Supplemental Document:
On Implicit Filter Level Sparsity in Convolutional Neural Networks

Dushyant Mehta[1,3] Kwang In Kim[2] Christian Theobalt[1,3]
[1]MPI For Informatics [2]UNIST [3]Saarland Informatics Campus

In this supplemental document, we provide additional experiments that show how filter level sparsity manifests under different gradient descent flavours and regularization settings (Sec. 1), and that it even manifests with Leaky ReLU. We also show the emergence of feature selectivity in Adam in multiple layers, and discuss its implications on the extent of sparsity (Sec. 2). In Section 3 we consider additional hyperparameters that influence the emergent sparsity. In Section 4 we provide specifics for some of the experiments reported in the main document.

1. Layer-wise Sparsity in *BasicNet*

In Section 2.3 and Table 2 in the main paper, we demonstrated that for BasicNet on CIFAR-100, Adam shows feature sparsity in both early layers and later layers, while SGD only shows sparsity in the early layers. We establish in the main paper that Adam learns selective features in the later layers which contribute to this additional sparsity. In Table 1 we show similar trends in layer-wise sparsity also emerge when trained on CIFAR-10.

Sparsity with AMSGrad: In Table 2 we compare the extent of sparsity of Adam with AMSGrad [2]. Given that AMSGrad tracks the long term history of squared gradients, we expect the effect of L2 regularization in the low gradient regime to be dampened, and for it to lead to less sparsity. For BasicNet, on CIFAR-100, with L2 regularization of 10^{-4}, AMSGrad only shows sparsity in the later layers, and overall only 13% of features are inactive. For a comparable test error for Adam, 47% of the features are inactive. In Table 4 we show the feature sparsity by activation and by γ for BasicNet with AMSGrad, Adamax and RMSProp, trained for CIFAR-10/100.

Sparsity with Leaky ReLU: Leaky ReLU is anecdotally [1] believed to address the 'dying ReLU' problem by preventing features from being inactivated. The cause of feature level sparsity is believed to be the accidental inactivation of features, which gradients from Leaky ReLU can help revive. We have however shown there are systemic processes underlying the emergence of feature level sparsity, and those would continue to persist even with Leaky ReLU. Though our original definition of feature selectivity does not apply here, it can be modified to make a distinction between data points which produce positive activations for a feature vs. the data points that produce a negative activation. For typical values of the negative slope (0.01 or 0.1) of Leaky ReLU, the more selective features (as per the updated definition) would continue to see lower gradients than the less selective features, and would consequently see relatively higher effect of regularization. For BasicNet trained on CIFAR-100 with Adam, in Table 2 we see that using Leaky ReLU has a minor overall impact on the emergent sparsity. See Section 3 for more effective ways of reducing filter level sparsity in ReLU networks.

2. On Feature Selectivity in Adam

In Figure 1, we show the the distribution of the scales (γ) and biases (β) of layers C6 and C5 of *BasicNet*, trained on CIFAR-100. We consider SGD and Adam, each with a low and high regularization value. For both C6 and C5, Adam learns exclusively negative biases and positive scales, which results in features having a higher degree of selectivity (i.e, activating for only small subsets of the training corpus). In case of SGD, a subset of features learns positive biases, indicating more universal (less selective) features.

Figure 2 shows feature selectivity also emerges in the later layers when trained on CIFAR-10, in agreement with the results presented for CIFAR-100 in Fig. 3 of the main paper.

Higher feature selectivity leads to parameters spending more iterations in a low gradient regime. In Figure 3, we show the effect of the coupling of L2 regularization with the update step of various adaptive gradient descent approaches in a low gradient regime. Adaptive gradient approaches exhibit strong regularization in low gradient regime even with low regularization values. This disproportionate action of the regularizer, combined with the propensity of certain adaptive gradient methods for learning selective features, results in a higher degree of feature level sparsity with adaptive approaches than vanilla SGD, or when using weight decay.

3. Effect of Other Hyperparameters on Sparsity

Having shown in the main paper and in Sec. 2 that feature selectivity results directly from negative bias (β) values when the scale values (γ) are positive, we investigate the effect of β initialization value on the resulting sparsity. As shown in Table 3 for BasicNet trained with Adam on CIFAR 100, a slightly negative initialization value of -0.1 does not affect the level of sparsity. However, a positive initialization value of 1.0 results in higher sparsity. This shows that attempting to address the emergent sparsity by changing the initialization of β may be counter productive.

