Mixture of Experts - Experiment 1-7

1. Introduction

   Mixture of experts is a class of model, where output is formed by several experts together. By experts, we refer to machine learning models. The outputs of experts are combined by weighted sum assigned by some gating function. We want to answer the question: (1) Can the jointly learned gate and experts outperform the distinctly learned experts with a pre-designed gate, which we think is ideal. (2) If the mixture of experts outperforms a single expert; (3) How the number of experts effect the performance. To answer the questions, we describe the architecture of mixture of experts we use in section 2, describe the dataset in section 3, describe the experiments in section 4, and show results in section 5.

2. Architecture

   The mixture of experts will be implemented on a classification task, so the output data Y will be a vector containing probabilities for each class. The output of mixture of experts is formed by weighted sum of each experts, where the weights are given by a gating network:

$$Y = \sum_i G(X)_i E_i(X)$$

   Where X is the input, $G(X)_i$ is the weight assigned to the expert i, and $E_i(X)$ is the output of expert i, which has same dimension as output Y.

   We will have different selections of G and E, depending on what is the aim of each experiment. We describe those selections in following subsections.

   2.1 Experts

   There are 2 choices for experts: the linear model with softmax output, and the convolutional neural network. We call them linear expert and convolutional expert.

   The linear expert is represented by the following equation:

$$E_i(X) = softmax(W_i X)$$

   Where W_i is the weight matrix to be learned. The matrix multiplication $W_i X$ represents a linearly computed score vector with the number of classes as its size. The score vector is then transformed into probabilities by the softmax-function. The softmax-function is defined as following:

$$softmax(\boldsymbol{a})_c = \frac{\exp(a_c)}{\sum_d \exp(a_d)}$$

   The convolutional expert is defined by following code:

```
layer = X
layer = tf.layers.conv2d(layer,64,(3,3), padding='same', activation=tf.nn.relu)
layer = tf.layers.conv2d(layer,64,(3,3), padding='same', activation=tf.nn.relu)
layer = tf.layers.flatten(layer)
```

```
layer = tf.layers.dense(layer, 64, tf.nn.relu)
Y = tf.layers.dense(layer, num_classes, tf.nn.softmax)
```

2.2 Gating networks

We experiment on 4 choices of gating: no-gate, pre-designed gate, linear gate, and 1-hidden-layer gate.

With no-gate, we set the number of experts to be 1. So, $G(X)_1$ = 1 for the only expert 1 and $Y = E_1(X)$, which is the same as a model without the mixture of experts.

The pre-designed gates output a one-hot vector for each input. The elements of the one-hot are zeros, except being one on a specific index. The specific index is given explicitly and is equal to the index of the super-label of the input. Because the number of super-labels of a given dataset is fixed, the pre-designed gate has always a fixed number of experts on that dataset. The dataset and super-labels will be covered in next section.

The linear gate is the softmax output of a linear model with the dimension of the number of experts. The linear gate is represented by following equation:

$$G(X) = softmax(W_G X)$$

where W_G is the weight matrix to be learned via the back-propagation algorithm.

The 1-hidden-layer gate looks like this:

```
layer = tf.layers.flatten(X)
layer = tf.layers.dense(layer,64,tf.nn.relu)
Gate = tf.layers.dense(layer, num_experts, tf.nn.softmax)
```

3. Dataset

Depending on experiment, we will use 2 datasets: mixture of datasets and cifar-100.

The mixture of datasets is a combination of mnist and cifar-10. The mnist are images of handwritten digits, with 60000 training images of size (28,28,1) and 10000 test images. The cifar-10 are images of objects. It is more complex to classify than mnist. It contains 50000 training images of size (32,32,3) and 10000 test images. Both mnist and cifar-10 have 10 classes, so the combined dataset will have 20 classes in total, where the first 10 classes are from mnist and the second 10 classes are from cifar-10. The number of mnist images of different classes are different, and all cifar-10 classes have same number of images. All Images from both datasets are resized to 32x32 and then gray-scaled to a single color-channel, so the mixture of datasets have 110000 training images of size (32, 32, 1) and 20000 test images.

The cifar-100 is a dataset containing images of objects. The sizes of images are (32,32,3), and there are 50000 training and 10000 testing images. There are 100 classes with same number of images. The 100 classes are grouped into 20 super-classes, each contains 5 basic classes.

3.1 super-labels and pre-designed gate

To use the pre-designed gate, we must define the super-labels of each image in datasets.

For the mixture of datasets, we let the super-label of an image be the source dataset. Then the predesigned gate is $G(X) = [1,0]\ if\ X\ is\ MNIST\ else\ [0,1]$, and it always have 2 experts, each expert is has the classification task on the source dataset.
The cifar-100 dataset has already 20 super-classes. So, the pre-designed gate on cifar-100 will have always 20 experts, where each expert has a 5-class classification task.

The labels of cifar-100 are showed in Figure 1 and super-labels are in Figure 2. They are not grouped by indices in original data.

Index	Label	Index	Label	Index	Label	Index	Label	Index	Label
0	apple	20	chair	40	Lamp	60	plain	80	squirrel
1	aquarium_fish	21	chimpanzee	41	lawn_mower	61	plate	81	streetcar
2	baby	22	clock	42	Leopard	62	poppy	82	sunflower
3	bear	23	cloud	43	Lion	63	porcupine	83	sweet_pepper
4	beaver	24	cockroach	44	Lizard	64	possum	84	table
5	bed	25	couch	45	Lobster	65	rabbit	85	tank
6	bee	26	crab	46	Man	66	raccoon	86	telephone
7	beetle	27	crocodile	47	maple_tree	67	ray	87	television
8	bicycle	28	cup	48	Motorcycle	68	road	88	tiger
9	bottle	29	dinosaur	49	Mountain	69	rocket	89	tractor
10	bowl	30	dolphin	50	Mouse	70	rose	90	train
11	boy	31	elephant	51	Mushroom	71	sea	91	trout
12	bridge	32	flatfish	52	oak_tree	72	seal	92	tulip
13	bus	33	forest	53	Orange	73	shark	93	turtle
14	butterfly	34	fox	54	Orchid	74	shrew	94	wardrobe
15	camel	35	girl	55	Otter	75	skunk	95	whale
16	can	36	hamster	56	palm_tree	76	skyscraper	96	willow_tree
17	castle	37	house	57	Pear	77	snail	97	wolf
18	caterpillar	38	kangaroo	58	pickup_truck	78	snake	98	woman
19	cattle	39	keyboard	59	pine_tree	79	spider	99	worm

Figure 1. Labels of cifar-100

Index	Super-Label	Basic Labels				
0	aquatic_mammals	4	30	55	72	95
1	fish	1	32	67	73	91
2	flowers	54	62	70	82	92
3	food_containers	9	10	16	28	61
4	fruit_and_vegetables	0	51	53	57	83
5	household_electrical_devices	22	39	40	86	87
6	household_furniture	5	20	25	84	94
7	insects	6	7	14	18	24
8	large_carnivores	3	42	43	88	97
9	large_man-made_outdoor_things	12	17	37	68	76
10	large_natural_outdoor_scenes	23	33	49	60	71
11	large_omnivores_and_herbivores	15	19	21	31	38
12	medium_mammals	34	63	64	66	75
13	non-insect_invertebrates	26	45	77	79	99
14	people	2	11	35	46	98
15	reptiles	27	29	44	78	93
16	small_mammals	36	50	65	74	80
17	trees	47	52	56	59	96
18	vehicles_1	8	13	48	58	90
19	vehicles_2	41	69	81	85	89

Figure 2. Super labels of cifar-100

4. Experiments

We have 5 experiments using different dataset, experts and gating networks. We plot different things depend on the setting.

