Intermediary Pruning of Deep Neural Nets using Fisher Information Matrices

Özgür Soysal
soysal@stanford.edu

1 Abstract

Deep neural networks often achieve state-of-the-art performance by relying on substantial parameter counts, which can make training and deployment costly in memory and compute. Pruning aims to reduce this redundancy by removing parameters that contribute little to the model's predictions. This paper proposes a novel mid-training pruning scheme that uses the Fisher Information Matrix (FIM) and gradient directions to identify removable parameters while training is still ongoing. Rather than pruning only after convergence, this method interleaves optimization with pruning by estimating parameter importance via efficient estimation of an augmented FIM and progressively mask low-score weights as learning continues. This produces a sparsity trajectory that adapts to the model's evolving sensitivity, enabling earlier and more aggressive compressions with reduced performance loss. This paper benchmarks the approach against unpruned baselines, evaluating final accuracy, training stability, and find that augmented FIM based mid-training pruning better preserves accuracy.

2 Problem Definition

Let $F_{\mathcal{A}}$ be the set of neural networks $\{f_\theta | \theta \in \mathcal{A}\}$ with architecture $\mathcal{A}$. The objective of neural network pruning / compression as used in this report is to find a function $g : \mathcal{A} \to \mathcal{A}$ such that $\mathcal{L}_{f_{g(\theta)}}(\mathcal{X}_{test}) \approx \mathcal{L}_{f_\theta}(\mathcal{X}_{test})$ with g being as restrictive as possible.

3 Related Works

The theoretical basis of Fisher Information based pruning stems from LeCun's 1990 paper "Optimal Brain Damage" which approximated the change in loss $\delta\mathcal{L}$ caused by pruning a parameter as a function of the Hessian diagonal [1]. This was generalized by Hassibi and Stork in "Optimal Brain Surgeon" which accounted for off-diagonal Hessian elements to update the remaining weights [2].

While these are successful methods, they require estimating the full Hessian and thus are computationally expensive with $O(d^3)$ time complexity. Consequently, much of the subsequent literature deals with formulating efficient approximations of the Hessian, commonly utilizing the Fisher Information Matrix of parameters under the assumption that the model is near convergence (where $\mathcal{H} \approx \mathcal{I}$).

Martens and Grosse introduced Kronecker-Factored Approximate Curvature (K-FAC) for optimization, which Wang et al. (2019) adapted for pruning in EigenDamage. By approximating the FIM as a block-diagonal matrix where each block corresponds to a layer's weights, they enabled pruning that respects the correlation within layers [3].

A breakthrough in efficiency was proposed by Singh and Alistarh with WoodFisher. They utilized the Woodbury matrix identity to incrementally estimate the inverse Hessian (or Fisher) using a limited number of gradients. This allowed for accurate one-shot pruning of large ResNet models, outperforming magnitude-based baselines by a large margin [4].

Direct applications of Fisher Information for channel and weight selection have seen success in computer vision applications. Fisher Pruning formulated a greedy pruning metric based on the expected increase in loss approximated by the Fisher information, applying it successfully to dense networks for gaze prediction [5]. More recently, Group Fisher Pruning extended this to coupled channels in residual networks like ResNets and DenseNets, utilizing a unified metric to prune entire structures while maintaining gradient flow [6].

Kurtić extended the WoodFisher framework to Transformer architectures, demonstrating that second-order information is critical for pruning attention heads without large degradations in accuracy [7].

Recent a method called RAHP which was submitted to ICLR 2026 combined Fisher Information with other metrics (such as CLEVER scores) to prune attention heads in a way that preserves not just accuracy, but also the robustness of the model[8].

All of the aformentioned frameworks mainly work with the individual elements of the FIM and thus fail to capture how cross-parameter interactions effect the performance of the model. Additionally, FIM can be used to manufacture informative linear maps of a layer to capture more of the information than magnitude based methods of the same sparsity level.

