Mixtral of Experts

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, William El Sayed

Abstract

We introduce Mixtral 8x7B, a Sparse Mixture of Experts (SMoE) language model. Mixtral has the same architecture as Mistral 7B, with the difference that each layer is composed of 8 feedforward blocks (i.e. experts). For every token, at each layer, a router network selects two experts to process the current state and combine their outputs. Even though each token only sees two experts, the selected experts can be different at each timestep. As a result, each token has access to 47B parameters, but only uses 13B active parameters during inference. Mixtral was trained with a context size of 32k tokens and it outperforms or matches Llama 2 70B and GPT-3.5 across all evaluated benchmarks. In particular, Mixtral vastly outperforms Llama 2 70B on mathematics, code generation, and multilingual benchmarks. We also provide a model fine-tuned to follow instructions, Mixtral 8x7B – Instruct, that surpasses GPT-3.5 Turbo, Claude-2.1, Gemini Pro, and Llama 2 70B – chat model on human benchmarks. Both the base and instruct models are released under the Apache 2.0 license.

Code: https://github.com/mistralai/mistral-src
Webpage: https://mistral.ai/news/mixtral-of-experts/

1 Introduction

In this paper, we present Mixtral 8x7B, a sparse mixture of experts model (SMoE) with open weights, licensed under Apache 2.0. Mixtral outperforms Llama 2 70B and GPT-3.5 on most benchmarks. As it only uses a subset of its parameters for every token, Mixtral allows faster inference speed at low batch-sizes, and higher throughput at large batch-sizes.

Mixtral is a sparse mixture-of-experts network. It is a decoder-only model where the feedforward block picks from a set of 8 distinct groups of parameters. At every layer, for every token, a router network chooses two of these groups (the "experts") to process the token and combine their output additively. This technique increases the number of parameters of a model while controlling cost and latency, as the model only uses a fraction of the total set of parameters per token.

Mixtral is pretrained with multilingual data using a context size of 32k tokens. It either matches or exceeds the performance of Llama 2 70B and GPT-3.5, over several benchmarks. In particular,

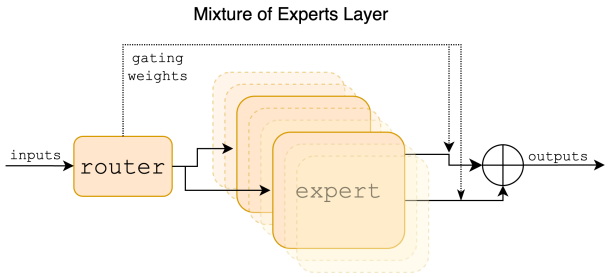

Figure 1: Mixture of Experts Layer. Each input vector is assigned to 2 of the 8 experts by a router. The layer's output is the weighted sum of the outputs of the two selected experts. In Mixtral, an expert is a standard feedforward block as in a vanilla transformer architecture.

Mixtral demonstrates superior capabilities in mathematics, code generation, and tasks that require multilingual understanding, significantly outperforming Llama 2 70B in these domains. Experiments show that Mixtral is able to successfully retrieve information from its context window of 32k tokens, regardless of the sequence length and the location of the information in the sequence.

We also present Mixtral 8x7B – Instruct, a chat model fine-tuned to follow instructions using supervised fine-tuning and Direct Preference Optimization [25]. Its performance notably surpasses that of GPT-3.5 Turbo, Claude-2.1, Gemini Pro, and Llama 2 70B – chat model on human evaluation benchmarks. Mixtral – Instruct also demonstrates reduced biases, and a more balanced sentiment profile in benchmarks such as BBQ, and BOLD.

We release both Mixtral 8x7B and Mixtral 8x7B – Instruct under the Apache 2.0 license[1], free for academic and commercial usage, ensuring broad accessibility and potential for diverse applications. To enable the community to run Mixtral with a fully open-source stack, we submitted changes to the vLLM project, which integrates Megablocks CUDA kernels for efficient inference. Skypilot also allows the deployment of vLLM endpoints on any instance in the cloud.

2 Architectural details

Mixtral is based on a transformer architecture [31] and uses the same modifications as described in [18], with the notable exceptions that Mixtral supports a fully dense context length of 32k tokens, and the feedforward blocks are replaced by Mixture-of-Expert layers (Section 2.1). The model architecture parameters are summarized in Table 1.

Parameter	Value
dim	4096
n_layers	32
head_dim	128
hidden_dim	14336
n_heads	32
n_kv_heads	8
context_len	32768
vocab_size	32000
num_experts	8
top_k_experts	2

Table 1: Model architecture.

