 90.53 90.74	d = 7 90.53 90.74	- MoE 85.23 87.14	CE 89.94 90.19	#	nets	Acc AVG MAX	d = 0 95.43 98.62	WD kWTA d = 3 95.36 98.62	BC = 5 = 5 = 95.36 = 98.62	d = 7 95.36 98.62	MoE 94.49 98.57	CE 95.79 99.05
#	nets 2	Acc AVG MAX STD	d = 0 90.64 91.11 0.29	KMN kWTA d = 3 90.53 90.74 0.12	IST -ENN d = 5 90.53 90.74 0.12	d = 7 90.53 90.74 0.12	MoE 85.23 87.14 0.99	CE 89.94 90.19 0.12	#	nets 2	Acc AVG MAX STD	d = 0 95.43 98.62 1.98	WD kWTA d = 3 95.36 98.62 2.48	BC -ENN $d = 5 95.36 98.62 2.48 $	d = 7 95.36 98.62 2.48	MoE 94.49 98.57 2.37	CE 95.79 99.05 2.13
#	nets 2	Acc AVG MAX STD	d = 0 90.64 91.11 0.29 * H		IST d = 5 90.53 90.74 0.12 3, $p = 0$	$d = 7$ 90.53 90.74 0.12 3.99×1	MoE 85.23 87.14 0.99 0 ⁻⁸	CE 89.94 90.19 0.12	#	nets 2	Acc AVG MAX STD	d = 0 95.43 98.62 1.98 (ns)	WD = 1.4 $kWTA = 3$ 95.36 98.62 2.48 $H = 1.4$	BC d = 5 95.36 98.62 2.48 0, p =	d = 7 95.36 98.62 2.48 9.24 ×	$\begin{array}{c} \textbf{MoE} \\ 94.49 \\ 98.57 \\ 2.37 \\ 10^{-1} \end{array}$	CE 95.79 99.05 2.13
#	nets 2	Acc AVG MAX STD AVG	d = 0 90.64 91.11 0.29 * <i>H</i> 91.16	KMN kWT d = 3 90.53 90.74 0.12 = 41.63 91.17	IST -ENN d = 5 90.53 90.74 0.12 3, $p = 6$ 91.17	$d = 7$ 90.53 90.74 0.12 3.99×1 91.17	MoE 85.23 87.14 0.99 0 ⁻⁸ 81.12	CE 89.94 90.19 0.12 89.47	#	nets 2	Acc AVG MAX STD AVG	d = 0 95.43 98.62 1.98 (ns) 94.76	WD kWTA d = 3 95.36 98.62 2.48 H = 1.4 95.64	BC -ENN d = 5 95.36 98.62 2.48 10, p = 95.64	d = 7 95.36 98.62 2.48 9.24 × 95.64	MoE 94.49 98.57 2.37 10 ⁻¹ 92.68	CE 95.79 99.05 2.13
#	nets 2	Acc AVG MAX STD AVG MAX	d = 0 90.64 91.11 0.29 * H 91.16 91.4	KMN kWT d = 3 90.53 90.74 0.12 = 41.63 91.17 91.51	IST = 1 b b c b c b c b c c b c c c c c c c c c c	$d = 7$ 90.53 90.74 0.12 3.99×1 91.17 91.51	MoE 85.23 87.14 0.99 0 ⁻⁸ 81.12 87.59	CE 89.94 90.19 0.12 89.47 89.61	#	nets 2	Acc AVG MAX STD AVG MAX	d = 0 95.43 98.62 1.98 (ns) 94.76 98.15 $ $		BC -ENN d = 5 95.36 98.62 2.48 10, p = 95.64 99.07	d = 7 95.36 98.62 2.48 9.24 × 95.64 99.07	$\begin{array}{c} \textbf{MoE} \\ 94.49 \\ 98.57 \\ 2.37 \\ 10^{-1} \\ 92.68 \\ 95.45 \\ \end{array}$	CE 95.79 99.05 2.13 95.35 98.17
#	nets 2 3	Acc AVG MAX STD AVG MAX STD	d = 0 90.64 91.11 0.29 * H 91.16 91.4 0.14 * H	KMN kWTA d = 3 90.53 90.74 0.12 = 41.63 91.17 91.51 0.19	IST d = 5 90.53 90.74 0.12 3, $p = 691.1791.510.19$	$d = 7$ 90.53 90.74 0.12 3.99×1 91.51 0.19	MoE 85.23 87.14 0.99 0 ⁻⁸ 81.12 87.59 2.77	CE 89.94 90.19 0.12 89.47 89.61 0.12	#	