Fisher Mask Nodes for Language Model Merging

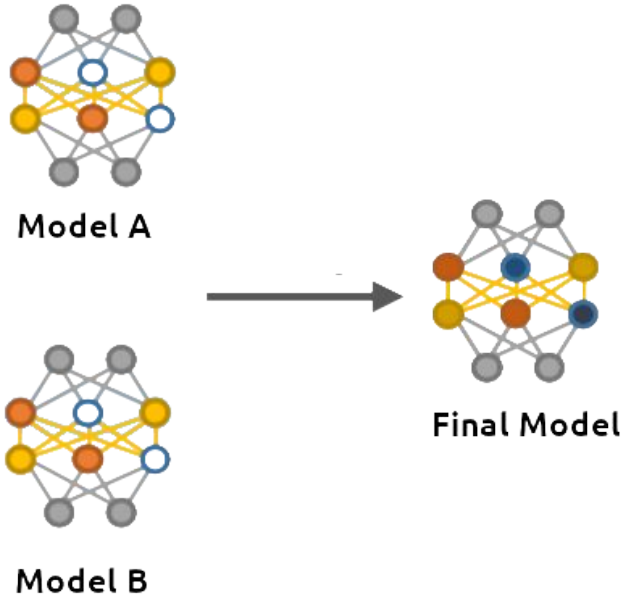
Thennal D K, Ganesh Nathan, Suchithra M S

Indian Institute of Information Technology Kottayam

LREC-COLING 2024 — Turin, Italy, May 20-25 2024

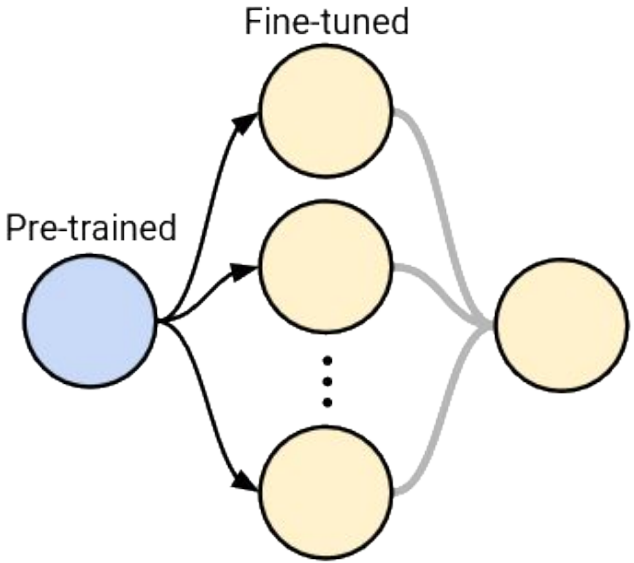
Model Merging

- Process of combining multiple deep learning models into one.
- Generally models of the same architecture but with different parameters.
- Provides a single model that can do the tasks of both Model A and Model B.



Model Merging on Finetunes

- Restrict ourselves to merge finetuned models derived from a common parent pretrained model.
- Bypasses issues regarding permutation symmetry and alignment between neural networks.
- Facilitates the creation of a single multi-task model from a collection of finetuned models on different tasks.
- Our work introduces novel method for model merging on finetunes in the context of language models.



Fisher Weight Merging

- Presented by Matena et al. [1] as a new model merging algorithm.
- The Fisher Information, a statistical measure of the importance of each parameter, is used as weights to merge models.

$$F_\theta = \mathbb{E}_x \left[\mathop{\mathbb{E}}_{y \sim p_\theta(y|x)} \nabla_\theta \log p_\theta(y|x) \nabla_\theta \log p_\theta(y|x)^T \right]$$

[1] Matena, Michael S., and Colin A. Raffel. "Merging models with fisher-weighted averaging." Advances in Neural Information Processing Systems 35 (2022): 17703-17716.

Fisher Weight Merging

- Full Fisher Information matrix is not feasible to calculate and store, so diagonal is used:

$$\hat{F}_\theta = \frac{1}{N}\sum_{i=1}^{N} \mathop{\mathbb{E}}_{y\sim p_\theta(y|x_i)} (\nabla_\theta \log p_\theta(y|x_i))^2,$$

$$\theta^{*(j)} = \frac{\sum_{i=1}^{M} \lambda_i F_i^{(j)} \theta_i^{(j)}}{\sum_{i=1}^{M} \lambda_i F_i^{(j)}},$$

- Still computationally intensive: have to calculate gradients for all parameters, with data in validation set.

