Tricks for Training Sparse Translation Models

Dheeru Dua♡ **Shruti Bhosale**♠ **Vedanuj Goswami**♠ **James Cross**♠
Mike Lewis♠ **Angela Fan**♠
♡University of California, Irvine, USA
♠Facebook AI
dduа@uci.edu

Abstract

Multi-task learning with an unbalanced data distribution skews learning towards high resource tasks, especially when model capacity is fixed and fully shared across all tasks. Sparse scaling architectures, such as BASELayers, provide flexible mechanisms for tasks to have a variable number of parameters, which can be useful to counterbalance skewed data distributions. However, we find that that BASELayers sparse model for multilingual machine translation can perform poorly out of the box, and propose two straightforward techniques to mitigate this — a temperature heating mechanism and dense pre-training. Overall, these methods improve performance on two multilingual translation benchmarks compared to standard BASELayers and dense scaling baselines, and in combination, more than 2x model convergence speed.

1 Introduction

Training a universal model capable of handling many different tasks is a longstanding ambition in natural language processing (Collobert and Weston, 2008; Ruder, 2017; McCann et al., 2018), with recent progress driven by training transformer models on a wide range of tasks (Xue et al., 2021; Khashabi et al., 2020; Lu et al., 2020). A central challenge in multi-task learning is accounting for the dramatically varying amounts of training data available for different tasks, which can lead to overfitting on low-resource tasks whilst simultaneously underfitting on tasks with abundant training data.

In this work, we study multilingual machine translation as a multi-task learning problem (Dong et al., 2015; Firat et al., 2016), where a single model is trained to translate between many language pairs (Fan et al., 2021). Multilingual learning has the potential of crosslingual transfer, allowing low-resource languages to benefit from high-resource data (Conneau et al., 2020). However, in practice, this positive transfer is often mitigated by interference between languages (Arivazhagan et al., 2019; Tan et al., 2019; Zhang et al., 2020). This is because all languages, irrespective of the amount of data, are trained with a fixed model capacity (Lepikhin et al., 2020), leading to insufficient specialized capacity. Recent efforts have focused on sparse architectures (Lewis et al., 2021) to train high capacity models, but these overfit to low-resource languages and have worse performance than dense architectures (Fan et al., 2021; Tran et al., 2021). We analyze the learning patterns of experts throughout training and identify a fundamental problem: experts specialize early on and rarely change specialization.

We propose two straightforward techniques to improve BASELayers-based sparse architectures (Lewis et al., 2021) for multitask learning: first, we slowly ramp the number of instances from low-resource tasks over epochs rather than having a fixed sampling ratio (Arivazhagan et al., 2019). This promotes cross-lingual transfer and reduces over-fitting as the model witnesses low-resource task instances in the later epochs. Second, we train a dense architecture before switching to sparse training. Intuitively, we learn a generalized representation that can transfer across all tasks first with a dense model and then gradually sparsify and specialize the experts to different tasks. Overall with these two modifications, we observe improvement in low-resource performance by 0.6 BLEU on WMT-15 benchmark and 1.1 BLEU on ML-50 benchmark — whilst halving the training time.

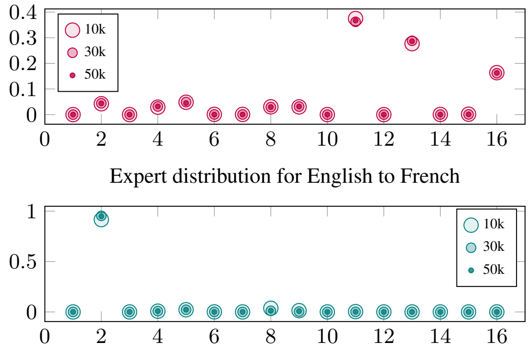
2 Methods

We motivate the need for preventing early expert specialization and describe our proposal to circumvent it and more than double convergence speed.

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3340 - 3345
July 10-15, 2022 ©2022 Association for Computational Linguistics

2.1 Expert Utilization Rarely Changes

Sparse scaling schemes, such as BASELayers or Mixture-of-Experts, enable sparse computation by distributing model capacity across sets of experts. In each forward pass, only a small subset of experts are utilized, leading to incredibly compute-efficient scaling. The challenge, however, is the *routing function* — or how experts can be balanced so they actually specialize to learn different things (Kudugunta et al., 2020). When the routing mechanism is unbalanced, all the tasks degenerate to using only a single specific expert for all tasks (Lepikhin et al., 2020) — essentially wasting parameters. BASELayers (Lewis et al., 2021) employ a simple mechanism that learns a balanced routing without the need for additional auxiliary losses. We focus on BASELayers as it has straightforward and simple training and has previously been shown to have strong performance.