nets 2 3	Acc AVG MAX STD AVG MAX STD	d = 0 95.43 98.62 1.98 (ns) 94.76 98.15 1.92		BC = 5 95.36 98.62 2.48 10, $p =$ 95.64 99.07 2.33	d = 7 95.36 98.62 2.48 9.24 × 95.64 99.07 2.33	$\begin{array}{c} \textbf{MoE} \\ 94.49 \\ 98.57 \\ 2.37 \\ 10^{-1} \\ 92.68 \\ 95.45 \\ 2.36 \\ 10^{-1} \end{array}$	CE 95.79 99.05 2.13 95.35 98.17 2.45
#	nets 2 3	Acc AVG MAX STD AVG MAX STD	d = 0 90.64 91.11 0.29 * H 91.16 91.4 0.14 * H		IST d = 5 90.53 90.74 0.12 3, $p = 0$ 91.17 91.51 0.19 9, $p = 9$	d = 7 90.53 90.74 0.12 3.99 × 1 91.51 0.19 9.00 × 1	MoE 85.23 87.14 0.99 0 ⁻⁸ 81.12 87.59 2.77 0 ⁻⁸	CE 89.94 90.19 0.12 89.47 89.61 0.12	#	nets 2 3	Acc AVG MAX STD AVG MAX STD	d = 0 95.43 98.62 1.98 (ns) 94.76 98.15 1.92 (ns)		BC -ENN d = 5 95.36 98.62 2.48 40, p = 95.64 99.07 2.33 20, p =	d = 7 95.36 98.62 2.48 9.24 × 95.64 99.07 2.33 1.02 ×	$\begin{array}{c} \textbf{MoE} \\ 94.49 \\ 98.57 \\ 2.37 \\ 10^{-1} \\ 92.68 \\ 95.45 \\ 2.36 \\ 10^{-1} \end{array}$	CE 95.79 99.05 2.13 95.35 98.17 2.45
#	nets 2 3	Acc AVG MAX STD AVG MAX STD AVG	d = 0 90.64 91.11 0.29 * H 91.16 91.4 0.14 * H 91.39		IST d = 5 90.53 90.74 0.12 3, $p = 0$ 91.17 91.51 0.19 9, $p = 9$ 91.31	d = 7 90.53 90.74 0.12 3.99 × 1 91.51 0.19 0.00 × 1 91.31	MoE 85.23 87.14 0.99 0 ⁻⁸ 81.12 87.59 2.77 0 ⁻⁸ 77.55	CE 89.94 90.19 0.12 89.47 89.61 0.12 88.72 88.72	#	nets 2 3	Acc AVG MAX STD AVG MAX STD AVG	d = 0 95.43 98.62 1.98 (ns) 94.76 98.15 1.92 (ns) 94.98 94.98		BC -ENN d = 5 95.36 98.62 2.48 0, p = 95.64 99.07 2.33 20, p = 95.97 95.97	d = 7 95.36 98.62 2.48 9.24 × 95.64 99.07 2.33 1.02 × 95.97	$\begin{array}{c} \textbf{MoE} \\ 94.49 \\ 98.57 \\ 2.37 \\ 10^{-1} \\ 92.68 \\ 95.45 \\ 2.36 \\ 10^{-1} \\ 91.79 \\ 91.79 \end{array}$	CE 95.79 99.05 2.13 95.35 98.17 2.45 95.65
#	nets 2 3 4	Acc AVG MAX STD AVG MAX STD AVG MAX	d = 0 90.64 91.11 0.29 * H 91.16 91.4 0.14 * H 91.39 91.68	KMN kWTA d = 3 90.53 90.74 0.12 = 41.63 91.17 91.51 0.19 = 41.09 91.31 91.54	IST d -ENN d = 5 90.53 90.74 0.12 3, $p = 0$ 91.17 91.51 0.19 9, $p = 9$ 91.31 91.54	$\begin{array}{c} \mathbf{d} = 7 \\ 90.53 \\ 90.74 \\ 0.12 \\ 3.99 \times 1 \\ 91.51 \\ 0.19 \\ 90.00 \times 1 \\ 91.31 \\ 91.31 \\ 91.54 \end{array}$	MoE 85.23 87.14 0.99 0 ⁻⁸ 81.12 87.59 2.77 0 ⁻⁸ 77.55 83.04	CE 89.94 90.19 0.12 89.47 89.61 0.12 88.72 88.72 88.94 0.12	#	nets 2 3 4	Acc AVG MAX STD AVG MAX STD AVG MAX	d = 0 95.43 98.62 1.98 (ns) 94.76 98.15 1.92 (ns) 94.98 98.98 98.62		BC -ENN