Do Experts Specialize? A Mechanistic Exploration of Mixture of Experts Models

James Poetzscher
Stanford University
jamesp25@stanford.edu

Abstract

A large number of prominent large-language-models have, over the past few years, adopted the Mixture of Experts architecture to great success. These models, which seek to increase the parameter to compute ratio use multiple sparse MLPs, called experts, instead of a single dense MLP. A classic conception, given the name "experts" is that experts specialize in a subject much the way a human expert would. We seek to mechanistically explore Mixture of Expert models, adopting techniques from the mechanistic interpretability literature in an attempt to understand what experts actually learn, and to what degree they specialize. We present a novel feature based explanation of MoEs performance and conduct 3 experiments to address various subquestions on Mixture of Expert specialization. We find that experts do specialize, they often specialize along lines intuitive to humans, and that the representational complexity of an input affects the degree to which experts specialize.

1. Introduction

Mixture of Experts (MoE) is an architecture used in some of the most prominent large-language models today. GPT-4 is rumored to use a Mixture of Experts (MoE) model featuring 8 220B parameter experts, Mistral 8x7B uses a MoE model with 8 experts, and DataBricks recently announced a 16-expert MoE model [8, 16]. MoE models replace a standard dense feed-forward layer, where inputs are passed through the same MLP with multiple MLPs, called experts [6]. Each input is routed to K of these experts, where K denotes the number of experts each input passes through [6]. If the inputs are tokens as in the NLP case, this looks like feeding each token through K of N different MLPs, instead of all through the same MLP. In our case, we work with images and so this looks like feeding each image through K of N different MLPs. In order to determine which of the N experts each token is routed through, the MoE features a router which is a learned linear layer that projects each image to an N-dimensional vector (again, N is the number of experts) [18]. Then we take the softmax of this vector and select the K highest valued experts to route each image to. MoEs are a powerful architecture that allow us to increase the parameter to compute ratio in our model, enabling us to build more expressive, powerful models without increasing compute [7].

One question that remains unanswered, however, is what these experts actually learn. Is each expert in an MoE layer a replica with the same set of parameters such that which expert each input is routed to essentially makes no difference—that is MoEs would be essentially useless. Or, do experts each contain a unique set of parameters (which compose a unique set of features) and therefore each expert maps input to output differently. If so, we say that experts specialize and then seek to understand what these unique sets of features that each expert learns look like, and what inputs are they best suited to—in other words what do the experts specialize in? Is it the case that experts specialize in high-level subjects like "animals" or "vehicles" or is it the case that while experts do learn unique sets of features, they don't specialize in these high-level topics that map onto how a human might group the inputs. We finally seek to explore how the representational complexity of an input sequence to an MoE layer affects the ability of an expert to specialize.

By exploring what experts in an MoE layer do, addressing the above questions, and building up a mechanistic understanding of MoE, we will be able to develop better MoE-based architectures that capitalize on this new understanding; optimizing the data fed to the model, the number of experts in an MoE layer, the position of MoE layers within the model, and so on, to maximize the expressiveness of these models. Understanding MoE is also a crucial step to develop alignment strategies for the various current and future LLMs using MoE.

In what follows we train a series of toy MoE models on image data and suggest that experts do specialize, although per-expert specialization decreases as we increase the number of experts in an MoE layer. Often, this specialization is along high-level lines. We also find that MoE's positioned earlier in the model's architecture depth-wise are less able to specialize which we argue arises from the fact that the input earlier in the model is less representationally complex

which negatively affects the routers ability to discern what expert to send an input to and therefore expert's need to be more similar.

2. Related Work

Our work explores the Mixture of Experts architecture, introduced by Shazeer *et al*. [15]. In this work, the authors present MoE as a form of conditional computation that allows us to increase the model's capacity without proportionally increasing required compute [15]. Specifically, rather than only use a single dense MLP, we use multiple of these MLPs, each called an "expert", and maintain equivalent compute cost by only activating a single MLP for any given token (of course, this is a simplified view and we often activate more than 1 MLP). We adopt this view of MoE, increasing model size without increasing compute, as opposed to the equally viable alternative presented by Gale *et al*. [6]; here, MoE is seen as taking a large dense model, and sparsifying the model, only passing input through a certain subset of the MLPs, reducing compute while maintaining model capacity [6].