4.1 Experiment 1 – Mixture datasets, linear experts, different gating functions

Here, we use the mixture of datasets. The experts are selected to be linear. We iterate through all gating choices – no-gating, pre-designed gate, linear gate and 1-hidden-layer gate. set the number of experts to 2 for all except no-gating.

By this experiment, we would see the basic performance of the mixture of experts in a single setting. To see the performance, we plot the epoch-accuracy curves of different gating choices on both train and test sets. We expect that the pre-designed gate gives the upper-bound, because it already separates the classes from mnist and cifar-10 and experts are learned to be specialized on those datasets. We also expect that no-gating gives the lower-bound, since it is a simpler model.

We also define and plot the activation and the confusion measures of the experts.

The activation of expert i on a set D is the average of the gate of expert i on all elements of D, or equally:

$$Activation(i|D) = \frac{1}{|D|}\sum_{x\in D} G(x)_i$$

We plot the activation of experts on super-labels (mnist and cifar-10 datasets) as matrices, and we also plot the activation on every basic labels graphically.

The activation tells about how the gating network is directing inputs to each experts, and we could see whether experts are specialized.

Then we define the confusion. The confusion on a set D of the label pair i, j is the number of true label i predicted to be label j in the set D, or equally:

$$Confusion(i, j|D) = \sum_{x\in D} \mathbf{1}(label(x) = i)\mathbf{1}\big(argmax\big(y(x)\big) = j\big)$$

Where $\mathbf{1}$ is 1 if the input value is true and 0 otherwise. The term $argmax(y)$ is the predicted label given that y is the output probability distribution of the model.

The expert confusion is then defined by the confusion on the expert responsible set. The responsible set of expert e is the subset of all data, that the gate value of expert e is the largest, or equally:

$$R_e = \{x|argmax\big(G(x)\big) = e\ \}$$

And:

$$Confusion_e(i,j) = Confusion(i,j|R_e) = \sum_{x \in R_e} \mathbf{1}(label(x) = i)\mathbf{1}\big(argmax\big(y(x)\big) = j\big)$$

The confusion as matrices tells whether the model/the expert is likely to predict to some labels.

4.2 Experiment 2 – Mixture datasets, convolutional experts, different gating functions

The only change from experiment 1 is that the experts are selected to be convolutional.

We expect the accuracy of all gates to be high. We also plot the activation and confusion to see how the gates and the experts behave.

4.3 Experiment 3 – Mixture datasets, linear experts, different number of experts

The scope of this experiment is to show how does the model and gating behave on different number of experts.

We do not keep the convolutional experts from experiment 2 and change them back to linear experts. We select the gating to be 1-hidden-layer.

Instead of epoch-accuracy curves, we plot the number-of-experts-accuracy curves.

We plot the activation matrices and the confusion matrices of one case.

4.4 Experiment 4 – Cifar-100 dataset, linear experts, different number of experts

The only change from experiment 3 is here we have cifar-100 dataset.

Cifar-100 dataset is more complex since it has 100 classes. We experiment linear experts on this dataset and see if mixture of linear experts improve performance.

Additionally we experiment on 20-experts pre-designed gate using Cifar-100 20 super-labels and compare their performance.

4.5 Experiment 5 – Cifar-100 dataset, convolutional experts, different number of experts

The only change from experiment 4 is here we have convolutional experts.

We also experiment to see if mixture of experts improve performance and we also compare them with pre-designed gate with convolutional experts.

4.6 Experiment 6 – Mixture dataset, convolutional experts, different number of experts

The mixtures of experts in experiment 3-5 have different behaviours, 3. has increasing accuracy when number of experts increase, 4. has decreasing and 5. has stable accuracy.

We experiment using convolutional experts on mixture dataset and see if which case does it has.

4.7 Experiment 7 – Mixture dataset, heterogenouos experts, different gating functions.

The experts are sometime specialized or sometime ignored. Here we try to experiment if the experts of different complexity can be specialized on different subset of data with different complexity. We let expert 1 to be linear and expert 2 to be convolutional.

5. Results and analysis

5.1 Result 1 – Mixture dataset, linear Experts, different gating functions

The Figure 3 shows the epoch-accuracy plot of experiment 1. The jointly learned gate networks outperformed both single expert and pre-designed gate when the experts are linear.

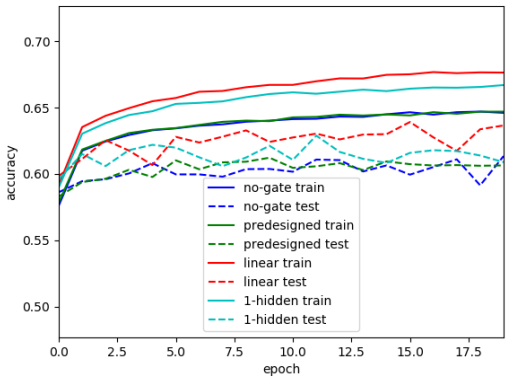

Figure 3. The accuracy curves of different models with linear experts on mixture dataset. Blue is the single expert, green is the pre-designed gate, red is the linear gate, cyan is 1-hidden-layer gate. Solid line is on train data, dashed line on test data.

The activation matrices of super-labels (Figure 4) shows that the frequency of using experts are unbalanced. In the case of 1-hidden-layer gate, expert 2 predicts on only part of mnist, whereas expert 1 predicts on whole cifar-10 and part of mnist.

Model		Expert 1	Expert 2
Pre-designed gate	Mnist	1.00	0.00
	Cifar	0.00	1.00
Linear gate	Mnist	0.81	0.19
	Cifar	0.39	0.61
1-hidden layer gate	Mnist	0.34	0.66
	Cifar	1.00	0.00

Figure 4 The activation matrices of super-labels on the test set after training.

To see how 1-hidden-layer gate behave, we plot the activation per class in Figure 5 and confusions of 1-hidden-layer gate in Figure 6,**Error! Reference source not found.**. The expert 1 and expert 2 on mnist are specialized to predict classes {0,2,3,4,8,9} and {5,6,7}. Both experts could predict class 1. Expert 2 is confused on cifar-10 while expert 1 is not predicting cifar-10 images.

