Mixture of Experts - Experiment 1-5

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: without-gating, pre-designed gate, linear gate, and 1-hidden-layer gate.

Without gating, the number of experts is 1. So, $G(X)_1$ = 1 for the only expert 1 and $Y = E_1(X)$.

The pre-designed gate is only available for the mixture of dataset described in section 3. The number of experts is 2, and the gating can be expressed as such: $G(X) = IF\ IsMnist(X)\ THEN\ [1,0]\ ELSE\ [0,1]$. The function $IsMnist$ is related to the datasets, and it returns true, if the input X is from the dataset Mnist.

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.

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. Both mnist and cifar-10 have images with 1 of 10 labels, 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. Images from both datasets are resized to 32x32 and then gray-scaled.

The cifar-100 is a more complex version of cifar-10 with the number of class being 100. We do not gray-scale the images.

4. Experiments

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

4.1 Experiment 1

Here, we use the mixture of datasets. The experts are selected to be linear. We iterate through all gating choices and set the number of experts to 2 for all except without-gating.

We plot the epoch-accuracy curves of different gating choices on both train and test set. The epoch-accuracy curves would tell how well the models perform. We assume 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 plot a table of the activation matrices, where the define the activation rate are defined as:

$$ActivationRate(i|D) = Mean[G(X)_i|X\ in\ D]$$

This table could tell us, how does the gate separate the data into the experts.

4.2 Experiment 2

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

We expect the model to be too powerful, and the accuracy is too high to see clear difference.

4.3 Experiment 3

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 numberOfExperts-accuracy curves.

We plot the activation matrices and the confusion matrices. There are different confusion matrices: Overall confusion and expert confusion.

The cell $C_{i,j}$ of the overall confusion matrix C is the number of true label i predicted to be j.

The cell $C_{e,i,j}$ of the expert confusion matrix C_e is the number of true label i predicted to be j from the expert responsible set $R_e = \{X|argmax\big(G(X)\big) = e\ \}$.

We could see from confusion matrix, which true classes are responsible to classify, and how it success.

4.4 Experiment 4

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

4.5 Experiment 5

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

5. Results and analysis

5.1 Result 1

The **Error! Reference source not found.** 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.

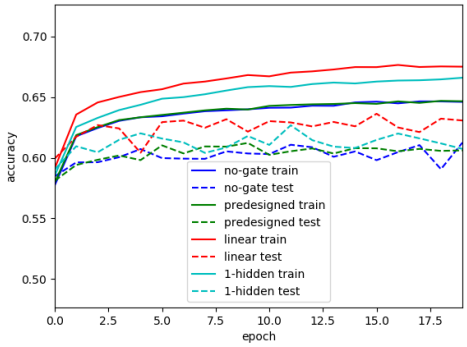

Figure 1. The accuracy curves of different models with linear experts on mixture dataset. Blue is the single expert, green is the pre-designed gate, red is the linear gate, cyan is 1-hidden-layer gate. Solid line is on train data, dashed line on test data.

The activation matrices (**Error! Reference source not found.**2) shows that the frequency of using experts are unbalanced. In the case of 1-hidden layer gate, expert 1 predicts on only part of mnist, whereas expert 2 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	0.87	0.13
	Cifar	0.43	0.57
1-hidden layer gate	Mnist	0.54	0.46
	Cifar	0.00	1.00

Figure 2 The activation matrices on the test set after training.

Since the confusion matrices is large, only matrices of overall confusion and expert 2 confusion of 1-hidden layer gate are showed (Figure 3, 4). The mnist images are simpler to be classified as

shown in overall confusion, and the linear expert 2 classifies most cifar-10 images into the 9. Class of cifar-10.

OVERALL CONFUSION	PREDICTION																			
	962	0	2	1	0	5	1	2	4	3	0	0	0	0	0	0	0	0	0	0
	0	1126	2	1	0	1	3	1	1	0	0	0	0	0	0	0	0	0	0	0
	5	4	951	21	5	3	7	11	23	2	0	0	0	0	0	0	0	0	0	0
	4	0	12	943	1	15	1	13	12	9	0	0	0	0	0	0	0	0	0	0
	3	0	3	1	938	0	7	1	3	26	0	0	0	0	0	0	0	0	0	0
	5	1	2	30	2	817	12	1	15	7	0	0	0	0	0	0	0	0	0	0
	3	3	4	0	4	8	932	2	1	1	0	0	0	0	0	0	0	0	0	0
	0	6	9	4	5	1	0	985	2	16	0	0	0	0	0	0	0	0	0	0
TRUE	5	1	4	29	3	12	5	13	893	9	0	0	0	0	0	0	0	0	0	0
	6	7	1	5	24	1	1	9	4	951	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	188	20	32	76	64	4	11	80	483	42
	0	0	0	0	0	0	0	0	0	0	22	229	9	63	74	2	27	55	332	187
	0	0	0	0	0	0	0	0	0	0	55	28	96	146	158	11	56	96	333	21
	0	0	0	0	0	0	0	0	0	0	49	27	49	253	172	28	46	75	245	56
	0	0	0	0	0	0	0	0	0	0	41	16	55	166	304	7	48	110	225	28
	0	0	0	0	0	1	0	0	0	0	48	13	54	172	169	100	27	87	296	33
	0	0	0	0	1	0	0	0	0	0	35	49	37	234	153	12	132	74	230	43
	0	0	0	0	0	0	0	0	0	0	23	22	41	110	159	13	28	263	282	59
	0	0	0	0	0	0	0	0	0	0	26	30	8	62	26	10	9	40	714	75
	0	0	0	0	0	0	0	0	1	0	27	82	15	40	43	4	18	51	350	369

Figure 3 Overall confusion on 1-hidden-layer-gate on the test set after training.