Lepikhin *et al*. and Fedus *et al*., introduce the idea of a load-balancing loss. used to a roughly equal degree [10, 5]. Specifically, when training MoE models, the percentage of inputs routed to each expert is almost always unequal due to the random initialization of the model; in practice, a select few experts receive virtually all inputs [10]. This is obviously sub-optimal as the other experts are essentially wasted parameters. In order to overcome this training challenge, Lepikhin *et al*. introduce the idea of using an auxiliary loss which combines with the cross-entropy loss to produce our final loss [10]. We adopt the specific auxiliary loss presented by Fedus *et al*., where the auxiliary loss is a dot-product between the number of tokens routed to each expert and the mean-probability a given token is routed to each expert [5]. The loss is minimzed when every value in both vectors is $\frac{1}{N}$ where N represents the number of experts in the MoE layer. They scale the auxiliary loss such that it remains constant regardless of number of experts and then add it to the cross-entropy loss with a weighting term such that it doesn't overly influence training [5].

In addition to existing research on MoEs, our exploration builds on recent work in mechanistic interpretability which seeks to understand how models map input to output by analyzing low-level elements of the model (individual neurons, combinations of neurons etc.). Specifically, we adopt the concept of features as the fundamental unit which combine to map input to output, introduced by Olah *et al*. [1]. Features are the base building-block of Neural Networks, are combosed by the model's parameters and are represented by the model's neurons. When a neuron activates this is Features combine with one another to form more complex circuits that are able to interact with inputs, and detect key attributes that ultimately allow the model to classify the input *et al*. [12, 14]. We view MoEs as powerful because, by increasing the number of experts and therefore the number of neurons the model can represent, we are also increasing the total number of features the model can represent.

A key aspect of mechanistic interpretability research is training models that are quick to train, easy to explore [3]. Large models are incredibly intensive to train from scratch, making minor ablations, repeated trials, and multiple experiments not viable [13]. Instead, we follow the lead of Elhage *et al*. who use small toy models to explore superposition within neurons in an interpretable manner [3]. Our experiments similarly rely on using small toy models, and small datasets, to explore MoE. This allows us to examine individual layers in great-depth to understand whether and how experts are specializing on a given input, easily ablate or modify the architecture to test how this impacts specialization, and run hundreds of trials.

Much of the early mechanistic-intepretability work was focused on understanding convolutional layers in CNN image models, and attention layers in transformers [11, 4]. MLP layers were seen as particualrly difficult to work with given the presence of polysemantic neurons and the superposition theory. Recent work, however, has begun exploring MLP layers. Templeton *et al*., for example, explore Claude-3 Sonnet and extract interpretable features from the neurons within the model's MLPs [17]. We are unaware, however, of any formal explorations of the MLPs (experts) in MoE models, however, and therefore believe our exploration of MLPs in MoE layers, experts, and their specialization (interpreted through the feature lens), is a novel one.

3. Methods

We adopt the view that MoEs allow us to increase the parameters in our models, and therefore the number of neurons in our model, without increasing the theoretical compute, and build on this view below [15]. Specifically, when we replace a standard feed-forward layer with an MoE layer, instead of a single MLP with x parameters we now have N MLPs with x parameters. Crucially, every input still only interacts with x parameters as opposed to all Nx parameters the model contains (at least when $K = 1$), maintaining consistent compute cost. So why does increasing parameters (sparsely) without increasing compute and without increasing the parameters each input interacts with, improve a model's representational power? We frame the power of MoEs as an improvement through the lens of features. Specifically, to map input to output, a neural net maps the inputs through a series of features which activate to varying degrees. The degree to which a feature activates shapes the output for the next layer which again triggers a series of features and so on until finally reaching the final output. In simple terms, consider an image of a cat—this might

look like first triggering the "small feature", followed by the "eyes feature", followed by the "fur feature", and so on, each of which modifies the output such that collectively they modify the output in a way that the final linear layer now projects it to the "cat" output. Obviously, the more features a neural network can represent the more successfully it can distinguish between inputs and the more successfully it can map each input to its correct output. A neural net only able to represent the features "fur" and "small" is going to struggle to correctly classify whether an image is a car or a truck. Through this lens, we see why MoE is such a powerful architecture. Say, a single MLP can represent y features. If we add an additional MLP, we should be able to represent y more features and now have $2y$ features. Crucially, to maintain equivalent compute we are only able to send each input through one of the two experts. So, each input still only interacts with y features, but because the model contains two experts with two different sets of y features, we can, for each input, select which of the two experts's set of y features is best suited to the input. Naturally, then when we add a second expert, shifting our architecture from dense to MoE, so long as a single token is better suited to the features represented by the second expert, and so long as the router can send this token to the second expert, we should be able to achieve theoretically better performance. We introduce this fairly novel understanding of MoE and rely on it as we conduct our experiments and analysis.

Our goal is to develop a better understanding of the experts in a MoE model and what features they learn, particularly in relation to one another. This then, looks like a broad exploration of the unique features experts learn (and if they learn unique features) which we call specialization. We further divide this into 3 sub-questions and develop methods to answer each question which we detail here.