2.1 Sparse Mixture of Experts

We present a brief overview of the Mixture of Experts layer (Figure 1). For a more in-depth overview, see [12]. The output of the MoE module for a given input x is determined by the weighted sum of the outputs of the expert networks, where the weights are given by the gating network's output. i.e. given n expert networks $\{E_0, E_i, ..., E_{n-1}\}$, the output of the expert layer is given by:

$$\sum_{i=0}^{n-1} G(x)_i \cdot E_i(x).$$

Here, $G(x)_i$ denotes the n-dimensional output of the gating network for the i-th expert, and $E_i(x)$ is the output of the i-th expert network. If the gating vector is sparse, we can avoid computing the outputs of experts whose gates are zero. There are multiple alternative ways of implementing $G(x)$ [6, 15, 35], but a simple and performant one is implemented by taking the softmax over the Top-K logits of a linear layer [28]. We use

$$G(x) := \text{Softmax}(\text{TopK}(x \cdot W_g)),$$

where $(\text{TopK}(\ell))_i := \ell_i$ if ℓ_i is among the top-K coordinates of logits $\ell \in \mathbb{R}^n$ and $(\text{TopK}(\ell))_i := -\infty$ otherwise. The value of K – the number of experts used per token – is a hyper-parameter that modulates the amount of compute used to process each token. If one increases n while keeping K fixed, one

[1] https://mistral.ai/news/mixtral-of-experts/

can increase the model's parameter count while keeping its computational cost effectively constant. This motivates a distinction between the model's total parameter count (commonly referenced as the **sparse** parameter count), which grows with n, and the number of parameters used for processing an individual token (called the **active** parameter count), which grows with K up to n.

MoE layers can be run efficiently on single GPUs with high performance specialized kernels. For example, Megablocks [13] casts the feed-forward network (FFN) operations of the MoE layer as large sparse matrix multiplications, significantly enhancing the execution speed and naturally handling cases where different experts get a variable number of tokens assigned to them. Moreover, the MoE layer can be distributed to multiple GPUs through standard Model Parallelism techniques, and through a particular kind of partitioning strategy called Expert Parallelism (EP) [28]. During the MoE layer's execution, tokens meant to be processed by a specific expert are routed to the corresponding GPU for processing, and the expert's output is returned to the original token location. Note that EP introduces challenges in load balancing, as it is essential to distribute the workload evenly across the GPUs to prevent overloading individual GPUs or hitting computational bottlenecks.

In a Transformer model, the MoE layer is applied independently per token and replaces the feed-forward (FFN) sub-block of the transformer block. For Mixtral we use the same SwiGLU architecture as the expert function $E_i(x)$ and set $K = 2$. This means each token is routed to two SwiGLU sub-blocks with different sets of weights. Taking this all together, the output y for an input token x is computed as:

$$y = \sum_{i=0}^{n-1} \text{Softmax}(\text{Top2}(x \cdot W_g))_i \cdot \text{SwiGLU}_i(x).$$

This formulation is similar to the GShard architecture [21], with the exceptions that we replace all FFN sub-blocks by MoE layers while GShard replaces every other block, and that GShard uses a more elaborate gating strategy for the second expert assigned to each token.

3 Results

We compare Mixtral to Llama, and re-run all benchmarks with our own evaluation pipeline for fair comparison. We measure performance on a wide variety of tasks categorized as follow:

- **Commonsense Reasoning (0-shot):** Hellaswag [32], Winogrande [26], PIQA [3], SIQA [27], OpenbookQA [22], ARC-Easy, ARC-Challenge [8], CommonsenseQA [30]
- **World Knowledge (5-shot):** NaturalQuestions [20], TriviaQA [19]
- **Reading Comprehension (0-shot):** BoolQ [7], QuAC [5]
- **Math:** GSM8K [9] (8-shot) with maj@8 and MATH [17] (4-shot) with maj@4
- **Code:** Humaneval [4] (0-shot) and MBPP [1] (3-shot)
- **Popular aggregated results:** MMLU [16] (5-shot), BBH [29] (3-shot), and AGI Eval [34] (3-5-shot, English multiple-choice questions only)

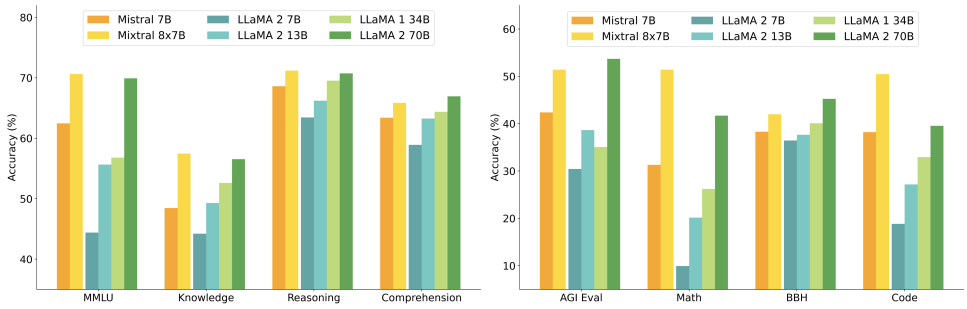

Figure 2: Performance of Mixtral and different Llama models on a wide range of benchmarks. All models were re-evaluated on all metrics with our evaluation pipeline for accurate comparison. Mixtral outperforms or matches Llama 2 70B on all benchmarks. In particular, it is vastly superior in mathematics and code generation.