4 Fisher Information Based Intermediary Pruning

The compression of deep neural networks can framed as an optimization problem on a Riemannian manifold. Let $\mathcal{M} = \{p_\theta(x) \in \mathbb{R}^d\}$ be the manifold of probability distributions parameterized by the network. The Fisher Information Matrix (FIM), $F(\theta)$, defines the Riemannian metric on $\mathcal{M}$.

Already existing methods solve the following trace maximization problem to find a projection $P \in \mathbb{R}^{d\times k}$:

$$\max_P \operatorname{Tr}(P^T F(\theta) P) \tag{1}$$

This formulation assumes that parameter importance is equivalent to local sensitivity of the training loss function. However, this equivalence does not exist when the current parameter θ_t is distant from the optimal θ^*. In such regimes, the "importance" of a parameter must account also for its utility in reducing the training loss and hence the gradient. This paper introduces a method to incorporate the learning objective into the pruning strategy.

4.1 Pruning Objective

Let θ^* be the optimal parameters for a model. The pruning loss $\mathcal{L}_p$ can be decomposed as:

$$\mathcal{L}_p = -\operatorname{Tr}((P^T F P)) + \lambda \left\|(I - PP^T)(\theta^* - \theta_t)\right\|_F^2. \tag{2}$$

Standard Fisher pruning minimizes the first term (which turns out to be the Cramer-Rao Lower Bound) whereas this paper's objective is to minimize the weighted sum.

4.2 Augmented Fisher Matrix

The true difference $\Delta\theta = \theta^* - \hat{\theta}$ is unavailable. To approximate it, a vector v is defined as the natural gradient direction required to move the current parameters θ_t toward the optimal parameters θ^* on a validation set $\mathcal{D}_{val}$.

Definition 1.

$$v \triangleq F^{-1}\nabla_\theta \mathcal{L}(\theta; \mathcal{D}_{val}) \approx (\theta^* - \theta_t) \tag{3}$$

After this definition, a projection P is seeked that preserves the curvature while simultaneously minimizing the projection error of the vector v.

$$\mathcal{L}(P) = -\operatorname{Tr}(P^T F P) + \lambda ||(I - PP^T)v||_F^2 \tag{4}$$

Here the second term penalizes losing the correction term v. The norm $||\cdot|||_F^2$ is the metric induced norm $x^T F x$.

Theorem 1. *Minimizing the loss $\mathcal{L}(P)$ is equivalent to performing PCA on the augmented Fisher Matrix $\tilde{F}$:*

$$\tilde{F} = F + \lambda(gg^T) \tag{5}$$

where $g = \nabla_\theta \mathcal{L}(\theta; \mathcal{D}_{val})$ is the Euclidean gradient on the validation set.

However, directly forming $\tilde{F}$ is computationally intractable. A scoring method that leverages the K-FAC structure of F to approximate $\tilde{F}$ efficiently is useful here.

4.3 Kronecker-Factored Approximate Curvature

The Kronecker-Factored Approximate Curvature assumption is utilized, accordingly the FIM for a layer is approximated as [3]:

$$F \approx A \otimes G \tag{6}$$

where $A = \mathbb{E}[xx^T]$ is the input covariance and $G = \mathbb{E}[\nabla_y \mathcal{L} \nabla_y \mathcal{L}^T]$ is the gradient covariance.

4.4 Efficient Calculation of Scores

The first step is to compute the eigendecomposition of the K-FAC factors on the training set:

$$A = Q_A \Lambda_A Q_A^T \tag{7}$$

$$G = Q_G \Lambda_G Q_G^T \tag{8}$$

The eigenvectors $Q_A \otimes Q_G$ form the natural basis of the layer.