Fisher Information in Pruning

- Fisher information also for pruning by Kwon et al. [2]
- Creation of mask nodes associated with the outputs of attention heads and feed-forward layer filters:

$$\mathrm{MHA}(\mathrm{x}; \mathrm{m}_l^{\mathrm{MHA}}) = \sum_{i=1}^{H} m_{l,i}^{\mathrm{MHA}} \circ \mathrm{Att}_i(\mathrm{x}),$$

$$\mathrm{FFN}(\mathrm{x}; \mathrm{m}_l^{\mathrm{FFN}}) = \Big(\sum_{i=1}^{N} m_{l,i}^{\mathrm{FFN}} \circ \mathrm{W}_{:,i}^{(2)} \sigma(\mathrm{W}_{i,:}^{(1)} \mathrm{x} + b_i^{(1)})\Big) + b^{(2)},$$

- Use the Fisher Information of the mask nodes to prune heads/filters.

[2] Kwon, Woosuk, et al. "A fast post-training pruning framework for transformers." Advances in Neural Information Processing Systems 35 (2022): 24101-24116.

Our Method

- Hypothesis: Fisher information of mask nodes can represent the importance of the parameters enclosed by the mask.
- Use mask node Fisher information as a proxy for the Fisher information of the corresponding block and merge parameters like in Fisher Weight Merging.
- Remove dependency on validation set and use train set for wider applicability.

Our Method

Define the masks similar to Kwon et al.

$$\text{For } h \in \{1, ..., H\}:$$
$$Y_h = \texttt{Attention}(X; W^{h,l}_{qkv})$$
$$Y = \{Y_1, Y_2, ..., Y_H\}$$
$$X \leftarrow X + W^o(Y \odot m^l_{mha}) + b^o$$

$$X \leftarrow X + W^l_{mlp2}\texttt{GELU}(m^l_{mlp} \odot W^l_{mlp1}X + b^l_{mlp1}) + b^l_{mlp2}$$

Our Method

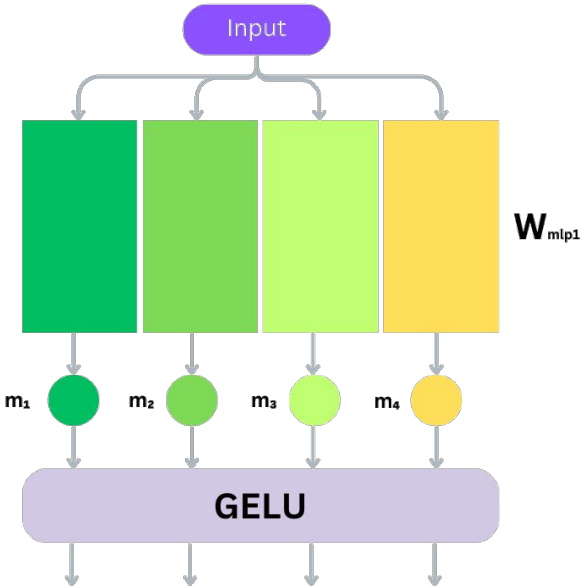
- Formulate the diagonal approx. of the Fisher Information Matrix.
- Associate F_j of each parameter with the Fisher information I_{ii} of the mask node m_i.