First, do experts specialize at all? Here, we're interested in whether each expert is unique and whether experts learn truly distinct sets of features. Presumably, if they do, these distinct sets of features are designed such that each maps some set of inputs to their correct output better than other experts but the set of inputs is not necessarily a high-level subject like "cats". To explore this we train a series of toy models and ablate the learned router with a random router. The idea behind this is that if experts specialize at all—that is have different features—for many inputs in the model there should be some optimal set of features and therefore an optimal expert that can best map the image to it's correct output (or to an output representation that the final linear layer can map to the correct output). A learned router will attempt to route each input to the optimal expert and while it won't be perfect it should route experts to their optimal expert more frequently than random chance. And so, if we replace the learned router with a random router that does simply route tokens to experts randomly, and experts do specialize—have unique features—we should expect to see a drop in accuracy. Moreover, the degree to which accuracy drops when we ablate the router in some model is a very strong indicator of how much experts in that model specialize.

Second, do experts specialize in highly-interpretable subjects (e.g. one expert specializes in vehicles while another specializes in animals)? A common view of MoE is that each expert would specialize in some broad category of input and would therefore learn the optimal set of features to map this input to output. This seems like a rational approach for humans to take but it's not clear that a model, if it does specialize, does so in this way. For a dataset consisting of images of animals and vehicles (i.e. CIFAR-10) it seems plausible that experts break down along these lines but also perfectly plausible that one expert specializes in images with some combination of colors in the top-left. This is an important question to address because if experts don't specialize in high-level subjects, interpreting MoE models and therefore aligning them and improving them through the lens of interpretability results becomes much more difficult. It's much easier to analyze MoE models and discern why they output the values they do, if we can predict, purely off of the input, the precise set of experts the input will flow through. To address this question we train a series of toy models, and explore the breakdown of expert routing given an image's output class.

Finally, how does the input's representational complexity affect the degree to which experts specialize? In order for experts to specialize and learn different sets of features, which are best suited to some set of inputs, the router needs to be able to discern which inputs are in this set and therefore route these inputs to that expert. When experts specialize in high-level subjects this looks basically like a classification problem; if one expert specializes in animals, the router has to determine whether a given image is an animal or not in order to determine whether it should route the image to that expert. The resemblance of routing and classification prompts us to explore how the representational complexity of the input to an MoE layer impacts expert specialization. Specifically, the ability of a model to classify the input prior to any of the convolutional layers, or after it passes through the first convolutional layer is significantly lower than it's ability to classify the input after passing through a series of convolutional layers. Early on, depthwise, in the model the representation isn't sufficiently built up such that the model could classify it well—each subsequent layer, instead, builds up this representation until classification is more feasible. Since we view routing as similar to classification we expect that routing inputs to a specific expert is more challenging earlier on in the model. If routing is difficult and inputs often get routed to the "wrong" expert it seems reasonable that experts have to learn very

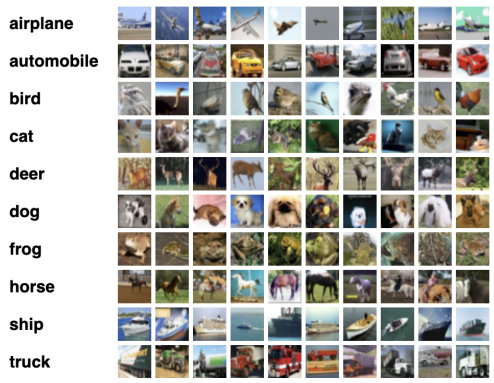

Figure 1. CIFAR-10 Dataset

similar features such that incorrect routing is less costly as all experts broadly share the same features and can roughly handle all inputs. It seems likely, then, that MoE layers in earlier layers in the model have experts with largely overlapping features to account for the fact that routing inputs to the "correct", optimal expert is more difficult as the input sequence isn't representationally complex (equivalent term here is representationally rich) enough. We evaluate this by shifting the location of the MoE layer depth-wise in the architecture and comparing the level of specialization.

Some of our training loop and evaluation code is based on [2]

4. Dataset and Features

We primarily use the CIFAR-10 dataset for all our experiments [9]. CIFAR-10 contains 50,000 training examples and 10,000 test examples. We divide the 50,000 training examples into 49,000 training images and 1,000 validation images. Each image is 32x32x3 and belongs to one of 10 classes: plane, car, bird, cat, deer, dog, frog, horse, ship, and truck. We normalize our data giving the dataset a mean of 0 and a standard deviation of 1.