Then the mean gradient matrix is computed $\bar{\mathcal{J}} = \mathbb{E}_{val}[\nabla_W \mathcal{L}]$ for the layer. This gradient is rotated into the natural basis:

$$\hat{\mathcal{J}} = Q_G^T \bar{\mathcal{J}} Q_A \tag{9}$$

Each entry $\hat{\mathcal{J}}_{ij}$ represents the magnitude of the bias correction required for the interaction between the i-th output mode and j-th input mode.

At the end, the eigenvalues of $\tilde{F}$ are approximated by the sum of the Fisher eigenvalues and the projected gradient energy. For each index pair (i, j):

$$S_{ij} = (\lambda_{G,i} \cdot \lambda_{A,j}) + \lambda(\hat{\mathcal{J}}_{ij})^2 \tag{10}$$

The final method is to fist compute S_{ij} for all potential connections. Then select the top-k indices (i, j) and construct the compressed layer using the selected eigenvectors and the corresponding rotated weights. In experiments, it turned out that using the gradients g to calculate v is too noisy for ResNet20, thus the momentum m_t is taken from the optimizer and used in place of g.

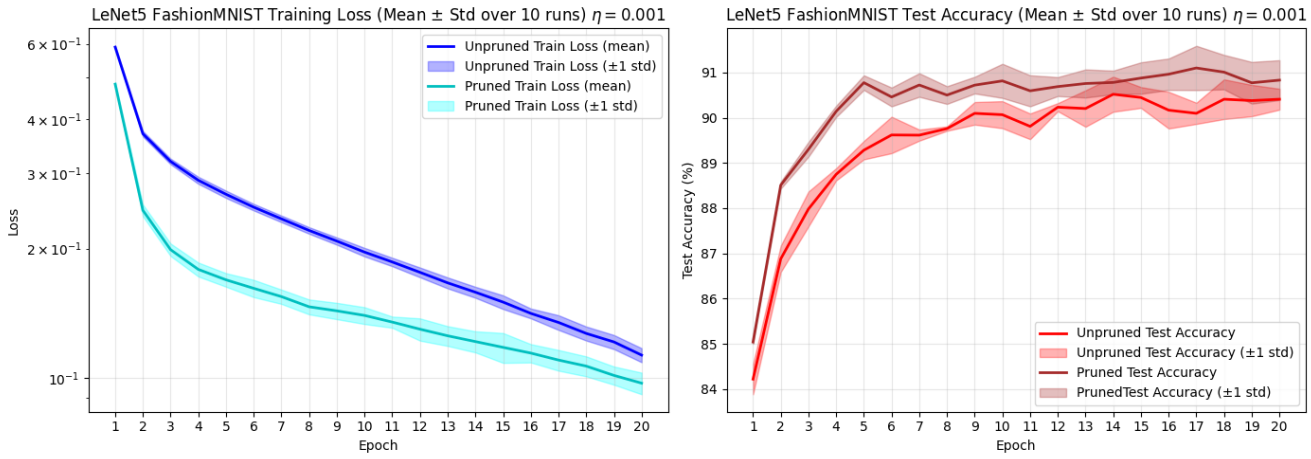

Figure 1: LeNet5 on FashionMNIST. We prune 10% of the connections at each epoch, retaining only around 12% of the connections at the end. The red benchmark is the unpruned network. The mean and standard deviation are calculated over 10 runs. The optimizer of choice is Adam [9]. $\eta = 0.001$. $\lambda = 0.1$

5 Results

The proposed method has only one hyperparameter, λ which the method is very insensitive to. Below, you can see the hyperparameter search and training results for LeNet5 on the Fashion MNIST dataset and ResNet20 on CIFAR10.

5.1 LeNet5 on FashionMNIST

Figure 1 illustrates the performance of the LeNet5 architecture trained on the FashionMNIST dataset, comparing the proposed Fisher-based intermediary pruning method against an unpruned baseline [10]. In this experiment an aggressive pruning schedule where 10% of the active connections are removed at each epoch is utilized. Consequently, the final model retains approximately 12% of its original connections. To ensure statistical robustness, the reported metrics represent the mean and standard deviation calculated over 10 independent runs.