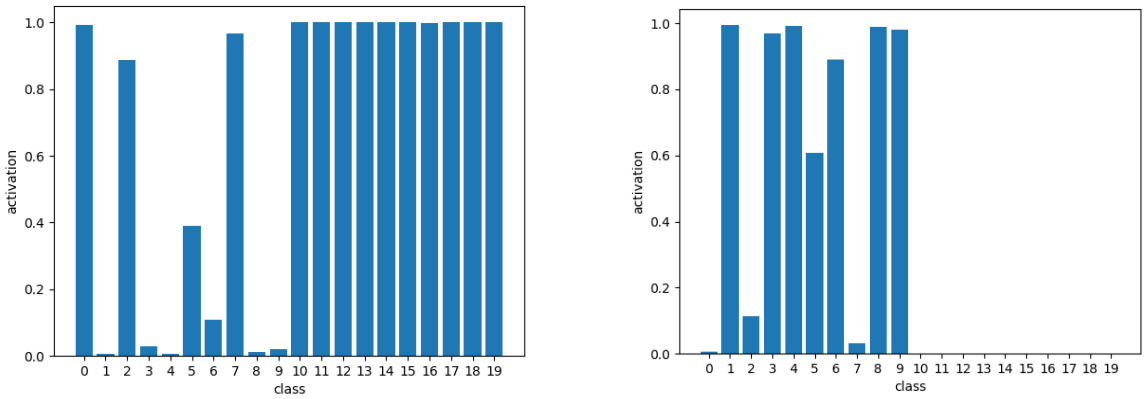

Figure 5. Activation of 1-hidden-layer gating per class. Left and right figures are of expert 1 and 2 correspondingly. Classes 0-9 are mnist labels and 10-19 are cifar-10 labels.

OVERALL CONFUS.		PREDICTION																				
	class	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	total
	0	965	0	1	1	0	3	4	2	2	2	0	0	0	0	0	0	0	0	0	0	**980**
	1	0	1117	1	3	0	0	4	1	9	0	0	0	0	0	0	0	0	0	0	0	**1135**
	2	14	1	977	10	4	2	4	10	8	2	0	0	0	0	0	0	0	0	0	0	**1032**
	3	0	1	6	949	2	11	1	5	26	9	0	0	0	0	0	0	0	0	0	0	**1010**
	4	0	1	11	0	931	1	9	2	2	25	0	0	0	0	0	0	0	0	0	0	**982**
	5	2	0	0	25	2	831	10	4	12	6	0	0	0	0	0	0	0	0	0	0	**892**
	6	6	3	0	0	7	16	918	0	8	0	0	0	0	0	0	0	0	0	0	0	**958**
	7	2	4	15	4	1	1	0	989	3	9	0	0	0	0	0	0	0	0	0	0	**1028**
TRUE	8	1	3	4	19	11	4	6	6	902	18	0	0	0	0	0	0	0	0	0	0	**974**
	9	4	6	0	12	21	4	0	12	5	945	0	0	0	0	0	0	0	0	0	0	**1009**
	10	0	0	0	0	0	0	0	0	0	0	191	20	33	72	69	3	13	77	480	42	**1000**
	11	0	0	0	0	0	0	0	0	0	0	20	233	11	65	72	2	27	61	334	175	**1000**
	12	0	0	0	0	0	0	0	0	0	0	61	28	90	147	168	11	59	91	324	21	**1000**
	13	0	0	0	0	0	0	0	0	0	0	51	28	46	264	167	26	45	73	243	57	**1000**
	14	0	0	0	0	0	0	0	0	0	0	42	17	52	163	303	4	46	111	233	29	**1000**
	15	0	0	0	0	0	1	0	0	0	0	50	13	54	173	168	96	31	85	299	30	**1000**
	16	0	0	0	0	1	0	0	0	0	0	30	44	34	229	163	11	133	76	228	51	**1000**
	17	0	0	0	0	0	0	0	0	0	0	23	20	42	113	163	13	25	254	293	54	**1000**
	18	0	0	1	0	0	0	0	0	0	0	25	31	6	60	25	10	9	40	717	76	**1000**
	19	1	0	0	0	0	0	0	0	0	0	32	80	15	44	41	2	14	52	346	373	**1000**
	total	**995**	**1136**	**1016**	**1023**	**980**	**874**	**956**	**1031**	**977**	**1016**	**525**	**514**	**383**	**1330**	**1339**	**178**	**402**	**920**	**3497**	**908**	**20000**

Figure 6. Overall confusion matrix of 1-hidden-layer gate. Row and column indices represent the true and predicted labels. Classes 0-9 and 10-19 are mnist and cifar-10 labels correspondingly. This is computed on the test set of 20000 elements.

EXPERT 1 CONFUSION		PREDICTION																			
	class	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	total
	0	965	0	1	0	0	3	2	2	0	1	0	0	0	0	0	0	0	0	0	0	**974**
	1	0	4	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	**6**
	2	14	1	890	3	1	2	1	10	1	0	0	0	0	0	0	0	0	0	0	0	**923**
	3	0	0	4	20	0	3	0	5	0	0	0	0	0	0	0	0	0	0	0	0	**32**
	4	0	0	4	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	**6**
	5	2	0	0	2	0	344	0	4	0	0	0	0	0	0	0	0	0	0	0	0	**352**
	6	6	0	0	0	0	2	95	0	0	0	0	0	0	0	0	0	0	0	0	0	**103**
	7	2	0	14	0	1	1	0	978	0	0	0	0	0	0	0	0	0	0	0	0	**996**
TRUE	8	1	0	3	1	0	1	0	6	0	0	0	0	0	0	0	0	0	0	0	0	**12**
	9	4	0	0	0	0	1	0	12	0	4	0	0	0	0	0	0	0	0	0	0	**21**
	10	0	0	0	0	0	0	0	0	0	0	191	20	33	72	69	3	13	77	480	42	**1000**
	11	0	0	0	0	0	0	0	0	0	0	20	233	11	65	72	2	27	61	334	175	**1000**
	12	0	0	0	0	0	0	0	0	0	0	61	28	90	147	168	11	59	91	324	21	**1000**
	13	0	0	0	0	0	0	0	0	0	0	51	28	46	264	167	26	45	73	243	57	**1000**
	14	0	0	0	0	0	0	0	0	0	0	42	17	52	163	303	4	46	111	233	29	**1000**
	15	0	0	0	0	0	1	0	0	0	0	50	13	54	173	168	96	31	85	299	30	**1000**
	16	0	0	0	0	0	0	0	0	0	0	30	44	34	229	163	11	133	76	228	51	**999**
	17	0	0	0	0	0	0	0	0	0	0	23	20	42	113	163	13	25	254	293	54	**1000**
	18	0	0	1	0	0	0	0	0	0	0	25	31	6	60	25	10	9	40	717	76	**1000**
	19	1	0	0	0	0	0	0	0	0	0	32	80	15	44	41	2	14	52	346	373	**1000**
	total	**995**	**5**	**918**	**26**	**2**	**359**	**98**	**1019**	**1**	**5**	**525**	**514**	**383**	**1330**	**1339**	**178**	**402**	**920**	**3497**	**908**	**13424**

Figure 5. Confusion matrix of expert 1 of 1-hidden-layer gate. Row and column indices represent the true and predicted labels. Classes 0-9 and 10-19 are mnist and cifar-10 labels correspondingly. This expert's responsible set has 13424 elements.