5. Results

Our base model architecture which forms the foundation of all our experiments uses 4 convolutional layers each with a stride of 1 and kernel size of 3. Each convolution is followed by a ReLU activation function. The first convolution has 128 filters and uses a 2x2 Max Pool, the second has 256 filters and no pooling layer, the third also has 256 filters with a 2x2 Max Pool, and the fourth has 512 filters with a 2x2 Max Pool. We follow these 4 convolutional layers with a 4x4 Max Pool. We have a final linear layer that projects our 512 dimensional sequence post-convolution and pooling to our output logits. Unless otherwise specified (such as in experiments testing the impact of MoE layer location depth-wise on specialization), we insert our MoE layer after the four convolutional layers and 4x4 pooling layer and before the final linear layer.

We train our model with the AdamW optimizer and the OneCycleLR learning rate scheduler. We set our max learning rate to 1e-2 and our weight decay to 1e-4. Across all experiments and all trials we train for 10 epochs and achieve a final test loss (at least for the base model) between 86% and 87.5%. Crucially, we are not optimizing for accuracy, we just want a relatively high-performing model.

We use the auxiliary loss described in Fedus et al., 2022 with a weight of 1e-5, meaning our total loss is the cross-entropy loss + 1e-5 aux loss. Without an auxiliary loss, many experts end up deeply underutilized, with the vast majority of images clustering in only some of the experts. We compare different weightings for the auxiliary loss, testing values between 1e-2, and 1e-7. We find 1e-5 optimal in terms of not hurting loss while also ensuring equal expert utilization. This differs from the standard weighting of 1e-2. We suspect this is because our CE loss is much lower than with a standard model on normal data; training on CIFAR-10, which is such a small dataset, results in our accuracy being very high. To account for this we need our auxiliary loss to also be lower than normal so that we maintain the same ratio between it and the CE loss.

5.1. Experiment 1

We first evaluate whether experts learn unique features at all; that is, do they specialize? We train a base model, which consists of 4 convolutional layers followed by our MoE layer and a final linear layer that outputs our class logits. In order to evaluate expert specialization we replace the learned router with a random router, which, for each token randomly selects one of the N experts to send the token through.

We train three different versions of this model, the first with 2 experts, the second with 3 experts and the third with 5 experts. For all three versions we use K=1: each image is sent to only one expert. For each version, we train the model 10 times (each time starting from scratch) and evaluate the clustering of classes with regard to the experts.

We find that across all 3 versions, accuracy drops when we replace the learned router with our random router, but not significantly. Specifically, averaged over 10 trials, accuracy in the 2-expert model drops by 3.46% on average, accuracy in the 3-expert model drops by 1.5% on average, and accuracy in the 5-expert model drops by 0.0066 on average. Still, in all 3 versions, across all 10 trials accuracy remains above 81%. Note, that simply replacing the experts with randomly initialized experts drops accuracy to around 10%: random chance.

The accuracy drop when we route tokens to random experts as opposed to their ideal expert (as determined by the router) suggests that experts do specialize—each learning

Experiment 2	2-Experts	3-Experts	5-Experts
Learned	86.77%	86.78%	86.03%
Random	83.77%	85.43%	85.57%
% Decrease	3.46%	1.56%	0.535%

Table 1. Test-set accuracy with learned router versus random routing for 2, 3, and 5-expert MoEs

a unique set of features that are best suited to some set of inputs; when tokens are routed to another expert they interact with a less optimal set of features and the model misclassifies the input more frequently. However, the fact that the accuracy drop is relatively minor gives us strong evidence that experts, despite specializing, still share the vast majority of their features. In other words, the features in an expert tailored specifically to the set of inputs the expert specializes in is a small fraction of their overall feature set; most features are general shared features required for mapping any input to output not just the specialized inputs. We suggest the strong overlap in features across experts is the result of two factors. First, in general, mapping animal images to their correct category and mapping vehicle images to the category will generally require many of the same tools: curve detectors, color detectors etc [**?**, 12]. While many of these features might be present in the prior convolutional layers, the MLP of course also has features and these features are still tasked with mapping input to output. And because the MLPs are another component which helps map images to classes and because regardless of input much of this task is the same, whatever features the MLPs do have then, are likely going to be similar across experts. Naturally, then, even experts designed only to handle the inputs they specialize in would still share lots of features. Second, the router isn't perfect. Routing (as we discuss in experiment 3) is difficult and the router often sends images to a suboptimal expert. In order to ensure that this frequent occurrence doesn't hamper loss significantly, experts need significant overlap so they can still perform reasonably well on inputs they don't "specialize" in.