d = 5 95.36 98.62 2.48 0, p = 95.64 99.07 2.33 20, p = 95.97 98.62 2.95.97	d = 7 95.36 98.62 2.48 9.24 × 95.64 99.07 2.33 1.02 × 95.97 98.62	$\begin{array}{c} \textbf{MoE} \\ 94.49 \\ 98.57 \\ 2.37 \\ 10^{-1} \\ 92.68 \\ 95.45 \\ 2.36 \\ 10^{-1} \\ 91.79 \\ 96.67 \\ 10^{-2} \\ 91.79 \\ 96.67 \\ 10^{-2} \\ 10$	CE 95.79 99.05 2.13 95.35 98.17 2.45 95.65 98.15
#	nets 2 3 4	Acc AVG MAX STD AVG MAX STD AVG MAX STD	d = 0 90.64 91.11 0.29 * H 91.16 91.4 0.14 * H 91.39 91.68 0.18		IST d -ENN d = 5 90.53 90.74 0.12 3, $p = 0$ 91.17 91.51 0.19 9, $p = 9$ 91.31 91.54 0.15	$\begin{array}{c} \mathbf{d} = 7 \\ 90.53 \\ 90.74 \\ 0.12 \\ 3.99 \times 1 \\ 91.51 \\ 0.19 \\ 90.00 \times 1 \\ 91.31 \\ 91.54 \\ 0.15 \\ 0.9 \end{array}$	MoE 85.23 87.14 0.99 0 ⁻⁸ 81.12 87.59 2.77 0 ⁻⁸ 77.55 83.04 2.89 0 ⁻⁸	CE 89.94 90.19 0.12 89.47 89.61 0.12 88.72 88.94 0.13	#	nets 2 3 4	Acc AVG MAX STD AVG MAX STD AVG MAX STD	d = 0 95.43 98.62 1.98 (ns) 94.76 98.15 1.92 (ns) 94.98 98.62 2.87 * U		BC -ENN d = 5 95.36 98.62 2.48 40, p = 95.64 99.07 2.33 20, p = 95.97 98.62 2.48 95.62 2.48 2.48 99.07 2.33 20, p = 95.62 2.48 2.48 90.07 2.33 2.33 20, p = 95.62 2.48 95.64 95.64 95.64 95.64 95.64 95.64 95.64 95.64 95.64 95.64 95.64 95.64 95.64 95.64 95.64 95.62 2.33 20, p = 95.97 95.62 2.26 2.26 2.26 2.26 2.26 2.26 2.26 2.26 2.26 2.26 2.26 95.64 95.64 95.64 95.64 95.64 95.64 95.64 95.64 95.64 95.64 95.64 95.64 95.62 2.26	d = 7 95.36 98.62 2.48 9.24 × 95.64 99.07 2.33 1.02 × 95.97 98.62 2.20 78 × 1	$\begin{array}{c} \textbf{MoE} \\ 94.49 \\ 98.57 \\ 2.37 \\ 10^{-1} \\ 92.68 \\ 95.45 \\ 2.36 \\ 10^{-1} \\ 91.79 \\ 96.67 \\ 4.20 \\ 0^{-2} \end{array}$	CE 95.79 99.05 2.13 95.35 98.17 2.45 95.65 98.15 2.00
#	nets 2 3 4	Acc AVG MAX STD AVG MAX STD AVG MAX STD	d = 0 90.64 91.11 0.29 * H 91.16 91.4 0.14 * H 91.39 91.68 0.18 * H	KMN kWT4 $d = 3$ 90.53 90.74 0.12 = 41.63 91.17 91.51 0.19 = 41.09 91.31 91.54 0.15 = 41.67	IST A -ENN $d = 5$ 90.53 90.74 0.12 $3, p = 0$ 91.17 91.51 0.19 $9, p = 9$ 91.31 91.54 0.15 $7, p = 0$ 0.15	$\begin{array}{c} \mathbf{d} = 7 \\ 90.53 \\ 90.74 \\ 0.12 \\ 3.99 \times 1 \\ 91.51 \\ 0.19 \\ 9.00 \times 1 \\ 91.31 \\ 91.54 \\ 0.15 \\ 3.88 \times 1 \\ 0.15 \\ \end{array}$	MoE 85.23 87.14 0.99 0 ⁻⁸ 81.12 87.59 2.77 0 ⁻⁸ 77.55 83.04 2.89 0 ⁻⁸	CE 89.94 90.19 0.12 89.47 89.61 0.12 88.72 88.94 0.13 0.13	#	nets 2 3 4	Acc AVG MAX STD AVG MAX STD AVG MAX STD	d = 0 95.43 98.62 1.98 (ns) 94.76 98.15 