5.2 Result 2 – Mixture datasets, convolutional experts, different gating functions

Comparing to the accuracy of linear experts, the accuracy of convolutional experts (Figure 7) is higher. The mixture of convolutional experts is not performing better than using a single expert in this task.

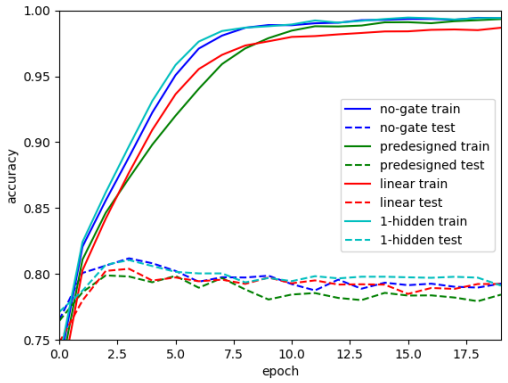

Figure 7 Epoch-Accuracy plot with convolutional experts on mixture dataset.

Figure 6 shows the activation on super-labels. The 1-hidden layer gate is specialized to separate most of mnist and cifar-10 images, whereas the linear gate is relying on expert 1 and ignoring expert 2.

Model		Expert 1	Expert 2
Linear gate	Mnist	1.00	0.00
	Cifar	1.00	0.00
1-hidden layer gate	Mnist	0.88	0.12
	Cifar	0.00	1.00

Figure 8 Activation matrix with convolutional experts on mixture dataset

5.3 Result 3 – Mixture dataset, linear experts, different number of experts

The number of experts is in the list [1,2,3,4,6,8,10,12,16,20]. Not every number in the range [1,20] is evaluated. Figure 9 shows that the accuracy increases when number of experts increases. The accuracy begins to converge when number of experts is greater than 4.

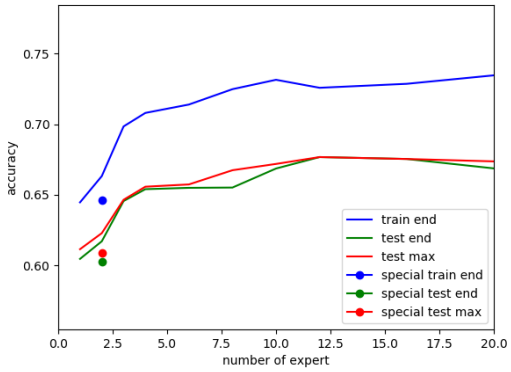

Figure 9 number of experts - accuracy plot on mixture dataset, with 1-hidden-layer gate and linear experts. The special dots represent the result of pre-designed gate. Blue is the accuracy on the train set after training, green is the accuracy on the test set after training, and red is the maximum accuracy after each training epoch on the test set.

From activation matrices (Figure 10), we could see, there are few experts being pinched off when number of experts is high. We could see also that some experts specialize on mnist, some on cifar-10, while some predicts both.

ACTIVATION	EXPERT																			
N=1																				
MNIST	1																			
CIFAR-10	1																			
N=2																				
MNIST	0.67	0.33																		
CIFAR-10	0	1																		
N=3																				
MNIST	0.32	0	0.68																	
CIFAR-10	0.38	0.54	0.08																	
N=4																				
MNIST	0.05	0.05	0.52	0.39																
CIFAR-10	0.41	0.34	0.02	0.23																
N=6																				
MNIST	0.34	0.23	0	0.43	0	0														
CIFAR-10	0.24	0.15	0.14	0.09	0.09	0.29														
N=8																				
MNIST	0.27	0.08	0.06	0	0	0.28	0.32	0												
CIFAR-10	0	0.22	0.19	0.16	0.02	0.08	0.15	0.18												
N=10																				
MNIST	0.18	0	0.28	0.22	0	0.32	0	0	0	0										
CIFAR-10	0.22	0.07	0.08	0.11	0.1	0.14	0.06	0.03	0.08	0.14										
N=12																				
MNIST	0.24	0.23	0.11	0	0	0	0	0	0	0	0.38	0.03								
CIFAR-10	0.04	0.17	0	0.08	0.24	0.06	0.06	0.08	0.1	0.06	0.01	0.09								
N=16																				
MNIST	0	0.46	0	0	0	0	0	0	0	0.02	0	0	0.32	0	0.2	0				
CIFAR-10	0.11	0.15	0.15	0.05	0.02	0.04	0.07	0.02	0.1	0.06	0.05	0.03	0.01	0.04	0.05	0.06				
N=20																				
MNIST	0	0	0	0	0.25	0	0	0	0	0	0.2	0	0	0	0	0.22	0	0.21	0.13	0
CIFAR-10	0.03	0	0.04	0.09	0.01	0.04	0.06	0	0.03	0.13	0.02	0.05	0.03	0.03	0	0.17	0.11	0.06	0.07	0.05

Figure 10. Activation matrices with different number of experts with linear experts on mixture dataset.

Here, we plot the activation (Figure 11) and confusion (Figure 12Figure 13Figure 14,Figure 15Figure 16) of 4 experts to see some details.