We also investigate the effect of scaling down the learning rate of γ and β compared to that for the rest of the network (Table 3). Scaling down the learning rate of γs by a factor of 10 results in a significant reduction of sparsity. This can likely be attributed to the decrease in effect of the L2 regularizer in the low gradient regime because it is directly scaled by the learning rate. This shows that tuning the learning of γ can be more effective than Leaky ReLU at controlling the emergent sparsity. On the other hand, scaling down the learning rate of βs by a factor of 10 results in a slight increase in the extent of sparsity.

4. Experimental Details

For all experiments, the learned BatchNorm scales (γ) are initialized with a value of 1, and the biases (β) with a value of 0. The reported numbers for all experiments on CIFAR10/100 are averaged over 3 runs. Those on TinyImageNet are averaged over 2 runs, and for ImageNet the results are from 1 run. On CIFAR10/100, VGG-16 follows the same learning rate schedule as *BasicNet*, as detailed in Section 2.1 in the main paper.

Table 1. Layerwise % filters pruned from BasicNet trained on CIFAR10, based on the $|\gamma| < 10^{-3}$ criteria. Also shown are pre-pruning and post-pruning test error, and the % of *convolutional* parameters pruned. C1-C7 indicate Convolution layer 1-7, and the numbers in parantheses indicate the total number of features per layer. Average of 3 runs. Also see Table 2 in the main document.

CIFAR10					% Sparsity by γ or % Filters Pruned								% Param	
		Train	Test	Test	C1	C2	C3	C4	C5	C6	C7	Total	Pruned	Pruned
		Loss	Loss	Err	(64)	(128)	(128)	(256)	(256)	(512)	(512)	(1856)	(4649664)	Test Err.
Adam	L2: 1e-3	0.29	0.41	13.1	59	57	42	74	76	97	98	83	97	13.5
	L2: 1e-4	0.06	0.43	10.5	44	22	6	45	54	96	95	70	90	10.5
	WD: 2e-4	0.22	0.42	13.4	57	27	9	19	46	77	91	60	83	13.4
	WD: 1e-4	0.07	0.42	11.2	45	4	0	0	14	51	78	40	63	11.2
SGD	L2: 1e-3	0.62	0.64	21.8	86	61	53	46	65	4	0	27	38	21.8
	L2: 5e-4	0.38	0.49	16.3	68	16	9	9	24	0	0	9	13	16.5
	WD: 1e-3	0.61	0.63	21.6	85	60	51	46	66	4	0	27	38	21.6
	WD: 5e-4	0.38	0.46	15.8	69	19	7	7	23	0	0	8	13	16.1

Table 2. Layerwise % filters pruned from BasicNet trained on CIFAR100, based on the $|\gamma| < 10^{-3}$ criteria. Also shown are pre-pruning and post-pruning test error. C1-C7 indicate Convolution layer 1-7, and the numbers in parantheses indicate the total number of features per layer. Average of 3 runs.

Adam vs AMSGrad (ReLU)					% Sparsity by γ or % Filters Pruned								
		Train	Test	Test	C1	C2	C3	C4	C5	C6	C7	Total	Pruned
		Loss	Loss	Err	(64)	(128)	(128)	(256)	(256)	(512)	(512)	(1856)	Test Err.
Adam	L2: 1e-3	1.06	1.41	39.0	56	47	43	68	72	91	85	76	39.3
	L2: 1e-4	0.10	1.98	36.6	41	20	9	33	34	67	55	47	36.6
AMSGrad	L2: 1e-2	3.01	2.87	71.9	79	91	91	96	96	98	96	95	71.9
	L2: 1e-4	0.04	1.90	35.6	0	0	0	0	1	25	23	13	35.6
	L2: 1e-6	0.01	3.23	40.2	0	0	0	0	0	0	0	0	40.2