1.92 (ns) 94.98 98.62 2.87 * H 05.02	WD] kWTA d = 3 95.36 98.62 2.48 H = 1.4 99.07 2.33 H = 9.2 95.97 98.62 2.20 = 12.50 05.40 9.07	BC d = 5 95.36 98.62 2.48 0, p = 95.64 99.07 2.33 20, p = 95.97 98.62 2.20 3, p = 2 2.20	d = 7 95.36 98.62 2.48 9.24 × 95.64 99.07 2.33 1.02 × 95.97 98.62 2.20 2.78 × 1 05.4 95.97 95.97 98.62 95.97 95.97 95.97 95.97 95.97 95.97 95.97 95.97 95.97 95	$\begin{array}{c} \textbf{MoE} \\ 94.49 \\ 98.57 \\ 2.37 \\ 10^{-1} \\ 92.68 \\ 95.45 \\ 2.36 \\ 10^{-1} \\ 91.79 \\ 96.67 \\ 4.20 \\ 0^{-2} \\ 0^{-2} \end{array}$	CE 95.79 99.05 2.13 95.35 98.17 2.45 95.65 98.15 2.00
#	nets 2 3 4	Acc AVG MAX STD AVG MAX STD AVG MAX STD AVG	d = 0 90.64 91.11 0.29 * H 91.16 91.4 0.14 91.39 91.68 0.18 * H 91.30 0.18	KMN kWT4 $d = 3$ 90.53 90.74 0.12 = 41.63 91.17 91.51 0.19 = 41.09 91.31 91.54 0.15 = 41.67	IST A -ENN $d = 5$ 90.53 90.74 0.12 $3, p = 6$ 91.17 91.51 0.19 $9, p = 9$ 91.31 91.54 0.15 $7, p = 6$ 91.52	$\begin{array}{c} \mathbf{d} = 7 \\ 90.53 \\ 90.74 \\ 0.12 \\ 3.99 \times 1 \\ 91.51 \\ 0.19 \\ 91.51 \\ 0.19 \\ 91.51 \\ 0.15 \\ \mathbf{5.88 \times 1} \\ 91.54 \\ 0.15 \\ \mathbf{5.88 \times 1} \\ 91.52 \\ 91.52 \end{array}$	MoE 85.23 87.14 0.99 0 ⁻⁸ 81.12 87.59 2.77 0 ⁻⁸ 77.55 83.04 2.89 0 ⁻⁸ 74.17 70.25 74.17	CE 89.94 90.19 0.12 89.47 89.61 0.12 88.72 88.94 0.13 87.87 89.60	#	nets 2 3 4	Acc AVG MAX STD AVG MAX STD AVG MAX STD AVG AVG	d = 0 95.43 98.62 1.98 (ns) 94.76 98.15 1.92 (ns) 94.98 98.62 2.87 * H 95.03 08.61		BC d = 5 95.36 98.62 2.48 0, p = 95.64 99.07 2.33 20, p = 95.97 98.62 2.20 3, p = 2 95.40 90.07	d = 7 95.36 98.62 2.48 9.24 × 95.64 99.07 2.33 1.02 × 95.97 98.62 2.20 2.78 × 1 95.40 90.05	MoE 94.49 98.57 2.37 10^{-1} 92.68 95.45 2.36 10^{-1} 91.79 96.67 4.20 0^{-2} 90.93 96.92	CE 99.05 2.13 95.35 98.17 2.45 95.65 98.15 2.00 95.04
#	nets 2 3 4 5	Acc AVG MAX STD AVG MAX STD AVG MAX STD AVG MAX	d = 0 90.64 91.11 0.29 * H 91.16 91.4 * H 91.39 91.68 0.18 * H 91.56 91.82 0.12	KMN kWTA $d = 3$ 90.53 90.74 0.12 = 41.63 91.17 91.51 0.19 = 41.03 91.31 91.54 0.15 = 41.67 91.32 91.34 0.15 = 41.67	IST $d = 5$ 90.53 90.74 0.12 3, $p = 6$ 91.17 91.51 0.19 9, $p = 9$ 91.31 91.54 0.15 7, $p = 6$ 91.52 91.76	$\begin{array}{c} \mathbf{d} = 7 \\ 90.53 \\ 90.74 \\ 0.12 \\ 3.99 \times 1 \\ 91.17 \\ 91.51 \\ 0.19 \\ 90.00 \times 1 \\ 91.31 \\ 91.54 \\ 0.15 \\ 5.88 \times 1 \\ 91.52 \\ 91.52 \\ 91.76 \\ 0.16 \\ \end{array}$	MoE 85.23 87.14 0.99 0 ⁻⁸ 