Sparsely-gated MoE Layers for CNN Interpretability

Svetlana Pavlitskaya\textsuperscript{1}, Christian Hubschneider\textsuperscript{1}, Lukas Struppek\textsuperscript{2}, and J. Marius Zöllner\textsuperscript{1,2}
\textsuperscript{1}FZI Research Center for Information Technology
\textsuperscript{2}Karlsruhe Institute of Technology
\textsuperscript{pavlitskaya@fzi.de}

Abstract

Sparsely-gated Mixture of Expert (MoE) layers have been recently successfully applied for scaling large transformers, especially for language modeling tasks. An intriguing side effect of sparse MoE layers is that they convey inherent interpretability to a model via natural expert specialization. In this work, we apply sparse MoE layers to CNNs for computer vision tasks and analyze the resulting effect on model interpretability. To stabilize MoE training, we present both soft and hard constraint-based approaches. With hard constraints, the weights of certain experts are allowed to become zero, while soft constraints balance the contribution of experts with an additional auxiliary loss. As a result, soft constraints handle expert utilization better and support the expert specialization process, while hard constraints maintain more generalized experts and increase overall model performance. Our findings demonstrate that experts can implicitly focus on individual sub-domains of the input space. For example, experts trained for CIFAR-100 image classification specialize in recognizing different domains such as flowers or animals without previous data clustering. Experiments with RetinaNet and the COCO dataset further indicate that object detection experts can also specialize in detecting objects of distinct sizes.

1. Introduction

Sparse MoE layers have recently gained popularity thanks to their ability to scale up models to billions and lately even trillions of parameters\cite{2,4,21}. The focus, however, was almost exclusively on transformer models for language modeling tasks. In this work, we insert MoE layers into convolutional neural networks (CNNs) and apply the approach to the basic computer vision tasks of image classification and object detection. To tackle a well-known problem of unstable expert training, we present soft and hard constraints, encouraging balanced expert utilization. The models further provide an additional hyperparameter to adjust the number of active experts in each forward pass and, thereby, the computational complexity.

Inherent model interpretability is one side effect of embedding MoE layers into model architectures. For the language modeling tasks, the experts were shown to mostly specialize on shallow concepts\cite{10,14,21}. All previous works rely on transformers. To the best of our knowledge, we are the first to study the impact of sparse MoE layers, embedded in CNNs, on model interpretability in computer vision tasks. Due to the larger receptive field, experts can focus on more high-level semantic concepts.

Our contributions can be summarized as follows:

\begin{itemize}
  \item we apply the concept of sparse MoE layers, primarily used in transformers so far, to CNNs,
  \item we analyze the semantics of the learned experts and evaluate the impact of different constraints for balancing load specialization on the interpretability strength of the model,
  \item we evaluate the concept on two separate tasks: image
\end{itemize}
2. Related Work

2.1. Sparse MoEs

Classic Mixture of Expert models, introduced by Jacobs et al. [7], contain a variable number of expert models and a single gate to combine the expert outputs. Eigen et al. [3] developed the idea of using MoEs as subcomponents of models with individually learned gates. This allows for sharing the remaining parts of architectures and enables multiple MoE layers within a single architecture. The transition to conditional computing through sparse expert activations was first explored by Shazeer et al. [18] using LSTM networks. By activating a fixed number of experts, it was possible to decouple parameter counts from FLOPs required for inference through MoEs.

Fedus et al. [4] took a further step towards sparse activations and showed that they can train transformer-based MoE models while activating only a single expert, resulting in MoE layers that add only little compute overhead. Several works [15,17] have since explored the single-expert regime, and Zoph et al. [21] derived guidelines to design sparse MoE effectively using transformers.

Sparse MoEs have gained popularity for massive language models [1,9,18]. Performance boosts, achieved via sparse MoEs, make training these models possible, although only on large clusters of powerful GPUs. In computer vision, the application of sparse MoE has centered around transformer models [15,16,20]. In contrast, in our work, we consider CNNs, which is still the most widely spread architecture in the computer vision area.