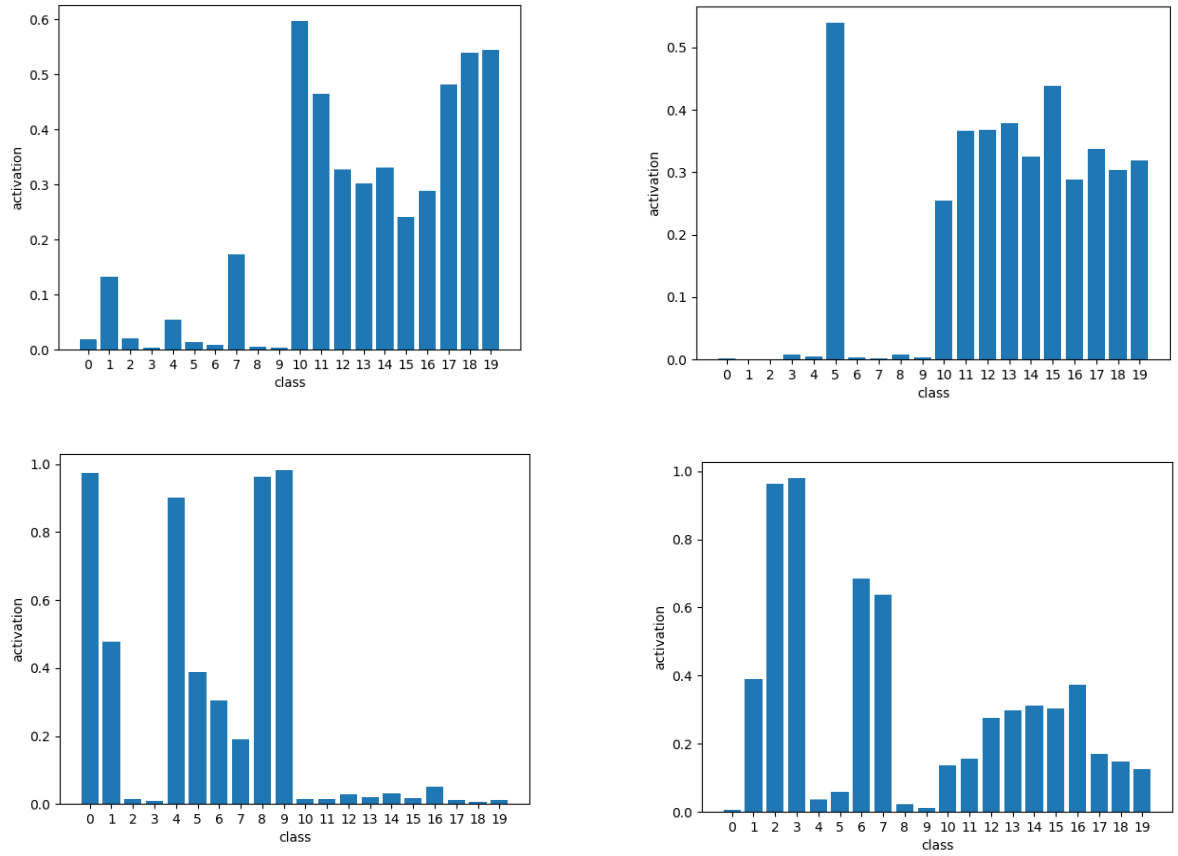

Figure 11. Activation per class of 4 experts. Top left is 1. expert, top right is 2., bot left is 3. And bot right is 4. Notice that each y axes have different scales.

OVERALL CONFUS.		PREDICTION																				
	class	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	total
	0	964	0	0	1	1	2	7	1	3	1	0	0	0	0	0	0	0	0	0	0	**980**
	1	0	1122	2	3	1	0	2	2	3	0	0	0	0	0	0	0	0	0	0	0	**1135**
	2	6	5	975	22	5	1	6	8	3	0	1	0	0	0	0	0	0	0	0	0	**1032**
	3	0	0	12	977	1	5	0	9	5	1	0	0	0	0	0	0	0	0	0	0	**1010**
	4	0	1	3	1	935	1	3	2	3	33	0	0	0	0	0	0	0	0	0	0	**982**
	5	4	0	3	17	2	843	7	1	10	5	0	0	0	0	0	0	0	0	0	0	**892**
	6	7	3	10	0	6	8	917	2	5	0	0	0	0	0	0	0	0	0	0	0	**958**
	7	0	5	18	12	5	1	0	976	3	8	0	0	0	0	0	0	0	0	0	0	**1028**
TRUE	8	4	0	4	15	9	16	2	5	909	10	0	0	0	0	0	0	0	0	0	0	**974**
	9	7	5	1	3	13	8	0	6	12	954	0	0	0	0	0	0	0	0	0	0	**1009**
	10	0	0	0	0	0	0	0	0	0	0	508	27	48	61	50	4	55	95	126	26	**1000**
	11	0	0	0	0	0	0	0	0	0	0	111	395	12	63	27	6	68	54	140	124	**1000**
	12	0	0	0	0	0	0	0	0	0	0	151	20	164	171	156	37	146	89	50	16	**1000**
	13	0	0	0	0	0	0	0	0	0	0	89	29	70	302	81	77	150	88	64	50	**1000**
	14	0	0	0	0	0	0	0	0	0	0	128	16	75	161	262	16	165	122	41	14	**1000**
	15	0	0	0	0	0	1	0	0	0	0	83	12	81	254	63	192	90	107	94	23	**1000**
	16	0	0	0	0	0	0	0	0	0	0	91	45	44	142	119	21	418	64	32	24	**1000**
	17	0	0	0	0	0	0	0	0	0	0	108	19	62	124	61	29	55	435	66	41	**1000**
	18	0	0	0	0	0	0	0	0	0	0	216	73	23	56	21	12	28	49	460	62	**1000**
	19	0	0	0	0	1	0	0	0	0	0	99	148	19	66	16	10	50	74	145	372	**1000**
	total	**992**	**1141**	**1028**	**1051**	**979**	**886**	**944**	**1012**	**956**	**1012**	**1585**	**784**	**598**	**1400**	**856**	**404**	**1225**	**1177**	**1218**	**752**	**20000**

Figure 12. Overall confusion of 4 experts.

EXPERT 1 CONFUSION		PREDICTION																				
	class	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	total
	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**17**
	1	0	135	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**135**
	2	1	1	18	0	1	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	**23**
	3	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	**4**
	4	0	0	0	0	46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**46**
	5	0	0	0	0	0	10	0	1	0	0	0	0	0	0	0	0	0	0	0	0	**11**
	6	1	0	2	0	4	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	**10**
	7	0	0	1	0	1	0	0	172	0	0	0	0	0	0	0	0	0	0	0	0	**174**
TRUE	8	0	0	0	0	2	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	**4**
	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
	10	0	0	0	0	0	0	0	0	0	0	465	17	24	7	11	0	17	82	79	16	**718**
	11	0	0	0	0	0	0	0	0	0	0	94	174	5	12	2	0	20	29	91	80	**507**
	12	0	0	0	0	0	0	0	0	0	0	132	12	70	20	21	1	18	57	19	7	**357**
	13	0	0	0	0	0	0	0	0	0	0	81	13	31	35	10	4	52	63	16	24	**329**
	14	0	0	0	0	0	0	0	0	0	0	111	9	36	25	36	0	22	86	22	6	**353**
	15	0	0	0	0	0	0	0	0	0	0	71	6	27	24	4	4	27	65	11	14	**253**
	16	0	0	0	0	0	0	0	0	0	0	85	29	16	37	20	0	96	32	14	12	**341**
	17	0	0	0	0	0	0	0	0	0	0	101	5	30	11	10	1	11	325	36	28	**558**
	18	0	0	0	0	0	0	0	0	0	0	200	41	9	7	3	0	7	37	281	38	**623**
	19	0	0	0	0	0	0	0	0	0	0	98	77	3	19	2	2	28	54	86	264	**633**
	total	**19**	**136**	**21**	**0**	**54**	**11**	**2**	**181**	**0**	**0**	**1438**	**383**	**251**	**197**	**119**	**12**	**298**	**830**	**655**	**489**	**5096**

Figure 13. Confusion of Expert 1 from 4 experts.