81.12 87.59 2.77 0 ⁻⁸ 77.55 83.04 2.89 0 ⁻⁸ 74.17 79.99 2.47	CE 89.94 90.19 0.12 89.47 89.61 0.12 88.72 88.94 0.13 87.87 88.02	#	nets 2 3 4 5	Acc AVG MAX STD AVG MAX STD AVG MAX STD AVG MAX STD	d = 0 95.43 98.62 1.98 (ns) 94.76 98.15 1.92 (ns) 94.98 98.62 2.87 * H 95.03 98.61 9.72		BC -ENN d = 5 95.36 98.62 2.48 0, p = 95.64 99.07 2.33 0, p = 95.97 98.62 2.20 3, p = 2 95.40 99.04 2.33 p = 2 95.40 99.05 2.92	d = 7 95.36 98.62 2.48 9.24 × 95.64 99.07 2.33 95.97 98.62 2.20 2.78 × 1 95.40 99.05 2.22	$\begin{array}{c} \textbf{MoE} \\ 94.49 \\ 98.57 \\ 2.37 \\ 10^{-1} \\ 92.68 \\ 95.45 \\ 2.36 \\ 10^{-1} \\ 91.79 \\ 96.67 \\ 4.20 \\ 0^{-2} \\ 90.93 \\ 96.93 \\ 96.93 \\ 2.60 \end{array}$	CE 95.79 99.05 2.13 95.35 98.17 2.45 95.65 98.15 2.00 95.04 98.61 2.47
#	nets 2 3 4 5	Acc AVG MAX STD AVG MAX STD AVG MAX STD AVG MAX STD	d = 0 90.64 91.11 0.29 * H 91.16 91.4 0.14 * H 91.39 91.63 0.18 * H 91.56 91.82 0.18	$\begin{array}{c} {\bf KMN} \\ {\bf kWTA} \\ {\bf d} = {\bf 3} \\ 90.53 \\ 90.74 \\ 0.12 \\ = 41.63 \\ {\bf 91.17} \\ 91.51 \\ 0.19 \\ = 41.09 \\ 91.31 \\ 91.54 \\ 0.15 \\ = 41.66 \\ 91.52 \\ 91.76 \\ 0.18 \\ \end{array}$	IST $d = 5$ 90.53 90.74 0.12 $3, p = 6$ 91.17 91.51 0.19 $9, p = 9$ 91.31 91.54 0.15 $7, p = 6$ 91.32 91.31 91.54 0.15 0.15 0.152 91.76 0.18	$\begin{array}{c} \mathbf{d} = 7 \\ 90.53 \\ 90.74 \\ 0.12 \\ 5.99 \times 1 \\ 91.51 \\ 0.19 \\ 90.00 \times 1 \\ 91.31 \\ 91.54 \\ 0.15 \\ 5.88 \times 1 \\ 91.52 \\ 91.76 \\ 0.18 \\ 0.18 \\ \end{array}$	MoE 85.23 87.14 0.99 0 ⁻⁸ 81.12 87.59 2.77 0 ⁻⁸ 77.55 83.04 2.89 0 ⁻⁸ 74.17 79.99 3.47 0 ⁻⁸	CE 89.94 90.19 0.12 89.47 89.61 0.12 88.72 88.94 0.13 87.87 88.02 0.09	#	nets 2 3 4 5	Acc AVG MAX STD AVG MAX STD AVG MAX STD AVG MAX STD	$\begin{array}{c} \mathbf{d} = 0 \\ 95.43 \\ 98.62 \\ 1.98 \\ (ns) \\ 94.76 \\ 98.15 \\ 1.92 \\ (ns) \\ 94.98 \\ 98.62 \\ 2.87 \\ * H \\ 95.03 \\ 98.61 \\ 2.73 \\ 98.61 \\ 2.73 \\ * H \end{array}$		BC -ENN d = 5 95.36 98.62 2.48 0, $p =$ 95.64 99.07 2.33 20, $p =$ 95.64 99.07 2.33 2.0, $p =$ 95.62 2.20 3, $p = 2$ 95.40 99.05 2.83 3, $p = 2$	d = 7 95.36 98.62 2.48 9.24 × 95.64 99.07 2.33 1.02 × 95.97 98.62 2.20 2.78 × 1 95.40 99.05 2.83	$\begin{array}{c} \textbf{MoE} \\ 94.49 \\ 98.57 \\ 2.37 \\ 10^{-1} \\ 92.68 \\ 95.45 \\ 2.36 \\ 10^{-1} \\ 91.79 \\ 96.67 \\ 4.20 \\ 0^{-2} \\ 90.93 \\ 96.33 \\ 2.60 \\ 0^{-2} \end{array}$	CE 99.05 2.13 95.35 98.17 2.45 95.65 98.15 2.00 95.04 98.61 2.47