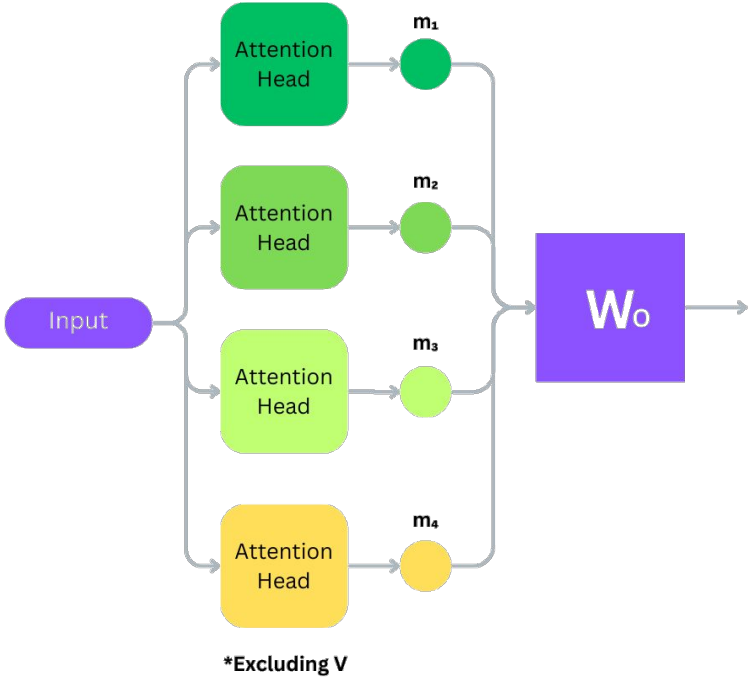
$$\theta^* = \frac{\sum_{j=1}^{M} \lambda_j F_j \theta_j}{\sum_{j=1}^{M} \lambda_j F_j} \qquad I_{ii} := \frac{1}{|D|} \sum_{(x,y)\in D} \left(\frac{\partial}{\partial m_i} L(x, y; 1) \right)^2$$

Associate Masks with Parameters

1. If the parameter is in the query and key matrices of a particular attention head the Fisher information of the corresponding mask taken. Including the value parameters here was found to empirically lead to a lower performance.
2. If the parameter is in a particular row of the intermediate layer of the feed-forward block, the Fisher information of the corresponding mask is taken.
3. For all other parameters, take $F_j = 1$, resulting in merging equivalent to plain averaging.

Associate Masks with Parameters





Results

Model (↓)	Averaging	Fisher (2022)			Ours		
No. of Samples (→)		128	2,048	32,768	128	2,048	32,768
BERT (Base)	91.2	92.0	91.7	91.9	**95.2** (+4.0)	92.8 (+1.6)	93.7 (+2.5)
BERT (Large)	85.8	87.0	86.1	89.2	**90.1** (+4.3)	88.1 (+2.3)	87.9 (+2.1)
BERT (Tiny)	**90.8**	89.6	89.2	90.4	86.1 (-4.7)	88.4 (-2.4)	88.7 (-2.1)
RoBERTa	86.2	91.5	88.5	88.5	**92.7** (+6.5)	89.4 (+3.2)	88.8 (+2.6)

Table 1: Normalized and aggregated metrics across tasks.

Results - Performance

Task (↓)	Ours, Time (s)			
Model (→)	**Tiny**	**Base**	**Large**	**RoBERTa**
MNLI	0.056	1.121	4.591	1.195
QQP	0.063	0.670	2.405	0.630
QNLI	0.058	1.117	4.321	1.119
SST2	0.046	0.565	3.624	0.516
MRPC	0.048	0.972	3.624	0.970
RTE	0.053	1.251	5.047	1.307
Mean	**0.055**	**0.959**	**4.003**	**0.993**

Table 2: Time taken for our method with 128 samples across all tasks and models.

Task (↓)	Fisher (2022), Time (s)			
Model (→)	**Tiny**	**Base**	**Large**	**RoBERTa**
MNLI	19.149	71.486	284.276	73.203
QQP	17.472	55.193	201.505	56.623
QNLI	16.510	55.462	201.746	56.457
SST2	16.836	56.132	203.036	57.657
MRPC	16.376	55.973	202.999	57.304
RTE	17.011	55.851	202.527	57.856
Mean	**17.101**	**58.009**	**214.299**	**59.383**

Table 3: Time taken for full Fisher calculation with 128 samples across all tasks and models.

Conclusion

- We introduce a novel method for model merging derived from Fisher-weighted merging.
- Our method is comparatively efficient, with elapsed time speed-ups between **57.4x** and **321.7x** across models.
- Our method does not rely on the validation set, making it widely applicable
- Future work needs to be done to evaluate 3+ model merging, along with novel mask architectures and extensions to different architectures

Thank You

Thennal D K
thennal10@gmail.com
thennal21bcs14@iiitkottayam.ac.in
@ThennalDK