The training dynamics, shown in Figure 1, demonstrate that the pruned model achieves a consistently lower training loss compared to the unpruned network. Despite the continuous removal of parameters, the pruned trajectory (cyan) drops below the unpruned baseline (blue) early in the training process. This indicates that the Fisher-guided removal of weights effectively reduces redundancy without hindering the model's ability to fit the training data.

In terms of generalization, Figure 1 reveals that the pruning strategy preserves test accuracy. The pruned model outperforms the unpruned baseline throughout the 20 epochs and achieves a final accuracy that is competitive with, and at points higher than, the fully connected network. This result validates that the method successfully identifies and retains the most informative parameters while not hindering further training, allowing for significant compression without performance degradation.

5.1.1 Sensitivity to λ

Running a hyperparameter search in $\lambda \in [0.01, 0.5]$, it is seen that the peak test accuracy is almost independent with λ. This shows that in averse compression rates, very weakly bleeding the displacement vector g into the augmented FIM suffices.

5.2 ResNet20 on CIFAR10

Figure 2 presents the results for the ResNet20 architecture trained on the CIFAR10 dataset [11]. For this deeper architecture, a more conservative pruning schedule was adopted, removing 5% of connections at each epoch to retain approximately 21% of the total connections by the end of training. Due to the noise observed when using raw gradients for this architecture, the method was adapted to use the optimizer's momentum m_t in place of the gradient g for the natural gradient approximation.

The training loss trajectory mirrors the trend seen in LeNet5, where the pruned network (cyan) achieves a significantly lower loss than the unpruned baseline (blue). This highlights that the pruned model converges to a tighter fit on the training data.

While the pruned model remains competitive, reaching approximately 85% accuracy, the discrepancy between the superior training loss and the slightly degraded test accuracy suggests that the remaining parameters may be overfitting to the training distribution.

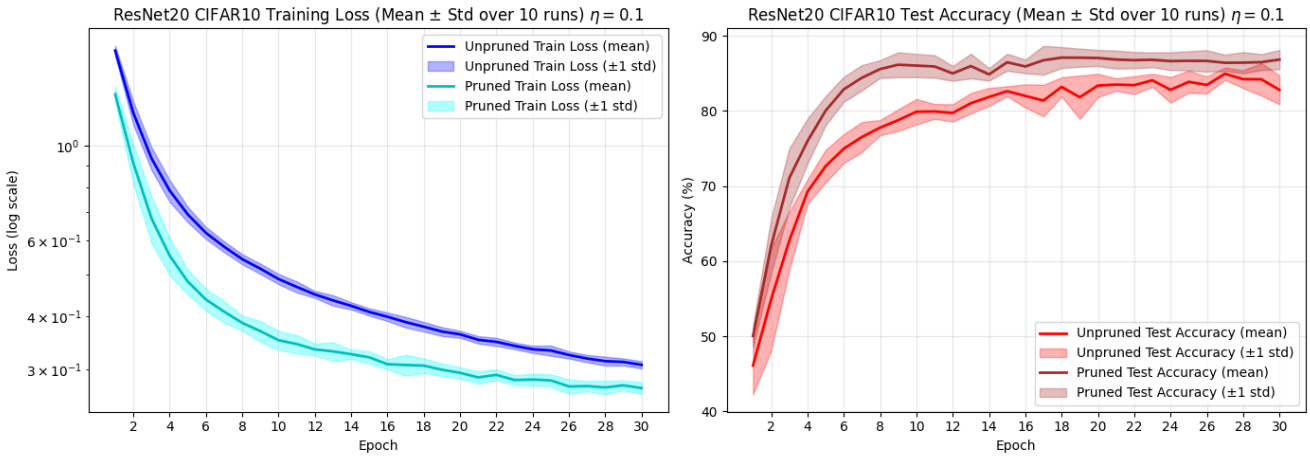

Figure 2: ResNet20 on CIFAR10. We prune 5% of the connections at each epoch, retaining only around 21% of the connections at the end. The red benchmark is the unpruned network. The mean and standard deviation are calculated over 10 runs. The optimizer of choice is Adam. $\eta = 0.1$. $\lambda = 0.25$