We trained our baseline and experimental models on the MNIST, Fashion-MNIST, and KMNIST datasets using mini-batch stochastic gradient descent (SGD) with momentum [12] of 9×10^{-1} , a learning rate of 1×10^{-1} decaying to 1×10^{-4} , and weight decay of 1×10^{-5} on a batch size of 100 for 10,800 iterations (equivalent to 20 epochs). As for the WDBC dataset, we used the same hyper-parameters except we trained our models for only 249 iterations (equivalent to 20 epochs). All these hyper-parameters were arbitrarily chosen since we did not perform hyper-parameter tuning for any of our models. This makes the comparison fair for our baseline and experimental models, and we also did not have the computational resources to do so, which is why we chose a simple architecture as the sub-network.

We recorded the accuracy and loss during both the training and validation phases. We then used the validation accuracy as the basis to checkpoint the best model parameters θ so far in the training. By the end of each training epoch, we load the best recorded parameters to be used by the model at test time.

3.3 Classification Performance

We evaluate the performance of our proposed approach in its different configurations as per the competition delay parameter d and compare it with our baseline models: Mixture-of-Experts (MoE) and Cooperative Ensemble (CE). The empirical evidence shows our proposed approach outperforms our baseline models on the benchmark datasets we used. However, we are not able to observe a proper trend in performance with respect to the varying values of d, and thus it may warrant further investigation.