2.2. Balancing Expert Utilization

A known MoE problem is the focus of gates on a small subset of all available experts. The weights assigned to other experts are permanently zero or negligibly small. Because some experts start to perform better in the first iterations, the gate increases their probability to be activated. Consequently, these few experts improve above average, and the gate assigns even higher weights to them. This self-reinforcing process continues, s.t. the optimizer ends up in a local minimum [3].

Eigen et al. [3] propose a hard constraint on the relative gating assignments to each expert applied during training. For this, the weights assigned to each expert are summed over all training samples. If this value surpasses the average gating assignment by some threshold, the weights for the corresponding expert are set to zero. The remaining positive weights are recomputed using softmax to maintain a convex combination of the experts.

Shazeer et al. [18] presented a soft constraint approach introducing an auxiliary importance loss (see below), which encourages equal importance for all experts during training. The number of training samples per expert may still vary since importance is used instead of the mean number of samples per expert.

2.3. Interpretability via Sparse MoEs

In the early experiments by Shazeer et al. [18] on the tasks of language modeling, the experts specialized on syntax and/or semantics. The authors provide examples of the specialization for three selected experts: one expert was used for the word innovation, another one for the article a, and another one on the concept of fast, rapid action.

Lewis et al. [10] have shown that experts specialize in very local syntactic information: experts learned clusters of numbers, abbreviations, possessive pronouns, etc. No specialization at the semantic level was observed.

Similar behavior was described in the later work by Zoph et al. [21]. The experts were found to specialize in punctuation, articles, conjunctions, proper nouns, and numbers. Interestingly, even in the case of multilingual sparse models, no specialization in languages was observed. Instead, the experts continued to focus on the same shallow concepts like punctuation, articles, or numbers. Mustafa et al. [14] has explored multimodal sparsely-activated models. The text experts specialized in nouns and adjectives, whereas image experts specialized on semantic concepts like body parts, textures, fauna, food, and doors.

The work on visual sparse transformers by Riquelme et al. [16] is closest to ours, because it deals with image data. Here, the experts specialize in discriminating between small sets of classes. Expert-class correlation is strong only for the last few layers, whereas no expert specialization was observed for early layers. Further work with visual transformers by Wu et al. [19] has demonstrated expert specialization across ImageNet classes.

In the case of visual transformers, image patches are tokens, routed to experts. In our work, the routing is performed at the level of whole images, which leads to semantic specialization at the image level.

3. Approach

Our approach comprises three components: (1) embedding sparse MoE layers in CNNs, (2) balancing expert utilization via constraints, and (3) expert specialization analysis revealing the gain in CNN interpretability.

3.1. Sparsely-gated MoEs for CNNs

We propose a method to embed sparse MoE layers into CNNs with the goal to achieve computational complexity similar to the baseline in terms of parameters. Without loss of generality, we consider a CNN consisting of residual blocks [6] (see Figure 1). The proposed ResBlock-MoE architecture uses a complete residual block as its expert. The
MoE layer encapsulates multiple replicas of the block that are activated and mixed using a gating module.

We consider two types of gates (see Figure 2): the GAP-FC gate consists of a GAP layer followed by a single fully connected layer, whereas the Conv-GAP-FC additionally contains a convolutional layer, which can use detailed, local information encoded in the input features.

3.2. Constraints to Balance Expert Utilization

Formally, an MoE consists of a set of $N$ experts $E_1, \ldots, E_N$. For a given input $x$, each expert $E_i$ produces an output $e_i(x)$. The gate computes a weight vector $G(x) = [g_1(x), \ldots, g_N(x)]$.

The final MoE output is computed as a weighted sum of the expert outputs: $F_{MoE}(x) = \sum_{i=1}^{N} g_i(x) e_i(x)$.

To measure the utilization of experts, we define an importance vector $I(X) = \sum_{x \in X} G(x)$ for each batch of training samples $X$, and the importance of a single expert $E_i$ as $I_i(X) = \sum_{x \in X} g_i(x)$ [18].