EXPERT 2 CONFUSION		PREDICTION																				
	class	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	total
	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
	3	0	0	0	1	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**5**
	4	0	0	0	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**4**
	5	0	0	0	0	0	475	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**475**
	6	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**2**
	7	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**
TRUE	8	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**6**
	9	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**2**
	10	0	0	0	0	0	0	0	0	0	0	34	6	17	45	5	2	6	11	34	5	**165**
	11	0	0	0	0	0	0	0	0	0	0	15	194	3	43	4	2	6	23	39	28	**357**
	12	0	0	0	0	0	0	0	0	0	0	16	7	71	118	34	14	9	24	30	6	**329**
	13	0	0	0	0	0	0	0	0	0	0	7	15	27	195	12	17	16	23	44	10	**366**
	14	0	0	0	0	0	0	0	0	0	0	14	6	27	107	49	3	3	29	15	5	**258**
	15	0	0	0	0	0	0	0	0	0	0	8	5	40	166	16	85	10	34	71	7	**442**
	16	0	0	0	0	0	0	0	0	0	0	3	13	9	69	9	5	27	25	11	2	**173**
	17	0	0	0	0	0	0	0	0	0	0	6	14	25	90	15	10	8	98	28	9	**303**
	18	0	0	0	0	0	0	0	0	0	0	14	27	12	37	5	4	4	11	159	11	**284**
	19	0	0	0	0	0	0	0	0	0	0	0	58	12	39	5	4	7	19	54	74	**272**
	total	**0**	**0**	**0**	**1**	**4**	**491**	**0**	**0**	**0**	**0**	**117**	**345**	**243**	**909**	**154**	**146**	**96**	**297**	**485**	**157**	**3445**

Figure 14. Confusion of Expert 2 from 4 experts.

EXPERT 3 CONFUSION		PREDICTION																				
	class	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	total
	0	946	0	0	0	1	1	4	1	3	1	0	0	0	0	0	0	0	0	0	0	**957**
	1	0	550	0	0	0	0	1	0	3	0	0	0	0	0	0	0	0	0	0	0	**554**
	2	5	0	2	0	2	1	0	1	3	0	0	0	0	0	0	0	0	0	0	0	**14**
	3	0	0	0	1	0	0	0	0	5	1	0	0	0	0	0	0	0	0	0	0	**7**
	4	0	1	1	0	854	0	2	1	3	33	0	0	0	0	0	0	0	0	0	0	**895**
	5	4	0	0	0	2	327	7	0	10	5	0	0	0	0	0	0	0	0	0	0	**355**
	6	5	2	1	0	2	5	270	0	5	0	0	0	0	0	0	0	0	0	0	0	**290**
	7	0	1	0	0	1	1	0	179	3	8	0	0	0	0	0	0	0	0	0	0	**193**
TRUE	8	4	0	1	0	6	9	2	2	908	10	0	0	0	0	0	0	0	0	0	0	**942**
	9	7	2	0	1	12	6	0	3	12	954	0	0	0	0	0	0	0	0	0	0	**997**
	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	**2**
	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	**2**
	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
	15	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**
	16	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	**3**
	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
	19	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**
	total	**971**	**556**	**5**	**2**	**881**	**351**	**286**	**187**	**955**	**1012**	**0**	**1**	**0**	**0**	**3**	**0**	**3**	**0**	**0**	**0**	**5213**

Figure 15. Confusion of Expert 3 from 4 experts.

EXPERT 4 CONFUSION		PREDICTION																				
	class	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	total
	0	1	0	0	1	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	**5**
	1	0	437	2	3	1	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	**446**
	2	0	4	955	22	2	0	6	5	0	0	1	0	0	0	0	0	0	0	0	0	**995**
	3	0	0	12	975	1	1	0	5	0	0	0	0	0	0	0	0	0	0	0	0	**994**
	4	0	0	2	1	32	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	**37**
	5	0	0	3	17	0	31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**51**
	6	1	1	7	0	0	1	645	1	0	0	0	0	0	0	0	0	0	0	0	0	**656**
	7	0	4	17	12	2	0	0	625	0	0	0	0	0	0	0	0	0	0	0	0	**660**
TRUE	8	0	0	3	15	1	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	**22**
	9	0	3	1	2	1	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	**10**
	10	0	0	0	0	0	0	0	0	0	0	9	4	7	9	34	2	32	2	13	5	**117**
	11	0	0	0	0	0	0	0	0	0	0	2	27	4	8	21	4	42	2	10	16	**136**
	12	0	0	0	0	0	0	0	0	0	0	3	1	23	33	100	22	118	8	1	3	**312**
	13	0	0	0	0	0	0	0	0	0	0	1	1	12	72	58	56	81	2	4	16	**303**
	14	0	0	0	0	0	0	0	0	0	0	3	1	12	29	177	13	140	7	4	3	**389**
	15	0	0	0	0	0	0	0	0	0	0	4	1	14	64	43	103	53	8	12	2	**304**
	16	0	0	0	0	0	0	0	0	0	0	3	2	19	36	89	16	294	7	7	10	**483**
	17	0	0	0	0	0	0	0	0	0	0	1	0	7	23	36	18	36	12	2	4	**139**
	18	0	0	0	0	0	0	0	0	0	0	2	5	2	12	13	8	17	1	20	13	**93**
	19	0	0	0	0	0	0	0	0	0	0	1	13	4	8	9	4	15	1	5	34	**94**
	total	**2**	**449**	**1002**	**1048**	**40**	**33**	**656**	**644**	**1**	**0**	**30**	**55**	**104**	**294**	**580**	**246**	**828**	**50**	**78**	**106**	**6246**

Figure 16. Confusion of Expert 4 from 4 experts

5.4 Result 4 – Cifar-100, Linear experts, different number of experts

This experiment has exactly same setting as previous one, except the dataset is changed to cifar-100. This causes input and output to be sized from (28,28,1) and 20 to (32,32,3) and 100.

However, the accuracy plot (Figure 9) is reversed case from previous one: Accuracy decreases when number of experts increases. With low number of experts, the mixture of experts performs better than a single expert, but with high number of experts, it performs worse. I could not find an intuitive explanation for this.