The fact that the accuracy drop is largest for the 2-expert model, followed by the 3-expert model, with the lowest accuracy drop for the 5-expert model suggests that the fewer experts we have the more experts specialize (a finding we also discuss with reference to high-level specialization). We believe this is because it's more difficult to discern which of 5 categories an input falls into as opposed to which of 2 categories the input falls into; this makes routing more difficult and error-prone as the number of experts increases and forces experts to specialize less and have more overlapping features to handle poorly-routed inputs. Note that this doesn't mean more experts are worse. While more features are shared across experts, we also have more features in total and so it's reasonable that while an expert in a 5-expert MoE adds fewer marginal features to the model than an expert in a 2-expert MoE, collectively the 5 experts still represent more total features than the 2 experts.

The implications of this are important, however. If a dense feedforward layer's MLP has N parameters, and x features, a second MLP (i.e. shifting the layer to a sparse-gated-MoE) increases the parameter count to $2N$, but it doesn't increase the feature count—the representational power of the model—to $2x$ Increasing parameter count sparsely (as in MoE) does generally increase compute time as while actual computation remains the same, memory costs are higher. So, given that the marginal increase in additional features decreases as we add more experts, and memory costs still increase linearly, we suggest that MoE models with tons of experts are likely sub-optimal. Where this line is, represents an important next step in MoE research.

5.2. Experiment 2

In experiment 1, we saw that experts specialize, at least to some degree. We now explore whether MoE layers specialize along high-level interpretable subject lines. Specifically, we analyze how each of the 10 classes in CIFAR-10 are sent through the MoE layers. Here, we're looking not only for interpretable class-based clustering for a single trial (i.e. vehicles to one expert, animals to the other) but crucially whether clustering patterns are consistent and universal across trials. Non-consistency would suggest experts don't really specialize in high-level subjects as the grouping is random; there's no "thought" put into how inputs are grouped. Sometimes an expert specializes in "cars" and "trucks" while other times it specializes in "cars" and "horses". This would make for a much less convincing case that experts learn high-level interpretable subjects (like vehicles vs animals). Instead, it would only suggest that the router does send things to experts based on their class but that the router does not group classes in an intetional way and doesn't exploit intuitive human groupings of classes..

We again conduct 3 versions of the experiment, one with 2 experts, one with 3 experts, and one with 5 experts (K=1), in order to see how general specialization varies by number of experts. We again train each version for 10 trials.

We first analyze the results of the 2-expert model. Across all 10 trials we notice striking clustering patterns. Specifically, in each trial, over 88.5% of plane, ship, car, and truck images end up in one expert while the vast majority of the remaining classes, the animal classes (cat, dog, deer, horse, frog, and bird), end up in the other expert. This breaks down into by far the most obvious human-interpretable bifurcation of the classes: vehicles versus. animals. The fact that this expert split is consistent across all our trials provides strong evidence that, at least when possible, experts do spe-

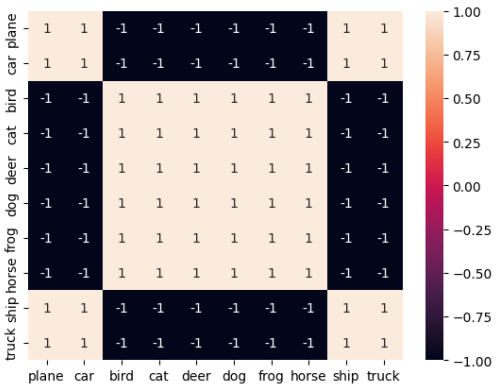

Figure 2. Coefficient matrix of 2-expert Model

Class	Expert 1	Expert 2
Plane	96.3%	3.7%
Car	95.7%	4.3%
Bird	31.7%	68.3%
Cat	17.6%	82.4%
Deer	11.3%	88.7%
Dog	24.2%	75.8%
Frog	3.2%	96.8%
Horse	32.8%	67.2%
Ship	96.5%	3.5%
Truck	93.7%	6.3%

Table 2. Class breakdown of expert routing in 2-expert MoE (averaged across 10 trials)

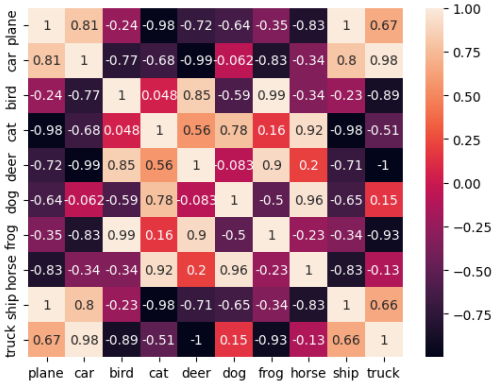

Figure 3. Coefficient matrix of 3-expert Model

cialize in high-level subjects.