We refer to the problem of unbalanced MoE expert utilization as dying experts, analogous to the dying ReLU [13]. We consider an expert as dead if it receives less than 1% average importance on the test set. To mitigate the problem, we propose one soft and two hard constraints

**Hard Constraints**: motivated by the work of Eigen et al. [3], we propose two hard constraints on importance. Both hard constraints are only active during training and deactivate experts for an entire batch.

We denote the mean importance of batch $X$ as $I(X)$ and define the relative importance of expert $E_i$ for $X$ as follows:

$$I_i^{rel}(X) = \frac{I_i(X) - I(X)}{I(X)}$$ (1)

In the relative importance constraint, the expert weight is zeroed for a batch, if the running relative importance of this expert exceeds the predefined threshold $m_{rel}$:

$$g_i(X_t) = 0 \iff \sum_{t'=1}^{t-1} I_i^{rel}(X_{t'}) > m_{rel}$$ (2)

For the mean importance constraint, we define the mean importance assigned to expert $E_i$ up to time step $t$:

$$I_i(X_t) = \frac{1}{t} \sum_{t'=1}^{t} I_i(X_{t'})$$ (3)

In this constraint, the expert weight is zeroed when the mean importance for this expert exceeds the mean importance of the batch by some predefined threshold $m_{mean}$:

$$g_i(X_{t+1}) = 0 \iff I_i(X_{t+1}) - I(X_t) > m_{mean}$$ (4)

The relative importance constraint takes a stronger focus on the recent past, whereas the mean importance approach takes a holistic view, with all past importance values having the same impact on the constraint.

**Soft Constraints**: the first soft-constrained approach, originally proposed by Shazeer et al. [18], is an auxiliary importance loss $L_{imp}$, following Equation 5. It uses the squared coefficient of variation of importance vector $I(X)$ for batch $X$.

$$L_{imp} = w_{imp} \cdot CV(I(X))^2 = w_{imp} \cdot \left(\frac{\sigma(I(X))}{\mu(I(X))}\right)^2$$ (5)

We propose another soft constraint that takes a probabilistic view on expert importance. For this, we interpret an MoE as a probability model in which class probabilities are marginalized over expert selection. Each weight $g_i(x)$ is thus the probability $p(E_i|x)$ to select a specific expert for a given input, and outputs of each expert $e_i(x)$ quantify the probability $p(c|E_i,x)$ of each class $c \in C$. The MoE output is then defined as follows:

$$F_{MoE}(x) = \sum_{i=1}^{N} p(E_i|x)p(c|E_i,x) = p(c|x)$$ (6)

We interpret the gate output as a discrete probability distribution $P$ with probability $P(E_i|\mathcal{X})$ for expert $E_i$ to be selected for input $\mathcal{X}$ ($\mathcal{X}$ is a random variable for input $x$).

In expectation, the gate should assign each expert $E_i$ the same average weighting, equal to $\mathbb{E}_X [P(E_i|\mathcal{X})] = \frac{1}{N}$. The expected weight assignment thus corresponds to a discrete uniform distribution $Q$ with probability $Q(E_i|\mathcal{X}) = Q(E_i) = \frac{1}{N}$.

We define an auxiliary KL divergence loss $L_{KL}$, as the KL-divergence $D_{KL}$ between $P$ and $Q$, weighted by hyperparameter $w_{KL}$. The probability $P(E_i|\mathcal{X}) = X$ is

$$I_i(X)$$

Figure 2. Gate architectures.
is computed as the average importance per sample in batch $X$. $L_{KL}$ is then defined as follows:

$$L_{KL} = w_{KL} \cdot D_{KL}(P||Q) = w_{KL} \cdot \sum_{i=1}^{N} P(E_i|X) \cdot \ln \left( \frac{P(E_i|X)}{Q(E_i)} \right)$$

$$= w_{KL} \cdot \sum_{i=1}^{N} \frac{I_i(X)}{|X|} \cdot \ln \left( \frac{I_i(X) \cdot N}{|X|} \right)$$

$L_{imp}$ penalizes inequality in importance distribution harder than $L_{KL}$, thus achieving an equal expert utilization. On the other hand, $L_{KL}$ leads to higher variance in the expert utilization but still avoids dying experts.