Adam With Leaky ReLU				% Sparsity by γ or % Filters Pruned								
	Train	Test	Test	C1	C2	C3	C4	C5	C6	C7	Total	Pruned
NegSlope=0.01	Loss	Loss	Err	(64)	(128)	(128)	(256)	(256)	(512)	(512)	(1856)	Test Err.
L2: 1e-3	1.07	1.41	39.1	49	40	39	62	61	81	85	70	39.4
L2: 1e-4	0.10	1.99	36.8	33	20	9	31	29	55	53	41	36.8
NegSlope=0.1												
L2: 1e-4	0.14	2.01	37.2	38	30	21	34	31	55	52	43	37.3

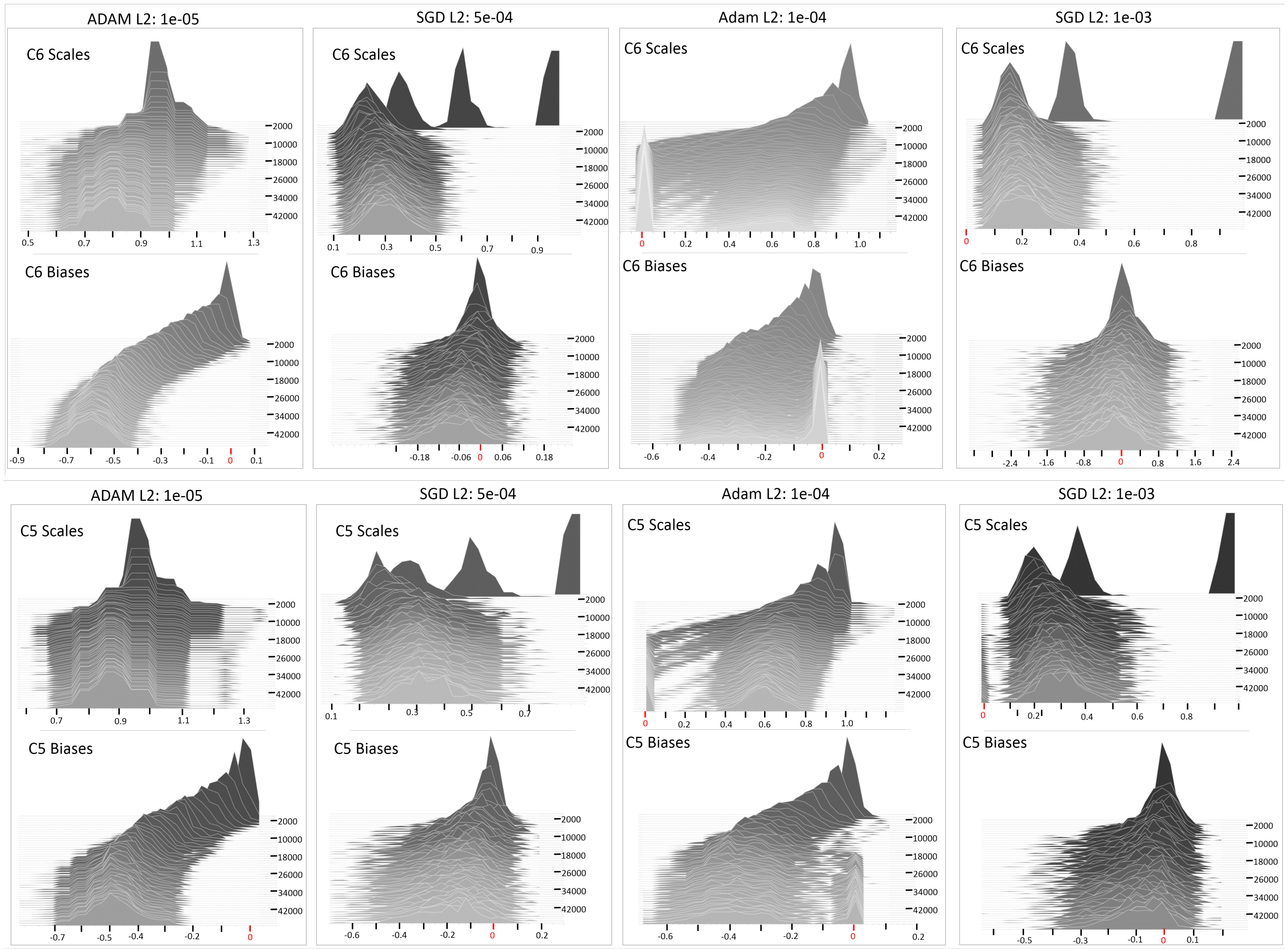

Figure 1. **Emergence of Feature Selectivity with Adam (Layer C6 and C5)** The evolution of the learned scales (γ, top row) and biases (β, bottom row) for layer C6 (top) and C5 (bottom) of *BasicNet* for Adam and SGD as training progresses, in both low and high L2 regularization regimes. Adam has distinctly negative biases, while SGD sees both positive and negative biases. For positive scale values, as seen for both Adam and SGD, this translates to greater feature selectivity in the case of Adam, which translates to a higher degree of sparsification when stronger regularization is used.