Expert distribution for English to French

Expert distribution for English to Romanian

Figure 1: Expert distribution for Romanian and French as training progresses (10k, 30k and 50k updates) on WMT-15 benchmark, where Romanian is low-resource and French is high-resource.

Even though BASELayers leads to effective utilization of all parameters, it limits parameter sharing across tasks, which is crucial when the data distribution is unbalanced — if the number of tasks and experts are the same, all tasks end up using a different set of experts. As a result, when applied to multilingual machine translation, the performance is worse than a corresponding dense architecture. Figure 1 demonstrates that the main reason for limited parameter sharing is that expert assignment is fixed incredibly early on in training and rarely changes. Instead of learning how to better utilize capacity across high and low-resource languages over the training process, expert capacity is essentially frozen. We describe two strategies for more effective utilization of expert capacity, which can be easily applied to improve both low and high-resource translation performance.

2.2 Balancing Low-Resource Tasks

Temperature Sampling: To ensure that low-resource tasks are well represented during model training, temperature sampling (Arivazhagan et al., 2019) is used to upsample low-resource tasks. If the data distribution across different tasks is p, then temperature sampling re-scales this distribution:

$$p \leftarrow \frac{p_n^{1/T}}{\sum_{n\in|\text{tasks}|} p_n^{1/T}} \qquad (1)$$

As we increase temperature from 1 to ∞, the sampling distribution changes from the original data distribution (e.g. highly skewed) to a uniform distribution (e.g. tasks are equally represented).

Temperature Heating: Instead of keeping the temperature fixed while sampling data for each task, we define temperature as a function of current epoch, $t = f(e)$

We define a minimum starting temperature t_s, which is gradually increased at each epoch e, with a square root factor defined over maximum number of epochs C. The conduction coefficient k influences the rate at which the temperature is increased over epochs.

$$t_e = \left(\sqrt{1+\frac{k}{C}e}\right)t_s \qquad (2)$$

In particular, we adopt square root scaling of temperature with each epoch, instead of linear to allow for gradual changes to the sampling distribution. During the initial steps of training, this trains with lower temperatures, meaning high-resource tasks are better represented than low-resource tasks. As a result, the experts are more uniformly assigned across high-resource tasks. Upon slowly introducing low-resource tasks by increasing temperature during the learning process, the gating mechanism learns to route low-resource tasks through experts which were initially trained with high-resource tasks. This promotes positive cross-lingual transfer from high-resource languages to linguistically similar low-resource languages.

2.3 Dense Pre-training

Architecturally, the sparsity in the output feed-forward layer of the transformer block can be viewed as a version of the same transformer on multiple GPUs with two main differences: the sparse

Model	WMT-15			ML-50			
	Low	High	All	Low	Mid	High	All
Dense	13.3	25.4	19.8	10.7	23.7	24.7	22.5
BASELayers	12.7	25.3	19.4	8.7	22.6	26.5	22.3
+ heat. (t_s=0.8)	12.9	26.4	20.1	8.9	22.9	26.5	22.5
+ heat. (t_s=1.0)	13.2	26.1	20.1	8.5	22.7	26.5	22.3
+ heat. (t_s=1.5)	13.1	26.1	20.0	9.3	22.9	26.5	22.4
+ heat. (t_s=2.0)	13.3	25.5	19.8	10.1	23.7	26.0	22.9

Table 1: Average BLEU over Low resource, High resource and All languages for different starting temperatures with a fixed conduction coefficient, k=1. The baselines are from our best performing dense and BASELayers models

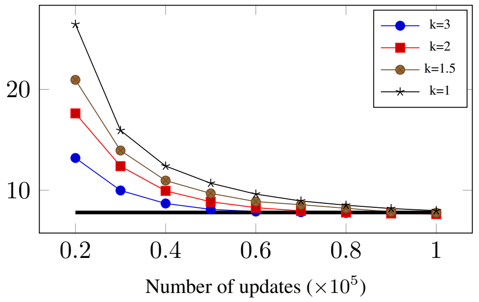


Figure 2: Validation ppl on ML-50 (t_s=0.8). Higher k show faster convergence.

feed-forward layers do not share parameters (have different initialization and gradients) and an additional gating mechanism decides which token should be routed to which expert. The alternative dense architecture would fully share parameters, so all parameters are utilized for each training example rather than routing to sparse parameters.