4. Experiments with Sparse MoEs for Image Classification

4.1. Experimental Setup

We use the ResNet-18 [6] architecture\(^1\) consisting of four ResNet blocks, while the first convolutional and pooling layers were replaced by a single 3x3 convolution. We run experiments on the CIFAR-100 dataset [8]. The MoE experts follow the architecture of the ResNet blocks, but we adjust the number of filters of each expert in a bottleneck manner, thus reducing the number of parameters to maintain comparable computational complexity. We also add an additional projection shortcut that connects the MoE layer input with its output. We evaluate embedding sparsely-gated MoE in each possible position in ResNet-18.

We train models with 4 experts and set the number of active experts to $k = 2$. We set weights for auxiliary losses to $w_{imp} = w_{KL} = 0.5$, and thresholds for hard constraints to $m_{rel} = 0.5$ and $m_{mean} = 0.3$.

4.2. Performance and Expert Utilization

Dying experts: soft constraints successfully mitigated the experts death (see Table 1). Both auxiliary losses minimize the variation in the importance per expert, whereas a higher variation in the number of samples per expert was observed for the KL-divergence loss. Hard constraints, however, demonstrated worse results. Mean importance models could not keep all experts alive in a single model, whereas relative importance models had no dying experts for models with 4 experts. Also, increasing the number of experts to 10 led to more variance in expert utilization.

Accuracy and computational cost: in our setting, embedding MoE layers into the model aims at boosting the interpretability strength, not at beating the baseline performance. Out of the evaluated models, the best results were achieved with the mean importance constraint (see Table 2). The choice of a constraint thus provides an evident trade-off between overall performance and the death of experts.

In all models, we keep the number of active experts $k = 2$ to maintain a computational budget comparable to the baseline, which reaches 0.56 GMac (Mac: multiply-accumulate). Although increasing the number of active experts $k$ led to better performance, each additionally activated expert adds about 0.06 to 0.08 GMac depending on the layer position. For $k = 3$, the overall model reaches on average 0.63 GMac, for $k = 4$ already 0.7 GMac. $k$ thus controls a trade-off between accuracy and computational complexity.

Impact of the gate architecture: We have evaluated replacing the GAP-FC gate with Conv-GAP-FC gate, exemplary for ResBlock-MoE with 4 experts at position 4.

---

\(^1\)Available at https://github.com/pytorch/vision/blob/master/torchvision/models/resnet.py

---

| MoE pos. | Importance loss | KL divergence loss | Relative importance | Mean importance |
|----------|----------------|-------------------|-------------------|----------------|
|          | $CV_{act}$ | $CV_{imp}$ | # | $CV_{act}$ | $CV_{imp}$ | # | $CV_{act}$ | $CV_{imp}$ | # |
| ResBlock-MoE-4 experts | 1 | 6.72 | 3.82 | 4 | 9.23 | 6.36 | 4 | 56.00 | 56.34 | 4 | 97.37 | 103.23 | 2.33 |
| ResBlock-MoE-10 experts | 1 | 10.69 | 10.43 | 10 | 14.93 | 13.47 | 10 | 160.60 | 168.11 | 5.33 | 199.77 | 200.02 | 2.00 |

Table 1. Expert utilization: coefficients of variation for the number of samples ($CV_{act}$) and importance ($CV_{imp}$) assigned to each expert, higher $CV$ value means higher variance in expert utilization. # stands for the number of live experts. We highlight cases, when all experts are alive.
Table 2. Mean accuracy and sample standard deviation of image classification models with sparse MoE layers and GAP-FC gate. We highlight cases that beat the baseline accuracy.