Table 3. Layerwise % filters pruned from BasicNet trained on CIFAR100, based on the $|\gamma| < 10^{-3}$ criteria. Also shown are pre-pruning and post-pruning test error. C1-C7 indicate Convolution layer 1-7, and the numbers in parantheses indicate the total number of features per layer. We analyse the effect of different initializations of βs, as well as the effect of different relative learning rates for γs and βs, when trained with Adam with L2 regularization of 10^{-4}. Average of 3 runs.

				% Sparsity by γ or % Filters Pruned								
	Train Loss	Test Loss	Test Err	C1 (64)	C2 (128)	C3 (128)	C4 (256)	C5 (256)	C6 (512)	C7 (512)	Total (1856)	Pruned Test Err.
Baseline (γ_{init}=1, β_{init}=0)	0.10	1.98	36.6	41	20	9	33	34	67	55	46	36.6
γ_{init}=1, β_{init}=−0.1	0.10	1.98	37.2	44	20	10	34	32	68	54	46	36.5
γ_{init}=1, β_{init}=1.0	0.14	2.04	38.4	47	29	25	36	46	69	61	53	38.4
Different Learning Rate Scaling for β and γ												
LR scale for γ: 0.1	0.08	1.90	35.0	16	6	1	13	20	52	49	33	35.0
LR scale for β: 0.1	0.12	1.98	37.1	42	26	21	41	48	70	55	51	37.1

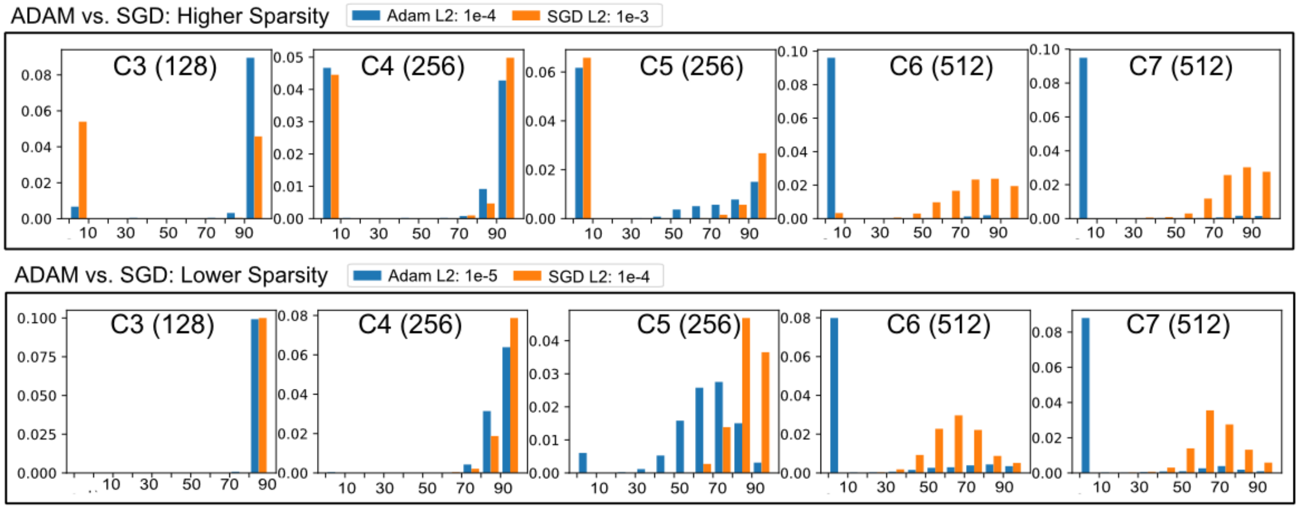

Figure 2. **Layer-wise Feature Selectivity** Feature universality for CIFAR 10, with Adam and SGD. X-axis shows the universality and Y-axis (×10) shows the fraction of features with that level of universality. For later layers, Adam tends to learn less universal features than SGD, which get pruned by the regularizer. Please be mindful of the differences in Y-axis scales between plots. Figure 3 in the main document shows similar plots for CIFAR100.