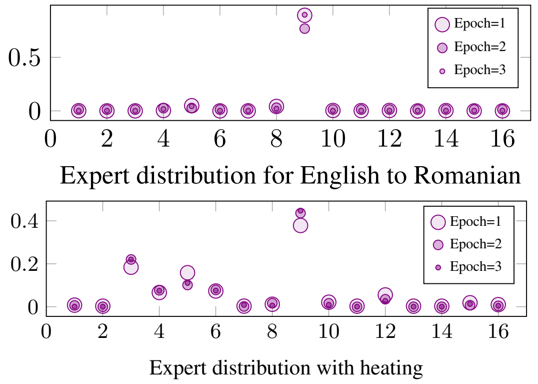
We propose first training a dense model for a fixed number of updates. Afterwards, we add a randomly initialized gating module and continue training the (output) feed-forward layers with sparsity, e.g. we do not average their gradients across compute nodes before back-propagating but update the weights individually in each node. As the sparse weights slowly diverge, they become more specialized towards specific tasks. Thus, models first learn a generalized representation when all parameters are fully shared, and then gradually specialize to handle different tasks. Training in this fashion not only improves the learning of specialized experts, but also increases convergence.

3 Experiments and Results

We experiment with `English` → `Many` multitasking on two benchmarks, WMT-15[1] and ML-50 (Tang et al., 2020) — the first includes 15 languages and the second 50 languages. We use a Transformer (Vaswani et al., 2017) sequence-to-sequence model with 6 encoder and decoder layers. We replace the final feed-forward layer of every alternate transformer block with a BASELayer. For ML50, we increase model capacity to 12 Transformer layers following Tang et al. (2020). We implement our methods in `fairseq` (Ott et al., 2019) and evaluate performance with BLEU.

3.1 Effectiveness of Temperature Heating

On WMT-15, training with BASELayers as a baseline has worse low-resource performance compared to a similarly sized dense model, losing 0.6 BLEU. However, as we increase temperature, we recover the loss in low-resource task performance and also see improvements in the high-resource languages. The heating technique improves the overall BASELayers model performance by +0.7 BLEU (at t_s = 0.8) (see Table 1). We observe similar trends in ML-50, where adding heating improves low-resource performance by +1.4 BLEU. Furthermore, temperature heating improves convergence speed. Given fixed t_s, the higher the k, the faster the model converges. As shown in Figure 2, the model converges to same validation perplexity with k=3 at 50k updates as 100k updates with k=1.



Expert distribution for English to Romanian

Expert distribution with heating

Figure 3: Expert distribution for low-resource English to Latvian as temperature is increased for WMT-15.

3.2 Dense Pre-training with Heating

For the WMT-15 benchmark, Table 2 demonstrates that with dense pre-training, the best performing model improves by +0.75 BLEU over baseline BASELayers model but at the cost of 12% more compute time. To resolve this, we reduce compu-

[1] http://www.statmt.org/wmt15/translation-task.html

Model	WMT-15					ML-50					
	High	Low	All (BLEU)	All (ChrF++)	Walltime	High	Mid	Low	All (BLEU)	All (ChrF++)	Walltime
Dense	25.2	12.7	19.4	49.1	76K	24.7	23.7	10.7	22.5	49.8	596K
+ heating	25.5	13.2	19.8	50.9	42K	25.9	23.4	10.5	**22.7**	50.3	173K
BASELayers	25.2	12.9	19.5	50.5	77K	26.6	22.6	8.7	22.2	49.7	221K
+ heating	26.0	13.0	19.9	52.3	42K	26.5	23.0	9.3	22.4	51.2	151K
Dense Pre-Train	26.3	13.3	**20.2**	54.7	86K	26.8	22.9	9.6	22.6	52.2	122K
+ heating	26.0	13.4	**20.1**	53.4	**31K**	26.7	23.1	9.8	**22.7**	53.0	**87K**

Table 2: Average BLEU across High and Low resource languages and Walltime (min) on WMT-15 and ML-50, with increasing number of dense pre-training steps at a starting fixed temperature of 1.5. Wall clock time is the total training time including dense pre-training and sparse fine-tuning until the model reaches validation perplexity of 5.99 for WMT-15 and 7.6 for ML-50.

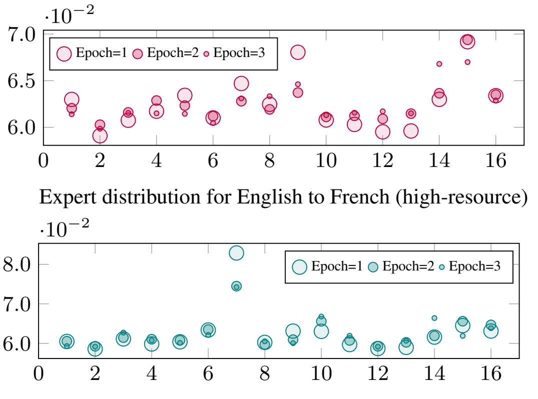
tation time by increasing the temperature, keeping the +0.7 BLEU improvement but reducing the computation time by ∼60%. Table 2 confirms a similar trends on ML-50. By combining Dense Pre-training with heating, we improve over baseline BASELayers model by +0.5 BLEU and 2.5x in convergence speed. However, heating can also be applied to the baselines. In those cases, on both benchmarks, we find that utilizing Dense Pre-training in combination with heating still has slightly better performance with significantly faster convergence.