We next train the same model with 3 experts instead of 2. Here, we notice that images of planes, cars, trucks, and ships are typically routed to the same expert with high-confidence.

Images of dogs, cats, and generally horses usually share another expert, though with less confidence and images of

Class	Expert 1	Expert 2	Expert 3
Plane	80.7%	17.8%	1.5%
Car	60.5%	8.9%	30.6%
Bird	28.5%	44.6%	26.9%
Cat	10.0%	36.6%	53.4%
Deer	6.1%	62.3%	31.6%
Dog	12.8%	10.6%	76.6%
Frog	2.4%	92.1%	5.5%
Horse	12.7%	25.6%	61.7%
Ship	70.6%	21.9%	7.9%
Truck	58.6%	3.7%	37.2%

Table 3. Class breakdown of expert routing in 3-expert MoE (averaged across 10 trials)

birds, deer, and frogs are also generally classed together. Compared to 2 experts, the routing is less consistent, but the general trend holds across the majority of trials. The routing here appears to still be broadly following high-level interpretable lines. Vehicles are classified together and then animals are divided into two groups. That dogs and cats are routed together is similarly unsurprising. Birds, deer, and frogs representing the third and final group may seem odd, however. We qualitatively explore the images from these classes to make sense of the groupings. When we explore the images of these animals in CIFAR-10 we find that frogs and deer have similar images with frequent green–brown backgrounds, and the animals themselves are both typically light brown with deer antlers somewhat resembling the body shape of a frog. Birds, meanwhile, are small and often brown and therefore seem to resemble a frog when looked at from a distance. Dogs and cats tend to have a distinctive body shape and similar texture given their fur, and many of the images have details of their faces, which distinguishes them from other animals. Horses may seem more similar to deer here, but we notice that while horses generally cluster with dogs and cats, this is a weaker correlation and that light brown horses (particularly in fields) often cluster with the deer/frog/bird class. This analysis suggests once again, that, when possible, experts are routing based on general concepts that map onto high-level human subjects like animal body shape and background setting. Insofar as experts fail to group by human discernable subject, occasionally sending a truck image to the frog class, for example—this seems to be a function of lacking the necessary detail to discern the difference between the two as opposed to a desired, intentional grouping.

Finally, we expand to 5 experts and notice that clustering by interpretable high-level subjects has almost completely broken down; no single expert gets a large percentage of some intuitive group. Still, the correlation matrix shows positive relationships between vehicle images and between

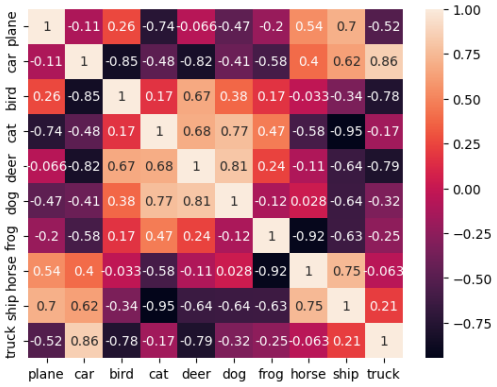

Figure 4. Coefficient matrix of 5-expert Model

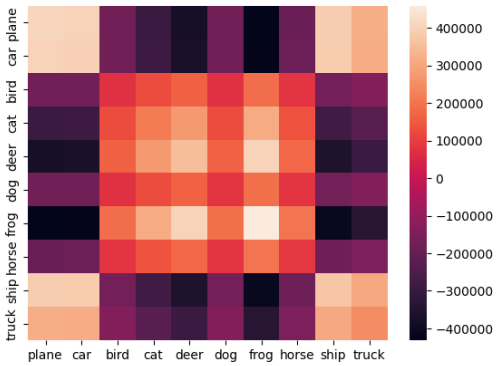

Figure 5. Covariance matrix of 2-expert Base Model

images of dogs and cats which map onto intuitive high-level subjects. While for all trials, we see at least some grouping by high-level subject, there's a clear and sharp decline in the strength of these groupings as we increase the number of experts. We suggest this again ties into the difficulty of routing discussed in experiment 1. As we increase the number of experts, the human-intuitive high-level groupings become more fine-grained—instead of vehicles vs. animals, it's now dogs and cats, vs small animals like birds and frogs, vs large mammals like horses and deer, and so on. The problem is the router needs to be able to discern whether an image is in group 1 or 2 or 3, and so on,in order to learn such a grouping. Crucially, distinguishing whether an image is a vehicle or an animal is easier than distinguishing whether an image is of a deer or a horse. It's unsurprising then that not only do experts specialize less as we increase the number of experts, they also specialize less along high-level lines.