Model	Active Params	MMLU	HellaS	WinoG	PIQA	Arc-e	Arc-c	NQ	TriQA	HumanE	MBPP	Math	GSM8K
LLaMA 2 7B	7B	44.4%	77.1%	69.5%	77.9%	68.7%	43.2%	17.5%	56.6%	11.6%	26.1%	3.9%	16.0%
LLaMA 2 13B	13B	55.6%	80.7%	72.9%	80.8%	75.2%	48.8%	16.7%	64.0%	18.9%	35.4%	6.0%	34.3%
LLaMA 1 33B	33B	56.8%	83.7%	76.2%	82.2%	79.6%	54.4%	24.1%	68.5%	25.0%	40.9%	8.4%	44.1%
LLaMA 2 70B	70B	69.9%	**85.4%**	**80.4%**	82.6%	79.9%	56.5%	25.4%	**73.0%**	29.3%	49.8%	13.8%	69.6%
Mistral 7B	7B	62.5%	81.0%	74.2%	82.2%	80.5%	54.9%	23.2%	62.5%	26.2%	50.2%	12.7%	50.0%
Mixtral 8x7B	13B	**70.6%**	84.4%	77.2%	**83.6%**	**83.1%**	**59.7%**	**30.6%**	71.5%	**40.2%**	**60.7%**	**28.4%**	**74.4%**

Table 2: Comparison of Mixtral with Llama. Mixtral outperforms or matches Llama 2 70B performance on almost all popular benchmarks while using 5x fewer active parameters during inference.

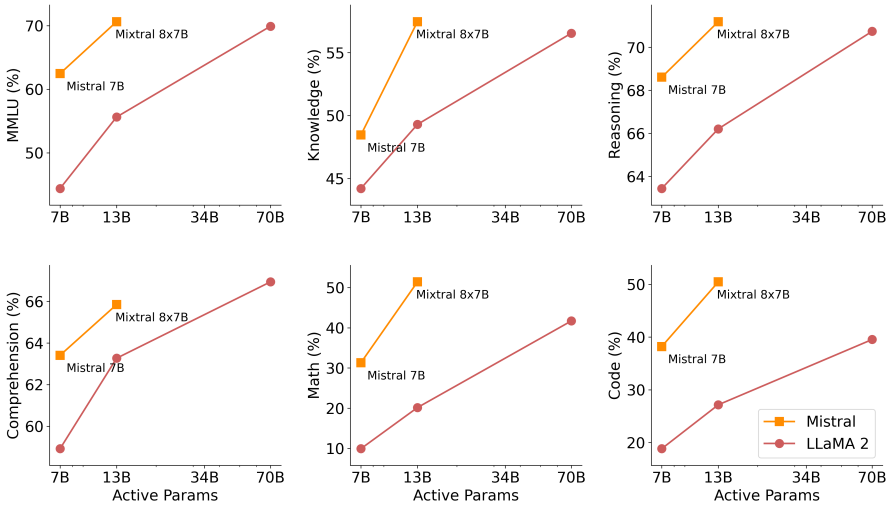

Figure 3: Results on MMLU, commonsense reasoning, world knowledge and reading comprehension, math and code for Mistral (7B/8x7B) vs Llama 2 (7B/13B/70B). Mixtral largely outperforms Llama 2 70B on all benchmarks, except on reading comprehension benchmarks while using 5x lower active parameters. It is also vastly superior to Llama 2 70B on code and math.

Detailed results for Mixtral, Mistral 7B and Llama 2 7B/13B/70B and Llama 1 34B[2] are reported in Table 2. Figure 2 compares the performance of Mixtral with the Llama models in different categories. Mixtral surpasses Llama 2 70B across most metrics. In particular, Mixtral displays a superior performance in code and mathematics benchmarks.

Size and Efficiency. We compare our performance to the Llama 2 family, aiming to understand Mixtral models' efficiency in the cost-performance spectrum (see Figure 3). As a sparse Mixture-of-Experts model, Mixtral only uses 13B active parameters for each token. With 5x lower active parameters, Mixtral is able to outperform Llama 2 70B across most categories.

Note that this analysis focuses on the active parameter count (see Section 2.1), which is directly proportional to the inference compute cost, but does not consider the memory costs and hardware utilization. The memory costs for serving Mixtral are proportional to its *sparse* parameter count, 47B, which is still smaller than Llama 2 70B. As for device utilization, we note that the SMoEs layer introduces additional overhead due to the routing mechanism and due to the increased memory loads when running more than one expert per device. They are more suitable for batched workloads where one can reach a good degree of arithmetic intensity.

Comparison with Llama 2 70B and GPT-3.5. In Table 3, we report the performance of Mixtral 8x7B compared to Llama 2 70B and GPT-3.5. We observe that Mixtral performs similarly or above the two other models. On MMLU, Mixtral obtains a better performance, despite its significantly smaller capacity (47B tokens compared to 70B). For MT Bench, we report the performance of the latest GPT-3.5-Turbo model available, `gpt-3.5-turbo-1106`.

[2]Since Llama 2 34B was not open-sourced, we report results for Llama 1 34B.