3.3 Effect on Expert Distribution

In standard BASELayer training, the learned expert distributions rarely change over training (see Figure 1). This prevents effective expert capacity utilization resulting in low-resource overfitting. In contrast, with our proposed techniques, the expert distribution changes and learns over training. Figure 3 compares expert distribution between fixed temperature sampling and temperature heating over epochs for a low-resource language, demonstrating that temperature heating leads experts to change and learn over time. Figure 4 shows that by utilizing dense pre-training, we observe a high entropy in the expert distribution and increased expert sharing, indicating positive cross-lingual transfer from similar high to low-resource languages.

4 Related Work

Data Sampling Low-resource tasks are upsampled to balance their representation when pooled with high-resource tasks. Temperature sampling (Arivazhagan et al., 2019) upsamples the data distribution based on fixed temperature, but can result in overfitting. Dynamic sampler (Gottumukkala et al., 2020) selects instances based on current performance of task on dev set, which is useful in case of catastrophic forgetting. Learned data samplers (Wang et al., 2020) choose better are sample efficient but computationally expensive.



Expert distribution for English to French (high-resource)

Expert distribution for English to Romanian (low-resource)

Figure 4: Expert distribution as sparse fine-tuning progresses on a dense pre-trained model.

Sparse Scaling Sparsely-gated MoE models (Shazeer et al., 2017; Du et al., 2021) use a routing mechanism that decides which expert a task should be routed to. This is the key element that governs effective (better representation) and efficient (balanced assignment) resource utilization. To promote a balanced assignment, routing techniques (Shazeer et al., 2017; Lepikhin et al., 2020; Fedus et al., 2021) add a number of auxiliary task to encourage the use of diverse set of experts. BASELayers (Lewis et al., 2021) circumvents this by treating the routing mechanism as a linear expert-to-task assignment problem, without the need of auxiliary loss. Routing networks (Rosenbaum et al., 2018) learn better task representations by clustering and disentangling parameters conditioned on input.

5 Conclusion

We analyze the problem of balancing shared and specialized capacity in multitask learning, focusing on multilingual machine translation. We present two straightforward tricks to significantly increase convergence rate of mixture-of-expert models and improve their performance relative to dense baselines on two benchmarks.

References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, volume 307 of *ACM International Conference Proceeding Series*, pages 160–167. ACM.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, Beijing, China. Association for Computational Linguistics.

Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten 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, Z. Chen, and Claire Cui. 2021. Glam: Efficient scaling of language models with mixture-of-experts. *ArXiv preprint*, abs/2112.06905.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

William Fedus, Barret Zoph, and Noam Shazeer. 2021. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *ArXiv preprint*, abs/2101.03961.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.

Ananth Gottumukkala, Dheeru Dua, Sameer Singh, and Matt Gardner. 2020. Dynamic sampling strategies for multi-task reading comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 920–924, Online. Association for Computational Linguistics.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics.

Sneha Kudugunta, Yanping Huang, Ankur Bapna, Maxim Krikun, Dmitry Lepikhin, Thang Luong, and Orhan Firat. 2020. Exploring routing strategies for multilingual mixture-of-experts models.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2020. Gshard: Scaling giant models with conditional computation and automatic sharding. *ArXiv preprint*, abs/2006.16668.

Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. 2021. Base layers: Simplifying training of large, sparse models. *ArXiv preprint*, abs/2103.16716.

Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 2020. 12-in-1: Multi-task vision and language representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10434–10443. IEEE.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of* *the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Clemens Rosenbaum, Tim Klinger, and Matthew Riemer. 2018. Routing networks: Adaptive selection of non-linear functions for multi-task learning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *ArXiv preprint*, abs/1706.05098.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Xu Tan, Yi Ren, Di He, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2019. Multilingual neural machine translation with knowledge distillation. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *ArXiv preprint*, abs/2008.00401.

Chau Tran, Shruti Bhosale, James Cross, Philipp Koehn, Sergey Edunov, and Angela Fan. 2021. Facebook ai wmt21 news translation task submission. *ArXiv preprint*, abs/2108.03265.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Xinyi Wang, Yulia Tsvetkov, and Graham Neubig. 2020. Balancing training for multilingual neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8526–8537, Online. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Biao Zhang, Ankur Bapna, Rico Sennrich, and Orhan Firat. 2020. Share or not? learning to schedule language-specific capacity for multilingual translation. In *International Conference on Learning Representations*.