5.2.1 Sensitivity to λ

Unlike in LeNet5, residual connections improve the importance of using the augmented FIM rather than the naive FIM. The residual connections are very prone to early deletion by the FIM whereas the λgg^T term makes sure they survive the early epochs. However, gradually as the model starts to cenverge, the residual connections become informative enough. Due to these reasons, a higher λ is needed for ResNet20. The optimal value is found to be at $\lambda = 0.25$ however higher values also work well.

6 Discussion and Conclusion

This study introduced a novel intermediary pruning framework that leverages the Fisher Information Matrix (FIM) augmented by a gradient correction term. By interleaving optimization with pruning, the method aims to identify and retain parameters that are critical not just for carrying information for the training loss, but also for facilitating the trajectory toward the optimal solution.

The experimental results validate the performance of this approach across different architectures. For the LeNet5 model on FashionMNIST, the method successfully removed approximately 88% of the conections while matching or exceeding the test accuracy of the fully connected baseline. This confirms that for moderate-scale networks, the augmented Fisher Matrix successfully captures a better subset of weights than standard magnitude based baselines do, allowing for aggressive compression without performance degradation in subsequent training stages.

However, the application to deeper architectures like ResNet20 on CIFAR10 highlighted distinct challenges. While the method achieved superior training optimization shown by the significantly lower training loss compared to the unpruned baseline, it exhibited a generalization gap in test accuracy. This behavior suggests that the aggressive compression of the model may lead to the removal of "compensating connections" that prevent the model from learning the noise. Furthermore, the necessity of substituting the raw gradient g with the optimizer's momentum m_t to mitigate noise indicates that accurate estimation of the natural gradient direction remains an important task.

In conclusion, Fisher-guided intermediary pruning offers a promising mechanism for model compression, particularly in reducing parameter count without compromising future training accuracy. Future work may focus on refining the natural gradient augmentation potentially through improved regularization or adaptive sparsity schedules.

References

[1] Y. LeCun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 2, 1990.

[2] B. Hassibi and D. G. Stork, "Optimal brain surgeon," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 5, 1993.

[3] J. Martens and R. Grosse, "Optimizing neural networks with kronecker-factored approximate curvature," in *International Conference on Machine Learning (ICML)*, 2015, pp. 2408–2417.

[4] S. P. Singh and D. Alistarh, "Woodfisher: Efficient second-order approximation for neural network compression," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 18 098–18 109.

[5] L. Theis, I. Korshunova, A. Tejani, and F. Huszár, "Faster gaze prediction with dense networks and fisher pruning," in *arXiv preprint arXiv:1801.05787*, 2018.

[6] L. Liu, S. Zhang, Z. Kuang, A. Zhou, J.-H. Xue, X. Wang, Y. Chen, W. Yan, G. Wan, and W. Zhang, "Group fisher pruning for practical network compression," in *International Conference on Machine Learning (ICML)*, 2021, pp. 6921–6932.

[7] E. Kurtić, D. Campos, T. Nguyen, E. Frantar, M. Kurtz, B. Fineran, M. Goin, and D. Alistarh, "The optimal bert surgeon: Scalable and accurate second-order pruning for large language models," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022, pp. 4163–4181.

[8] D. Levin and G. Singer, "Rahp: Robustness-aware head pruning for certified transformer models," OpenReview, ICLR 2026 Conference (withdrawn submission), Sep. 2025. [Online]. Available: https://openreview.net/forum?id=dsfOgH61ii

[9] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[10] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.