EXPERT 2 CONFUSION	PREDICTION																			
	13	0	1	0	0	1	1	2	0	1	0	0	0	0	0	0	0	0	0	0
	0	844	0	0	0	1	3	1	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	31	1	5	0	7	3	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	1	6	1	2	1	5	0	4	0	0	0	0	0	0	0	0	0	0
	2	0	2	0	937	0	7	0	1	26	0	0	0	0	0	0	0	0	0	0
	1	1	1	2	2	214	11	0	0	4	0	0	0	0	0	0	0	0	0	0
	0	1	1	0	4	4	916	2	0	1	0	0	0	0	0	0	0	0	0	0
	0	5	0	0	5	1	0	527	0	16	0	0	0	0	0	0	0	0	0	0
TRUE	0	0	0	1	3	2	5	1	0	6	0	0	0	0	0	0	0	0	0	0
	1	7	1	0	24	0	1	6	0	939	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	188	20	32	76	64	4	11	80	483	42
	0	0	0	0	0	0	0	0	0	0	22	229	9	63	74	2	27	55	332	187
	0	0	0	0	0	0	0	0	0	0	55	28	96	146	158	11	56	96	333	21
	0	0	0	0	0	0	0	0	0	0	49	27	49	253	172	28	46	75	245	56
	0	0	0	0	0	0	0	0	0	0	41	16	55	166	304	7	48	110	225	28
	0	0	0	0	0	0	0	0	0	0	48	13	54	172	169	100	27	87	296	33
	0	0	0	0	1	0	0	0	0	0	35	49	37	234	153	12	132	74	230	43
	0	0	0	0	0	0	0	0	0	0	23	22	41	110	159	13	28	263	282	59
	0	0	0	0	0	0	0	0	0	0	26	30	8	62	26	10	9	40	714	75
	0	0	0	0	0	0	0	0	1	0	27	82	15	40	43	4	18	51	350	369

Figure 4 Confusion matrix on expert 2 of 1-hidden-layer gate.

5.2 Result 2

The accuracy plot (Figure 5) shows that convolutional models are too powerful on training data, that mixture of experts makes barely difference.

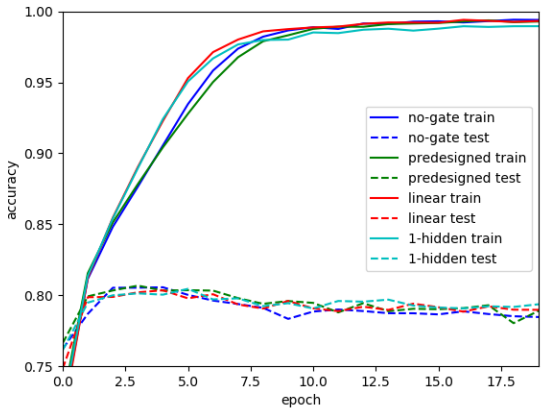

Figure 5 Epoch-Accuracy plot with convolutional experts on mixture dataset.

The activation matrix (Figure 6) shows, that the experts are highly specialized on one of mnist of cifar-10.

Model		Expert 1	Expert 2
Linear gate	Mnist	0.19	0.81
	Cifar	0.99	0.01
1-hidden layer gate	Mnist	0.00	1.00
	Cifar	1.00	0.00

Figure 6 Activation matrix with convolutional experts on mixture dataset

5.3 Result 3

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 7 shows that the accuracy increases when number of experts increases.

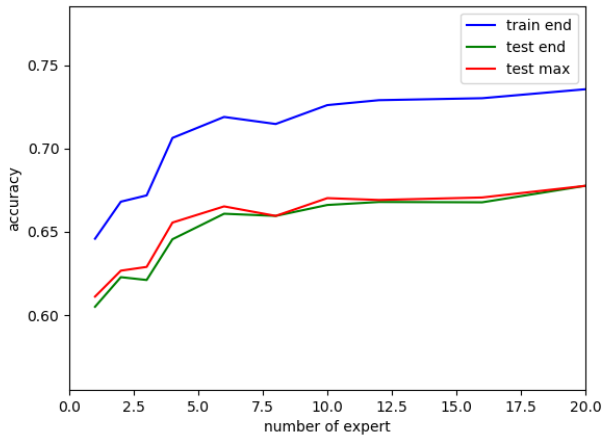

Figure 7 number of experts - accuracy plot on mixture dataset, with 1-hidden-layer gate and linear experts.