5.3. Experiment 3

We finally wanted to explore how the representational complexity (equivalently, representational richness) of the input sequence to the MoE influences specialization. We view the model as gradually building up the representational complexity of the input; taking a raw image, and gradually modifying it, through both the first 4 convolutional layers, as well as the MLP in the MoE layer, to contain important semantic information that allows the final linear layer to classify the input correctly. In this sense, then, the deeper an input is within the model, the more the modified input contains necessary information for classification. We discuss above the similarities between classification and routing. We have also, suggested, based on our analysis of the results from our 1st experiment, that additional experts result in reduced specialization-per-expert because the router is more error-prone when forced to distinguish between 5 groups compared to, say, 2 groups. Given this, we expect that in the same way that additional experts increase router-diffculty and reduce specialization, lack of representational complexity in the input increases router-difficulty and might also reduce specialization. To test this, we compare the specialization of experts in MoE models located earlier within the model's depth, compared to later within the model's depth. Specifically, we train a short model where we modify the base model to only use the first convolutional layer (which features a 2x2 Max Pool and 128 filters), and follow it with an 8x8 Max Pool before flattening the 512-dimensional input and passing it to the MoE layer which is followed by the final linear layer. We compare this to our base model which, as mentioned above, features 4 convolutional layers, a 4x4 pool, the MoE layer and then the final linear layer. We use the 8x8 pool in the short model to ensure that the flattened input to the MoE in both models is 512-dimensional; this ensures we don't make the router have to handle a larger input in one model and therefore impact the routing ability in an unintended manner. We compare expert specialization by analyzing the class-breakdown of expert routing (as in Experiment 2).

We find that human-interpretable specialization is significantly reduced in MoEs located in earlier layers. We compare the covariance matrix of the short-model's class-based expert routing to the base model's covariance matrix. We use the covariance matrix here because it better highlights the distinction in specialization between the two experts (correlation, by normalizing everything to $[-1, 1]$ results in all values being 1 or -1 in both matrices despite significant differences in covariance).

We don't compare results using random ablation (experiment 1 technique). The MoE in the shorter model represents a larger proportion of total model parameters (since it only has 1 convolutional layer compared to the base model's 4). We would therefore expect it to have a larger impact on loss. So, if ablating the learned router with a random router results in a larger percentage drop in loss compared to the base model this doesn't indicate greater specialization but is likely just a function of the layer being more important;

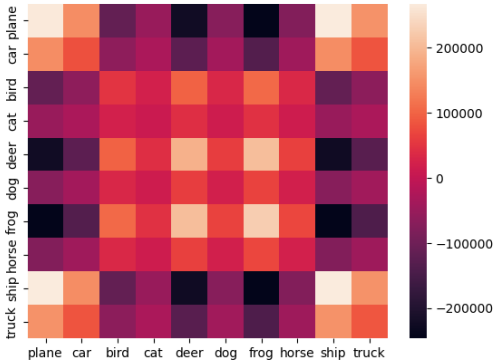

Figure 6. Covariance matrix of 2-expert Short Model

sub-optimal routing would therefore have a greater impact on loss in the short model compared to the base model for equivalently specialized experts.

These results, which demonstrate less high-level specialization among experts earlier in the model, suggest that the representational complexity of the input sequence influence the ability of the router to successfully gate inputs, in the same way representational complexity influences the ability of a model to classify an input; classification would have much lower accuracy if attempted after the first layer. In turn, to make up for less precise, routing, expert's specialize less and share more overlapping features. When combined with the findings from experiment 1——that increasing the number of experts in a layer reduces the expert's specialization——which we also theorized occurs due to routing difficulty, we believe this is strong and compelling evidence for this hypothesis; that representational complexity of input increases specialization. An alternative, viable, counter-argument is that routing along high-level intepretable subjects is similar to classification and so experts in early layers struggle to do this, but general specialization allows for experts to specialize in lower-level concepts (color of the top-left of an image). It seems reasonable that even while the input sequence isn't particularly representationally rich, the router is still able to correctly discern whether a given input contains one of these lower-level, simpler features, even if discerning whether an image is of a large, furry, mammal is too difficult. That is, specialization may still occur, just not high-level specialization. We believe exploring this question in further depth requires a neuron-based analysis and represents an important and natural next step for our work.

6. Conclusion

We conduct a novel exploration of MoE models, presenting a feature-based explanation of MoEs power, mechanistically interpreting their experts and exploring key aspects of how they work. We find that experts specialize, and do so along high-level human interpretable lines; that increasing the number of experts decreases expert specialization likely due to difficulty routing; and that less representationally rich experts specialize along high-level lines less. Our findings help breakdown how MoE models behave, why they perform well, and have significant implications for training future MoE models. For example, that experts specialize less (at least along high-level lines) suggests that training hierarchical MoEs where the number of experts in an MoE layer increases depth-wise through the model as the input's representational complexity is gradually built up is a reasonable idea.