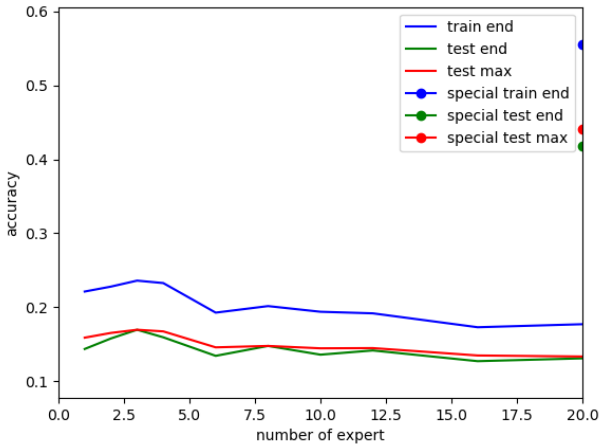

Figure 17. number of experts - accuracy plot on cifar-100 dataset. The special dots represent the result of pre-designed gate. Blue is the accuracy on the train set after training, green is the accuracy on the test set after training, and red is the maximum accuracy after each training epoch on the test set.

We plot the super-label activation table (Figure 18) and activation-per-class (Figure 19) of 4 experts. In this version of implementation, I forgot to group the classes into super-classes (i.e. the classes 0-4 are not super-class 0 and the classes 5-9 are not super-class 1 and so on), the activation-per-class figure is less informing. The expert 1 of 4 experts seems to be ignored and expert 2 has activation greater than 0.5 for most of classes.

Model		Expert 1	Expert 2	Expert 3	Expert 4
4 experts	aquatic_mammals	0.01	0.57	0.23	0.19
	fish	0.03	0.56	0.23	0.18
	flowers	0.01	0.53	0.42	0.04
	food_containers	0.01	0.58	0.31	0.09
	fruit_and_vegetables	0.01	0.52	0.41	0.06
	household_electrical_devices	0.01	0.58	0.29	0.12
	household_furniture	0.01	0.60	0.33	0.06
	insects	0.01	0.58	0.33	0.08
	large_carnivores	0.02	0.55	0.35	0.08
	large_man-made_outdoor_things	0.01	0.55	0.18	0.27
	large_natural_outdoor_scenes	0.01	0.56	0.18	0.25
	large_omnivores_and_herbivores	0.01	0.58	0.30	0.12
	medium_mammals	0.02	0.58	0.31	0.09
	non-insect_invertebrates	0.02	0.57	0.31	0.09
	people	0.02	0.58	0.35	0.06
	reptiles	0.02	0.58	0.29	0.11
	small_mammals	0.02	0.57	0.33	0.08
	trees	0.00	0.43	0.19	0.37
	vehicles_1	0.01	0.64	0.21	0.15
	vehicles_2	0.01	0.64	0.20	0.15

Figure 18. Super-label activation on 4 experts.

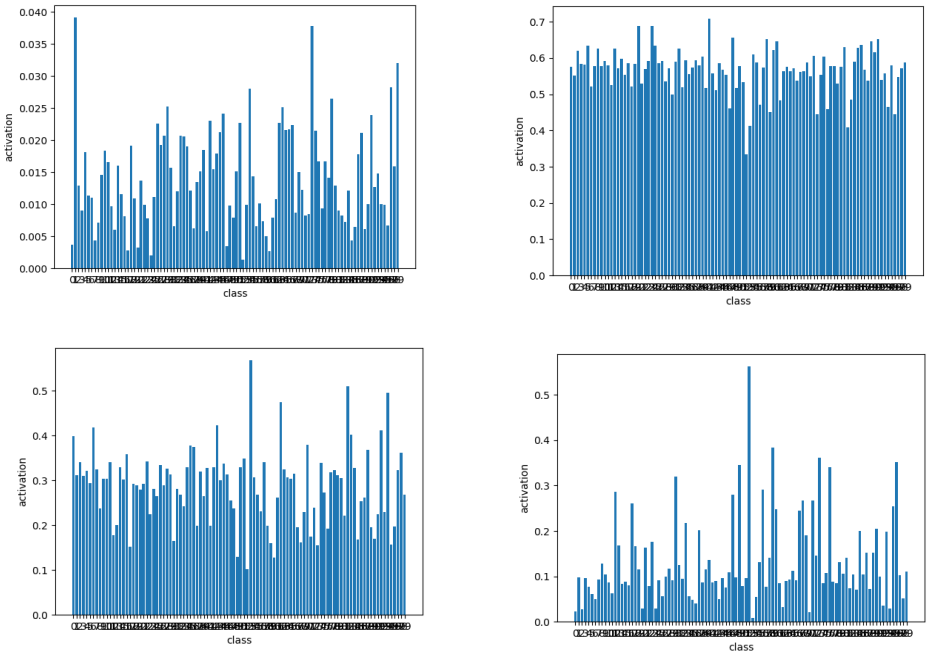

Figure 19. The per-class-activation on 4 experts. The x-axis is classes from 0 to 99. Classes are not grouped into super-classes by indices. Notice the y-axis has different scales.

5.5 Result 5 – Cifar-100, convolutional experts, different number of experts.

The number-of-experts-accuracy plot on this experiment (Figure 20) shows a different result from results 3 and 4. The different number of experts has little to no effect to the performance.

In both result 4 and 5, the pre-designed gate is performing significantly better than learning the gate. This result is different from models trained on mixture datasets, where the pre-designed gate is not performing better than a single expert.

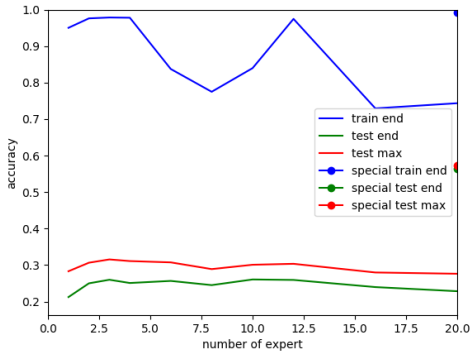

Figure 20. number of experts - accuracy plot on cifar-100 dataset with convolutional experts. The special dots represent the result of pre-designed gate. Blue is the accuracy on the train set after training, green is the accuracy on the test set after training, and red is the maximum accuracy after each training epoch on the test set.

For comparison, the per-class-activation of 4 experts is plotted. We could already see from it that every other expert is ignored except expert 1. The reason of stable accuracy of different number of experts might be this: there might be always only 1 expert that is activated.