| MoE position | Importance loss | KL-divergence loss | Relative importance constraint | Mean importance constraint |
|--------------|-----------------|--------------------|-------------------------------|--------------------------|
| Baseline     |                 |                    |                               | 72.62 ± 0.29             |
| 1            | 72.24 ± 0.49    | 72.72 ± 0.36       | 72.21 ± 0.42                 | 73.00 ± 0.40             |
| ResBlock-MoE, 4 experts | 2 | 72.18 ± 0.29 | 72.25 ± 0.17 | 72.18 ± 0.48 | 72.95 ± 0.35 |
| 3            | 71.65 ± 0.43    | 71.54 ± 0.22       | 72.05 ± 0.15                 | 72.61 ± 0.37             |
| 4            | 71.95 ± 0.39    | 71.80 ± 0.23       | 73.10 ± 0.25                 | 72.57 ± 0.31             |
| ResBlock-MoE, 10 experts | 1 | 71.51 ± 0.23 | 71.32 ± 0.60 | 71.60 ± 0.09 | 72.28 ± 0.15 |
| 2            | 72.76 ± 0.50    | 71.87 ± 0.30       | 71.51 ± 0.44                 | 72.08 ± 0.32             |
| 3            | 71.47 ± 0.29    | 71.43 ± 0.12       | 70.82 ± 0.16                 | 72.05 ± 0.61             |
| 4            | 71.61 ± 0.16    | 71.99 ± 0.23       | 72.84 ± 0.30                 | 73.09 ± 0.35             |

Table 3. Correlation between weight assignment and expert performance for the models with the ResBlock-MoE layer at position 4.

| Correlation between | Importance loss | Relative importance constraint |
|---------------------|-----------------|-------------------------------|
|                      | Pearson       | Spearman | Pearson       | Spearman |
| ... expert accuracy and sparse weights | 0.8720 | 0.8870 | 0.5827 | 0.5765 |
| ... expert accuracy and non-sparse weights | 0.8705 | 0.8903 | 0.5919 | 0.5981 |
| ... expert accuracy and activations per expert | 0.8501 | 0.8301 | 0.5981 | 0.6172 |

and importance loss. This model reached the accuracy of 72.42 ± 0.27, beating the corresponding GAP-FC model (71.95 ± 0.39), but not the baseline. Hard-constrained models using Conv-GAP-FC gate suffer massively from dying experts, even with a decreased learning rate.

4.3. Interpretability via Sparsely-Gated MoEs

Dataset partitioning by gate: visual assessment of the gate logits, plotted with t-SNE (see Fig. 3) demonstrates, that MoE at position 1 leads to assignment based on the dominant colors of input images, whereas for position 4, the distinctions are much more faded. Furthermore, visible structures are less significant for the soft-constrained models, compared to the hard-constrained case.

The resulting sample assignment to different experts reveals more striking differences across constraints (see Fig. 4). For earlier layers, the gate divides the data into 2 major subdomains, while for position 4, the gate varies more between different expert combinations. Weight assignment in deeper MoE layers is thus based more on high-level features and leads to stronger differentiation. For hard-constrained models, visible structures are less significant.

A more complex Conv-GAP-FC gate subdivides the assignments clearer. The gate thus produces less unambiguous weight vectors and selects experts definitely.

Experts specialization: to analyze the implicit specialization of experts on distinct subdomains of the input space, we only activate one specific expert during evaluation and assign all weights to it. We then analyzed classes that received the largest weights in each expert during evaluation (Table 4). We could observe distinct repeating clusters of classes for different models, e.g., flowers, marine animals, trees, and furniture. Also, inserting MoE layers at the deeper positions in a network led to larger weights assigned to the experts, indicating better specialization.

Furthermore, we have determined, that a gate chooses the best-performing expert for the images of the corresponding cluster. For this, we extracted results for classes in which each expert is assigned the highest and lowest weights. Full MoE performs in 73 out of 100 classes at least as well as its best experts in this domain. The gate can thus identify experts for distinct domains and suitably support experts with inferior performance. It is consequently able to reasonably combine the output feature maps to improve the overall predictions.

