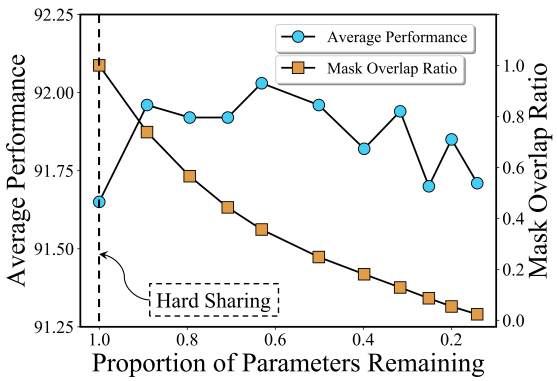

Figure 4: Average test performance and mask OR with different sparsity combinations.

with and without MTW are listed in Table 8. In most instances, the model with MTW achieves better performance.

	POS	NER	Chunking
CoNLL-2003			
Sparse sharing	95.56	90.35	91.55
− MTW	95.36	89.62	91.04
OntoNotes 5.0			
Sparse sharing	97.54	83.42	95.56
− MTW	97.53	81.15	95.48

Table 8: Test accuracy (for POS) and F1 (for NER and Chunking) on CoNLL-2003 and OntoNotes 5.0. MTW: Multi-Task Warmup.

Related Work

There are two lines of research related to our work – deep multi-task learning and sparse neural networks. Neural based multi-task learning approaches can be roughly grouped into three folds: (1) hard sharing, (2) hierarchical sharing, and (3) soft sharing. Hard sharing finds for a representation that is preferred by as many tasks as possible (**?**). In spite of its simple and parameter-efficient, it is only guaranteed to work for closely related tasks (**?**). Hierarchical sharing approaches put different level of tasks on the different network layer (**?**; **?**), which to some extent, relax the constraint about task relatedness. However, the hierarchy of tasks is usually designed by hand through the skill and insights of experts. Besides, tasks can hurt each other when they are embedded into the same hidden space (**?**). To mitigate negative transfer, soft sharing is proposed and achieves success in computer vision (**?**) and natural language processing (**?**; **?**). In spite of its flexible, soft sharing allows each task to have its separate parameters, thus is parameter inefficient. Different from these work, sparse sharing is a fine-grained parameter sharing strategy that is flexible to handle heterogeneous tasks and parameter efficient.

Our approach is also inspired by the sparsity of networks, especially the Lottery Ticket Hypothesis (**?**). **?** (**?**) finds that a subnet (winning ticket) – that can reach test accuracy comparable to the original network – can be produced through Iterative Magnitude Pruning (IMP). Further, **?** (**?**) introduce late resetting to stabilize the lottery ticket hypothesis at scale. Besides, **?** (**?**) confirms that winning ticket initializations exist in LSTM and NLP tasks. Our experiments also demonstrate that winning tickets do exist in sequence labeling tasks. In addition, it is worth noticing that sparse sharing architecture is also possible to be learned using variational dropout (**?**), l_0 regularization (**?**) or other pruning techniques.

Conclusion

Most existing neural-based multi-task learning models are done with parameter sharing, e.g. hard sharing, soft sharing, hierarchical sharing etc. These sharing mechanisms have some inevitable limitations: (1) hard sharing struggles with heterogeneous tasks, (2) soft sharing is parameter-inefficient, (3) hierarchical sharing depends on manually design. To alleviate the limitations, we propose a novel parameter sharing mechanism, named Sparse Sharing. The parameters in sparse sharing architectures are partially shared across tasks, which makes it flexible to handle loosely related tasks. To induce such architectures, we propose a simple yet efficient approach that can automatically extract subnets for each task. The obtained subnets are overlapped and trained in parallel. Our experimental results show that sparse sharing architectures achieve consistent improvement while requiring fewer parameters. Besides, our synthetic experiment shows that sparse sharing avoids negative transfer even when tasks are unrelated.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments. This work was supported by the National Key Research and Development Program of China (No. 2018YFC0831103), National Natural Science Foundation of China (No. 61672162), Shanghai Municipal Science and Technology Major Project (No. 2018SHZDZX01) and ZJLab.

Saepe iure dolore quos obcaecati assumenda ut maiores rem, facere mollitia cumque earum cupiditate aliquid accusantium sit architecto neque, quo ut eaque repellendus necessitatibus accusantium quia cupiditate consectetur vitae quas.Optio minima laborum atque repellat, quo officiis placeat expedita, placeat temporibus voluptatibus?Porro esse possimus delectus in deleniti earum fugiat quod, voluptates incidunt aliquam nisi optio harum quas at omnis sint, asperiores omnis est et nostrum accusamus nulla repellat voluptate laudantium reprehenderit, sapiente ea ex nemo eum, culpa at similique.Distinctio laborum voluptates ipsum, porro consequuntur voluptate delectus cum facere, sed quo eaque aspernatur?Quisquam perspiciatis sint ex obcaecati ducimus laborum quasi, perspiciatis pariatur molestiae

dolor fugit libero beatae, corporis amet adipisci soluta dolores in delectus veniam totam hic, quia