For the full classification performance results of our baseline and experimental models, we refer the reader to Table 2, from where we can observe the following:

- 1. MoE performed the least among the models in our experiments, which may be justified with our choice of mini-batch size of 100. MoE performs better on larger datasets and/or larger batch sizes [5, 16].
- 2. CE is indeed a competitive baseline as we can see from the performance margins when compared to our proposed model.
- 3. Our model in its different variations has consistently outperformed our baseline models in terms of average test accuracy (with the exception of two sub-networks for Fashion-MNIST and WDBC).
- 4. Our model has higher margins on its improved test accuracy on the KMNIST dataset, which we find appealing since the said dataset is also supposed to be more difficult than the MNIST dataset and thus it better demonstrates the performance gains using our model.
- 5. Finally, we can observe that there is a statistical significance among the differences in performance of the baseline and experimental models at p < 0.05 (on WDBC, for M = 4, 5 sub-networks), which indicates that the performance gains through our proposed approach are statistically significant.

3.4 Improving cooperation through competitive learning

In the context of our work, we refer to *cooperation* in a group of neural networks as the phenomenon when the members of the group contribute to the overall group performance. For instance, in CE, all the sub-networks contribute to the knowledge of one another as opposed to the traditional ensemble, where there is no interaction among the ensemble members [9]. Meanwhile, *specialization* is when members of a group of neural networks are tasked to a specific subset of the input data, which is the intention behind the design of MoE [5]. In this respect, *competition* can be thought of leading to specialization since it is when the winning units gain the right to respond to a particular subset of the dataset.



Fig. 1: Predictions of each sub-network on a sample MNIST data and their respective final outputs. In 1a, we can infer that MoE sub-networks 2 and 3 are specializing on class 1. In 1b, all CE sub-networks have high probability outputs for class 1. In 1c, all kWTA-ENN sub-networks contributed but with the kWTA activation function, the neurons for other classes were most likely inhibited at inference, thus its higher probability output than MoE and CE.

We argue that with our proposed approach, we employ the notion of all three: competition, specialization, and cooperation.

kWTA-ENN uses a kWTA activation function so that the neurons from its sub-networks could compete for their right to respond to a particular subset of the dataset. We demonstrate this in Figures 1 and 2. Let us recall that kWTA-ENN gets its outputs by computing a linear combination of the outputs of its sub-networks, and then passing the linear combination results to a kWTA activation function. As per the referred figures, even though each kWTA-ENN sub-network is not providing high probability output per class as compared to MoE and CE sub-networks, the final kWTA-ENN output is on par with the MoE and CE probability outputs.

We can then infer two things from this: (1) the kWTA activation function inhibits the neurons of the losing kWTA-ENN sub-networks, and (2) the probability outputs of the winning sub-network neurons enable the sub-networks to help one another. For instance, in Figure 1c, we can observe a probability output for class 1 from sub-network 1, however minimal, and a higher probability output for class 1 from sub-networks 2 and 3, but then their final output has even higher probability output for the same class when compared to MoE and CE



Fig. 2: Predictions of each sub-network on a sample KMNIST data and their respective final outputs. In 2a, we can infer that MoE sub-network 2 is specializing on class 6 ("ma"). In 2b, CE sub-network 3 was assisted by sub-network 2. In 2c, all kWTA-ENN sub-networks contributed but with the kWTA activation function, the neurons for other classes were most likely inhibited at inference, thus its higher probability output than MoE and CE.

probability outputs. The same could be observed in Figure 2c. This is because the losing neurons are inhibited in the competition process while retaining the winner neurons, thus improving the final probability output of the model.