Expert utilization vs. accuracy: next, we have evaluated the correlation between the average weights assigned to an expert (sparse and non-sparse, i.e. with all experts activated) and every single expert’s test accuracy per class. We have also evaluated the correlation between accuracy and the number of activations per expert. The results (see Table 3) indicate a strong relationship for the soft-constrained model. For the hard-constrained case, the experts are more generalized and do not show as large performance variations on different classes, the gate does not rely on the same experts for a certain class.
Table 4. Class-wise weight allocation for the ResNet-18 models. Results for the top 5 classes for which the experts receive the largest weights.

| Constr | Gate               | Pos | Expert 1 | Expert 2 | Expert 3 | Expert 4 |
|--------|--------------------|-----|----------|----------|----------|----------|
|        |                    |     | Class    | Weight   | Class    | Weight   |
|        |                    |     |          |          |          |          |
|        | GAP-FC             | 1   | orange   | 0.71     | caterpillar | 0.4 | chair    | 0.59 |
|        |                    |     | rose     | 0.64     | forest    | 0.38   | cockroach | 0.55 |
|        |                    |     | apple    | 0.59     | mushroom  | 0.34   | tank     | 0.46 |
|        |                    |     | sweet pepper | 0.58 | butterfly | 0.31   | plate    | 0.45 |
|        |                    |     | sunflower | 0.56     | aquarium fish | 0.30 | lawn mower | 0.43 |
|        |                    |     |          |          |          |          |          |          |
|        |                    | 4   | mountain | 0.86     | leopard   | 0.78   | poppy    | 0.80 |
|        |                    |     | dolphin  | 0.86     | forest    | 0.75   | orange   | 0.79 |
|        |                    |     | shark    | 0.80     | porcupine | 0.72   | rose     | 0.76 |
|        |                    |     | sea      | 0.79     | tiger     | 0.64   | tulip    | 0.71 |
|        |                    |     | whale    | 0.79     | mushroom  | 0.63   | sunflower | 0.71 |
|        |                    |     |          |          |          |          |          |          |
|        | GAP-FC-Conv        | 4   | plain    | 0.88     | chair     | 0.90   | leopard  | 0.77 |
|        |                    |     | mountain | 0.87     | telephone | 0.80   | porcupine | 0.75 |
|        |                    |     | sea      | 0.84     | cockroach | 0.78   | crocodile | 0.65 |
|        |                    |     | dolphin  | 0.82     | clock     | 0.63   | forest   | 0.64 |
|        |                    |     | cloud    | 0.78     | cup       | 0.60   | kangaroo | 0.63 |
|        |                    |     |          |          |          |          |          |          |
|        | Relative importance| 4   | chair    | 0.96     | plain     | 0.87   | poppy    | 0.86 |
|        |                    |     | television | 0.79 | willow tree | 0.82 | sunflower | 0.83 |
|        |                    |     | bottle   | 0.79     | oak tree  | 0.79   | rose     | 0.80 |
|        |                    |     | telephone | 0.75     | forest    | 0.68   | tulip    | 0.77 |
|        |                    |     | cup      | 0.74     | maple tree | 0.66   | orange   | 0.74 |

Figure 3. Visualization of gate logits using t-SNE for ResBlock-MoE with 4 experts.

Figure 4. Assignment of input samples to specific experts in models with ResBlock-MoE, visualized with t-SNE.
5. Experiments with Sparse MoEs for Object Detection

5.1. Experimental Setup

We use a pretrained\(^2\) RetinaNet [11] with the ResNet-50 backbone as a baseline, and the COCO dataset [12]. We train all models using $\gamma = 2$ and $\alpha = 0.25$ for focal loss. We report standard COCO metrics mAP@[.5, .95] and mAP@.5.

We embed the sparse MoE layers in two manners: (1) 2Block-MoE: by replacing the regression and classification subnets with 2 separate MoE blocks, and (2) SingleMoE: with a single gate shared between regressor and classifier. We keep the backbone weights frozen during training. Mean importance constraints are not included in the evaluation, since the models suffer from massive dying expert problems.

We also trained models with unfrozen weights but did not observe performance improvements. The gate is Conv-GAP-FC. All experts are initialized using the Kaiming approach [5] to learn more diverse features. Weighting factors are reduced to $w_{imp} = w_{KL} = 0.25$ to guarantee better expert utilization for deeper MoE layers. Additionally, we set $m_{rel} = 0.3$ to avoid dying experts. We only train models with 4 experts and set $k = 2$.