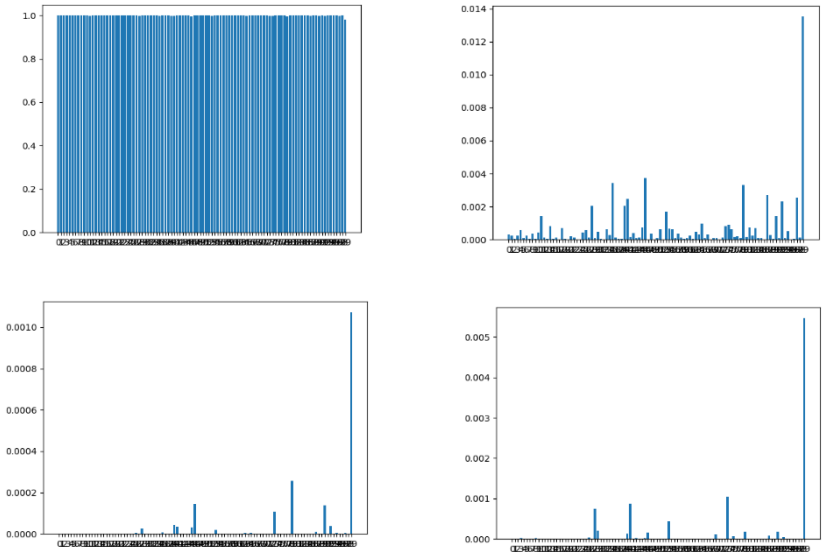

Figure 21. The per-class-activation on 4 convolutional experts. The x-axis is classes from 0 to 99. Classes are not grouped into super-classes by indices. Notice the y-axis has different scales.

5.6 Result 6 – Mixture dataset, convolutional experts, different number of experts

The accuracy (Figure 22) is also stable when number of experts increase. When we look at activations (Figure 23), we notice that there are many ignored experts. Some mixture has experts that are gated exactly as the pre-designed gate (with some 0-weighted experts). The per-class-activation and confusion of 4 experts are not plotted for comparison, because they behave similarly as single experts.

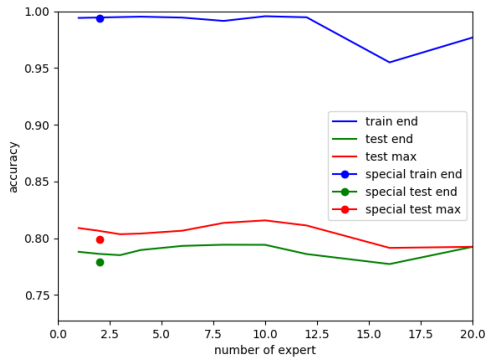

Figure 22. Number of experts - accuracy plot on mixture dataset with convolutional experts. Dots are results on pre-designed gate. Blue is accuracy on train set after training, green is accuracy on test set after training, red is the maximum accuracy on test set after each training epoch.

ACTIVATION	EXPERT																			
N=1																				
MNIST	1																			
CIFAR-10	1																			
N=2																				
MNIST	0	1																		
CIFAR-10	0	1																		
N=3																				
MNIST	0.61	0.05	0.35																	
CIFAR-10	0	1	0																	
N=4																				
MNIST	1	0	0	0																
CIFAR-10	0	1	0	0																
N=6																				
MNIST	0.78	0	0	0	0	0.22														
CIFAR-10	0	0	0	1	0	0														
N=8																				
MNIST	0.89	0.11	0.01	0	0	0	0	0												
CIFAR-10	0	0	0.02	0	0	0	0.98	0												
N=10																				
MNIST	0	0	0	0	0	0	0	0	1	0										
CIFAR-10	1	0	0	0	0	0	0	0	0	0										
N=12																				
MNIST	0	0	0	0	1	0	0	0	0	0	0	0								
CIFAR-10	0	0	0	0	0	0	1	0	0	0	0	0								
N=16																				
MNIST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1				
CIFAR-10	0	0.39	0	0	0	0	0	0	0	0	0	0	0	0	0	0.61				
N=20																				
MNIST	0	0.11	0	0	0	0	0	0	0	0	0.89	0	0	0	0	0	0	0	0	0
CIFAR-10	0	0	0	0	0	0	0	0	0	0	0.46	0	0	0	0.54	0	0	0	0	0

Figure 23. Super-label activation of different number of convolutional experts on mixture dataset.

5.7 Result 7 – Mixture dataset, heterogeneous experts, different gating functions

In this experiment, the mixture of experts is a mixture of a linear expert and a convolutional expert. We want to see if simpler (mnist) images can be directed to linear experts and more complex (cifar) images can be directed to convolutional expert. From both accuracy (Figure 24) and activation (Figure 25), we could see that the convolutional expert is handling images from both datasets.

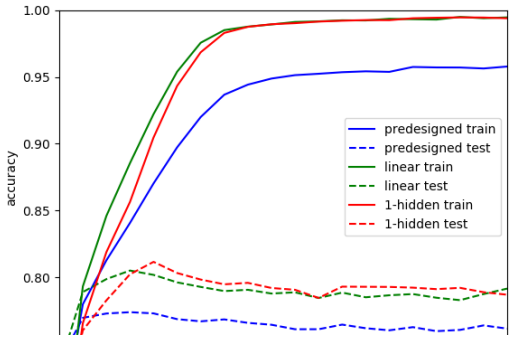

Figure 24. Accuracy on heterogeneous experts with different gates. Blue is the pre-designed gate, green is the linear gate, red is 1-hidden-layer gate. Solid line is on train data, dashed line on test data.

Model		Linear Expert	Convolutional Expert
Linear gate	Mnist	0.00	1.00
	Cifar	0.00	1.00
1-hidden layer gate	Mnist	0.00	1.00
	Cifar	0.00	1.00

Figure 25. Super-label activation table on heterogeneous experts.

6. MoE applications

(No changes here. I did not have enough time to look for more modern applications after the discussion 2 weeks ago)

Most of MoE-applications are related to time series mentioned in (Waterhouse, 1998). This includes control tasks with reinforcement learning, time series prediction and neural translation. As far as I understood, those tasks are related to capturing switching temporal patterns. Then the gating network tries to predict which temporal pattern is and the experts try to behave as they are in currently in the temporal pattern.

To capture temporal patterns and switch between them, MoE should be applied on every step, and gating receives input of current step and previous hidden state. In visual tasks, this kind of architecture might be more expensive then CNN. This also might be able to be generalized in the graph network framework. In tasks with only general features, the steps and the pattern between steps are undefined, and applying MoE per step would not be helpful.

Just one example of temporal patterns. Consider we have a computer generating series of numbers. In a period, it generates numbers by a rule (for example sin(x)) with noise. Later, the rule switched to another one (for example a series of zigzag) and continue to generate numbers during another period. This, with an alternative generating and switching rules, is a toy case of (Weigend;Mangeas;& Srivastava, 1995).

"Temporal patterns" mean here the rules generating the series in given periods. Gating network tries to predict if it is currently zigzag or sin(x). In some cases, the period is not decided randomly, but there could be some long-term trigger to be captured by hidden state, and temporal patterns are more complex. MoE should also be useful in this case.

Bibliography

Waterhouse, S. R. (1998). *Classification and Regression using Mixtures of Experts.*

Weigend, A. S., Mangeas, M., & Srivastava, A. N. (1995). *Nonlinear gated experts for time series:.*