From activation matrices (Figure 8), 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. Confusion matrices are not plotted here since I do not know which cases are interesting.

ACTIVATION	EXPERT																			
N=1																				
MNIST	1.00																			
CIFAR-10	1.00																			
N=2																				
MNIST	0.29	0.71																		
CIFAR-10	1.00	0.00																		
N=3																				
MNIST	0.46	0.17	0.36																	
CIFAR-10	0.00	1.00	0.00																	
N=4																				
MNIST	0.36	0.35	0.17	0.12																
CIFAR-10	0.53	0.33	0.07	0.08																
N=6																				
MNIST	0.00	0.00	0.21	0.37	0.16	0.26														
CIFAR-10	0.07	0.32	0.20	0.17	0.01	0.23														
N=8																				
MNIST	0.43	0.01	0.00	0.02	0.12	0.42	0.00	0.00												
CIFAR-10	0.00	0.06	0.21	0.27	0.27	0.00	0.12	0.06												
N=10																				
MNIST	0.00	0.34	0.00	0.00	0.08	0.43	0.00	0.00	0.00	0.15										
CIFAR-10	0.09	0.04	0.09	0.09	0.25	0.03	0.18	0.14	0.08	0.01										
N=12																				
MNIST	0.12	0.31	0.22	0.36	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00								
CIFAR-10	0.01	0.08	0.17	0.06	0.02	0.17	0.06	0.18	0.06	0.06	0.07	0.05								
N=16																				
MNIST	0.09	0.25	0.00	0.00	0.13	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.17	0.00	0.00	0.27				
CIFAR-10	0.00	0.12	0.06	0.05	0.16	0.03	0.07	0.21	0.00	0.01	0.08	0.05	0.09	0.04	0.02	0.02				
N=20																				
MNIST	0.00	0.00	0.13	0.24	0.00	0.00	0.15	0.00	0.00	0.00	0.00	0.00	0.26	0.00	0.00	0.00	0.00	0.00	0.23	0.00
CIFAR-10	0.09	0.04	0.03	0.02	0.05	0.05	0.02	0.04	0.00	0.14	0.03	0.03	0.04	0.06	0.06	0.03	0.07	0.03	0.13	0.02

Figure 8. Activation matrices with different number of experts with linear experts on mixture dataset.

5.4 Result 4

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.

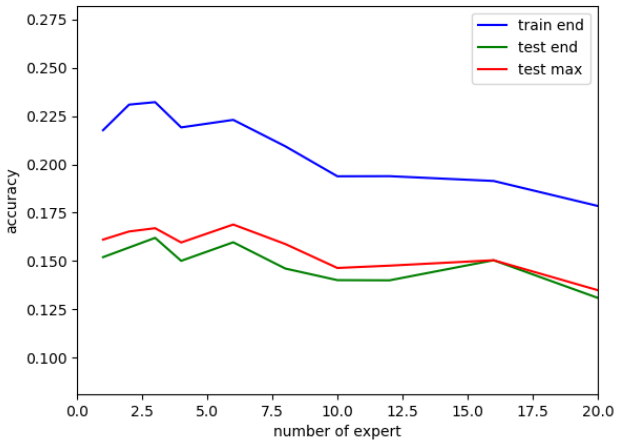

Figure 9. Accuracy plot on cifar100 dataset with linear experts.

However, the accuracy plot (Figure 9) is reversed case from previous one: Accuracy decreases when number of experts increases. I could not find an intuitive explanation for this. The confusion matrices are too large (100x100) to show.

5.5 Result 5

Due to memory issues, the number of experts is capped at 10.

The test accuracy (Figure 10) seemed to be stable independently from number of experts (though the plot is more zoomed in comparing to other accuracy plots).

The models are overfitting, and early-stop is required.

The gap on training accuracy might be a bit weird. The reason might be that the models need more epochs to converge (and overfit more) on training data. 20 epochs are trained for all models.

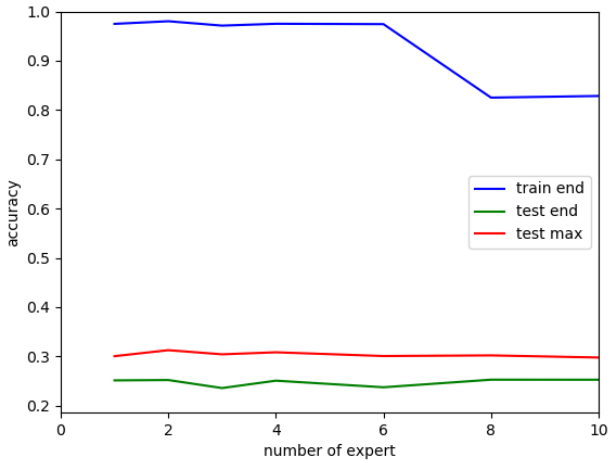

Figure 10. Accuracy with convolution experts on cifar-100

6. MoE applications

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:.*