5.2. Evaluation

**Performance:** the hard-constrained models performed slightly better than the soft-constrained ones, although none of the models with MoE layers outperformed the baseline (see Table 5). We further observe only a slight performance drop for the Single model compared to 2 separate gates.

**Specialization of the 2Block-MoE in regressor:** visual comparison of the BBoxes reveals, that all experts have problems estimating the precise object boundaries for an atypical pose, whereas predictions on an object with a clear front view vary less (see Fig. 5). The gate is not always able to pick the single best expert. Still, the inactive experts would both overestimate the bottom of the bounding box, while the selected experts both predict the bottom tightly. Predictions made by distinct experts in the hard-constrained model vary less than for models trained with soft constraints.

The classwise analysis and the evaluation for different feature levels have revealed that experts in regressor MoE layers do not specialize in predicting BBoxes for distinct classes. On the contrary, the gate selects experts mainly depending on the input of different feature map levels.

**Specialization of the 2Block-MoE in classifier:** we have analyzed the average assigned weights to each expert during evaluation. Expert 1 stands out and performs significantly worse than the other experts. The expert is mostly utilized on feature map level $P_7$ with about 90% of weight assignments. Consequently, the expert specializes in detecting large objects of this specific feature map size. It also receives about a quarter of weights for level $P_8$ but is not the top-weighted expert. On other levels, the expert is not significantly utilized. We conclude that expert 1 can detect large objects but, on the other side, is not able to detect and classify objects of smaller scales as precisely as the other experts. When comparing the other three experts in 2Block-MoE-KL, differences are still small but slightly larger than for the regressor experts. Figure 6 shows predictions of distinct classifier experts for an exemplary image.

Overall, MoE layers embedded in the classifier and regressor subnets, the decision units in the network, allow us to gain insights into the models’ decision processes. We can separately analyze each expert’s single predictions and the resulting MoE prediction. Distinct expert networks specialize in detecting objects of specific sizes.

**Impact of pretraining:** finally, we investigate the behavior of MoE models using pre-trained weights for expert networks. For this, we re-use baseline weights and add Gaussian noise to enable different expert specializations. We also train Conv4 models by only replacing the 4th convolutional layers in the regressor and classifier subnets with an MoE layer. For comparison, we also train Conv4 models without pre-trained weights (see Table 5).

| Model                  | mAP@.5 | mAP@[.5, .95] |
|------------------------|--------|--------------|
| Baseline               | 35.0   | 52.5         |
| 2Block-MoE-KL          | 33.5   | 50.9         |
| 2Block-MoE-Imp         | 33.5   | 50.9         |
| 2Block-MoE-Rel         | 33.7   | 51.0         |
| SingleMoE-KL           | 33.4   | 50.9         |
| 2Block-MoE-KL Conv4    | 33.6   | 50.9         |
| 2Block-MoE-Rel Conv4   | 33.6   | 50.9         |
| 2Block-MoE-KL Pretr.   | 35.1   | 52.7         |
| 2Block-MoE-Rel Pretr.  | 34.4   | 51.7         |
| 2Block-MoE-KL Conv4 Pretr. | 35.1   | 52.6         |

Table 5. Object detection mAP (%) of MoE models on COCO test-dev2017. We highlight cases that beat the baseline accuracy.

\(^2\)Available at [https://github.com/yhenon/pytorch-retinanet](https://github.com/yhenon/pytorch-retinanet)
6. Conclusion

In this work, we applied the sparsely-gated MoE layers to CNNs for computer vision tasks with the goal of increasing the model’s interpretability. We presented constraints to mitigate the dying expert problem, which tackle the problem from different angles and lead to different MoE behavior. Our analysis has revealed several interconnections between the proposed constraints on the one hand, and model performance as well as interpretability, on the other hand. Hard constraints result in better overall performance and generalized experts, although the mean importance constraint is particularly prone to the dying experts problem. Soft constraints lead to better expert specialization. The usage of constraints thus helps to control the interplay between model performance, training stability and expert specialization.