In Table 3, we further support this by showing the per-class accuracy of each kWTA-ENN sub-network with varying number of sub-networks. We can see that there is some apparent division of classes among the sub-networks even without pre-defining such divisions, but the final per-class accuracies of the entire model are even better than the per-class accuracies of the sub-networks, thus suggesting that there is indeed a sharing of responsibility among the sub-networks due to the inhibition of losing sub-network neurons and retention of the winning sub-network neurons, even with the competition in place.

Table 3: Classification results of each kWTA-ENN sub-network and kWTA-ENN itself on MNIST (3a) and KMNIST (3b) datasets. The tables show the test accuracy of each sub-network on each dataset class, indicating a degree of specialization among the sub-networks. Furthermore, the final model accuracy on each class shows that combining the sub-network outputs have stronger predictive capability. These divisions were in no way pre-determined but they show how cooperation by specialization can be done through competitive ensemble.



4 Conclusion and Future Works

We introduce the k-Winners-Take-All ensemble neural network (kWTA-ENN) which uses a kWTA activation function as the means to combine the subnetwork outputs in an ensemble as opposed to the conventional way of combining sub-network outputs through averaging, summation, or voting. Using a kWTA activation function induces competition among the sub-network neurons in an ensemble. This in turn leads to some form of specialization among them, thereby improving the overall performance of the ensemble.

Our comparative results showed that our proposed approach outperforms our baseline models, yielding the following test accuracies on benchmark datasets: 98.34% on MNIST, 88.06% on Fashion-MNIST, 91.56% on KMNIST, and 95.97% on WDBC. We intend to pursue further exploration into this subject by comparing the performance of our baseline and experimental models with respect to varying

mini-batch sizes, by training on other benchmark datasets, and finally, by using a more rigorous statistical treatment for a more formal comparison between our proposed model and our baseline models.

References

- 1. Breiman, Leo. "Stacked regressions." Machine Learning 24.1 (1996): 49-64.
- 2. Clanuwat, Tarin, et al. "Deep Learning for Classical Japanese Literature." arXiv preprint arXiv:1812.01718 (2018).
- Freund, Yoav, and Robert E. Schapire. "Experiments with a New Boosting Algorithm." ICML. Vol. 96. 1996.
- 4. Hansen, Lars Kai, and Peter Salamon. "Neural network ensembles." IEEE Transactions on Pattern Analysis and Machine Intelligence 12.10 (1990): 993-1001.
- Jacobs, Robert A., et al. "Adaptive Mixtures of Local Experts." Neural Computation 3.1 (1991): 79-87.
- Jordan, Michael I., and Robert A. Jacobs. "Hierarchies of adaptive experts." Advances in Neural Information Processing Systems. 1992.
- He, Kaiming, et al. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification." Proceedings of the IEEE international conference on computer vision. 2015.
- 8. LeCun, Yann. "The MNIST database of handwritten digits." http://yann.lecun.com/exdb/mnist/ (1998).
- Liu, Yong, and Xin Yao. "A cooperative ensemble learning system." 1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98CH36227). Vol. 3. IEEE, 1998.
- 10. Majani, E., Ruth Erlanson, and Yaser Abu-Mostafa. "On the K-winners-take-all network." (1989): 634-642.
- Paszke, Adam, et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library." Advances in Neural Information Processing Systems 32 (2019): 8026-8037.
- Qian, Ning. "On the momentum term in gradient descent learning algorithms." Neural Networks 12.1 (1999): 145-151.
- 13. Rumelhart, David E., and David Zipser. "Feature Discovery by Competitive Learning." Cognitive Science 9.1 (1985): 75-112.
- 14. Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors." nature 323.6088 (1986): 533-536.
- Schapire, Robert E. "The strength of weak learnability." Machine learning 5.2 (1990): 197-227.
- Shazeer, Noam, et al. "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer." arXiv preprint arXiv:1701.06538 (2017).
- Wolberg, William H., W. Nick Street, and Olvi L. Mangasarian. "Breast cancer Wisconsin (diagnostic) data set." UCI Machine Learning Repository [http://archive. ics. uci. edu/ml/] (1992).
- Xiao, Han, Kashif Rasul, and Roland Vollgraf. "Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms." arXiv preprint arXiv:1708.07747 (2017).