Further research is needed on three key fronts. First, expanding our results to analyze larger, more complex models and seeing if the same findings hold—while we expect the general, core ideas (like representational complexity of the input to an MoE layer positively impacts specialization) we need to verify this and also see whether other, tangential details differ in larger models. Second, exploring whether only high-level specialization or all specialization increases as representational complexity increases. And third, verifying whether the results we find here when focused on image models hold in the NLP space. MoEs are generally used in language-based models where the inputs are tokens representing words, sub-words, or characters. Ensuring these findings hold could have significant implications for future LLMs which are increasingly adopting the MoE architecture.

7. Contributions & Acknowledgements

James Poetzscher, as the sole author, was responsible for the entire project.

References

[1] L. S. Chris Olah, Alexander Mordvintsev. Feature visualization, 2017.

[2] CS231N. Assignment 2 part 5, 2024.

[3] N. Elhage, T. Hume, C. Olsson, N. Schiefer, T. Henighan, S. Kravec, Z. Hatfield-Dodds, R. Lasenby, D. Drain, C. Chen, R. Grosse, S. McCandlish, J. Kaplan, D. Amodei, M. Wattenberg, and C. Olah. Toy models of superposition, 2022.

[4] N. Elhage, N. Nanda, C. Olsson, T. Henighan, N. Joseph, B. Mann, A. Askell, Y. Bai, A. Chen, T. Conerly, N. DasSarma, D. Drain, D. Ganguli, Z. Hatfield-Dodds, D. Hernandez, A. Jones, J. Kernion, L. Lovitt, K. Ndousse, D. Amodei, T. Brown, J. Clark, J. Kaplan, S. McCandlish, and C. Olah. A mathematical framework for transformer circuits, 2021.

[5] W. Fedus, B. Zoph, and N. Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity, 2022.

[6] T. Gale, D. Narayanan, C. Young, and M. Zaharia. Megablocks: Efficient sparse training with mixture-of-experts, 2022.

[7] C. Hwang, W. Cui, Y. Xiong, Z. Yang, Z. Liu, H. Hu, Z. Wang, R. Salas, J. Jose, P. Ram, J. Chau, P. Cheng, F. Yang, M. Yang, and Y. Xiong. Tutel: Adaptive mixture-of-experts at scale, 2023.

[8] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. de las Casas, E. B. Hanna, F. Bressand, G. Lengyel, G. Bour, G. Lample, L. R. Lavaud, L. Saulnier, M.-A. Lachaux, P. Stock, S. Subramanian, S. Yang, S. Antoniak, T. L. Scao, T. Gervet, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed. Mixtral of experts, 2024.

[9] A. Krizhevsky. Learning multiple layers of features from tiny images, 2009.

[10] D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen. Gshard: Scaling giant models with conditional computation and automatic sharding, 2020.

[11] C. Olah, N. Cammarata, L. Schubert, G. Goh, M. Petrov, and S. Carter. An overview of early vision in inceptionv1, 2020.

[12] C. Olah, N. Cammarata, L. Schubert, G. Goh, M. Petrov, and S. Carter. Zoom in: An introduction to circuits, 2020.

[13] C. Olsson, N. Elhage, N. Nanda, N. Joseph, N. DasSarma, T. Henighan, B. Mann, A. Askell, Y. Bai, A. Chen, T. Conerly, D. Drain, D. Ganguli, Z. Hatfield-Dodds, D. Hernandez, S. Johnston, A. Jones, J. Kernion, L. Lovitt, K. Ndousse, D. Amodei, T. Brown, J. Clark, J. Kaplan, S. McCandlish, and C. Olah. In-context learning and induction heads, 2022.

[14] L. Schubert, C. Voss, N. Cammarata, G. Goh, and C. Olah. High-low frequency detectors, 2021.

[15] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer, 2017.

[16] T. M. R. Team. Introducing dbrx: A new state-of-the-art open llm, 2024.

[17] A. Templeton, T. Conerly, J. Marcus, J. Lindsey, T. Bricken, B. Chen, A. Pearce, C. Citro, E. Ameisen, A. Jones, H. Cunningham, N. L. Turner, C. McDougall, M. MacDiarmid, C. D. Freeman, T. R. Sumers, E. Rees, J. Batson, A. Jermyn, S. Carter, C. Olah, and T. Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet, 2024.

[18] B. Zoph, I. Bello, S. Kumar, N. Du, Y. Huang, J. Dean, N. Shazeer, and W. Fedus. St-moe: Designing stable and transferable sparse expert models, 2022.