Our experiments have revealed inherent interpretability for two evaluated computer vision tasks. For the image classification task, experts focused on distinct repeating class groups, whereas for object detection, they specialized in objects of distinct sizes. We hope that our insights pave the way for further research on the interpretability of deep neural networks.
Acknowledgement

The research leading to these results is funded by the German Federal Ministry for Economic Affairs and Climate Action within the project “KI Absicherung” (grant 19A19005W) and by KASTEL Security Research Labs. The authors would like to thank the consortium for the successful cooperation.

References

[1] Aidan Clark, Diego de Las Casas, Aurelia Guy, Arthur Mensch, Michela Paganini, Jordan Hoffmann, Bogdan Damoc, Blake A. Hechtman, Trevor Cai, Sebastian Borgeaud, George van den Driessche, Eliza Rutherford, Tom Hennigan, Matthew J. Johnson, Albin Cassirer, Chris Jones, Elena Buchatskaya, David Budden, Laurent Sifre, Simon Osindero, Oriol Vinyals, Marc’Aurelio Ranzato, Jack W. Rae, Erich Elsen, Koray Kavukcuoglu, and Karen Simonyan. Unified scaling laws for routed language models. In Proceedings of the International Conference on Machine Learning (ICML). PMLR, 2022.

[2] Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Lian Fendu, Maarten P. Bosma, Zongwei Zhou, Tao Wang, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen S. Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc V. Le, Yonghui Wu, and Zhifeng Chen. Glam: Efficient scaling of language models with mixture-of-experts. In Proceedings of the International Conference on Machine Learning (ICML). PMLR, 2022.

[3] David Eigen, Marc’Aurelio Ranzato, and Ilya Sutskever. Learning factored representations in a deep mixture of experts. In Yoshua Bengio and Yann LeCun, editors, Proceedings of the International Conference on Learning Representations (ICLR), 2014.

[4] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. CoRR, abs/2101.03961, 2021.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision (ICCV). IEEE, 2015.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2016.

[7] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. Neural computation, 3(1), 1991.

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

[9] HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. In Proceedings of the International Conference on Learning Representations (ICLR), 2021.

[10] Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. BASE layers: Simplifying training of large, sparse models. In Marina Meila and Tong Zhang, editors, Proceedings of the International Conference on Machine Learning (ICML). PMLR, 2021.

[11] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 3099–3007. IEEE Computer Society, 2017.

[12] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, Proceedings of the European Conference on Computer Vision (ECCV). Springer, 2014.

[13] Lu Lu, Yeonjong Shin, Yanhui Su, and George E. Karniadakis. Dying relu and initialization: Theory and numerical examples. CoRR, abs/1903.06733, 2019.

[14] Basil Mustafa, Carlos Riquelme, Joao Pujigervver, Rodolphe Jenatton, and Neil Houlsby. Multimodal contrastive learning with limoe: the language-image mixture of experts. CoRR, abs/2206.02770, 2022.

[15] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Cuturi, Wei Cai, Ian Goodfellow, and Jeff Dean. Gshard: Scaling giant models with conditional computation and automatic sharding. In Proceedings of the International Conference on Learning Representations (ICLR), 2022.

[16] Carlos Riquelme, Joao Pujigervver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. CoRR, abs/2106.05974, 2021.

[17] Stephen Roller, Sainbayar Sukhbaatar, Arthur Szlam, and Jason Weston. Hash Layers For Large Sparse Models. arXiv:2106.04426 [cs], 2021.

[18] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In Proceedings of the International Conference on Learning Representations (ICLR), 2017.

[19] Lemeng Wu, Mengchen Liu, Yinpeng Chen, Dongdong Chen, Xiyang Dai, and Lu Yuan. Residual mixture of experts. CoRR, abs/2206.05974, 2021.

[20] Fuzhao Xue, Ziji Shi, Futao Wei, Yuxuan Lou, Yong Liu, and Yang You. Go wider instead of deeper. CoRR, abs/2107.11817, 2021.

[21] Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. Designing Effective Sparse Expert Models. arXiv:2202.08906 [cs], 2022.