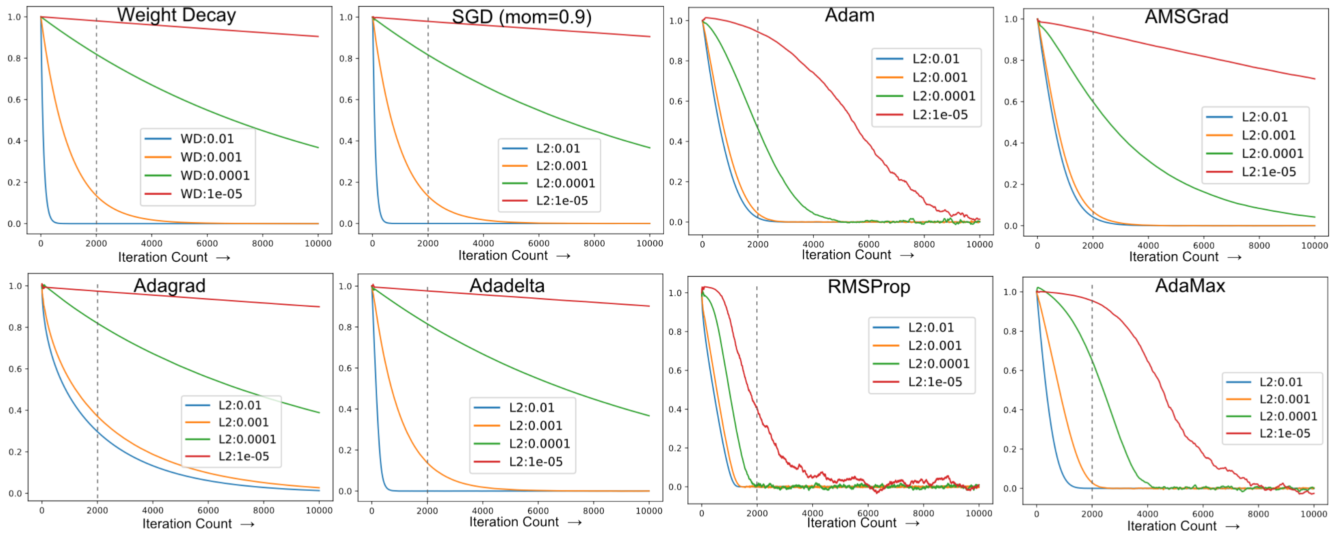

Figure 3. The action of regularization on a scalar value for a range of regularization values in the presence of simulated low gradients drawn from a mean=0, std=10^{-5} normal distribution. The gradients for the first 100 iterations are drawn from a mean=0, std=10^{-3} normal distribution to emulate a transition into low gradient regime rather than directly starting in the low gradient regime. The scalar is initialized with a value of 1. The learning rates are as follows: SGD(momentum=0.9,lr=0.1), ADAM(1e-3), AMSGrad(1e-3), Adagrad(1e-2), Adadelta(1.0), RMSProp(1e-3), AdaMax(2e-3). The action of the regularizer in low gradient regime is only one of the factors influencing sparsity. Different gradient descent flavours promote different levels of feature selectivity, which dictates the fraction of features that fall in the low gradient regime. Further, the optimizer and the mini-batch size affect together affect the duration different features spend in low gradient regime.

For experiments on ObjectNet3D [4] renderings, we use objects from the following 30 classes: aeroplane, bed, bench, bicycle, boat, bookshelf, bus, camera, chair, clock, eyeglasses, fan, flashlight, guitar, headphone, jar, kettle, keyboard, laptop, piano, racket, shoe, sofa, suitcase, teapot, toaster, train, trophy, tub, and wheelchair. The objects are rendered to 64x64 pixel images by randomly sampling (uniformly) the azimuth angle between -180 and 180 degrees, and the elevation between -15 and +45 degrees. The renderings are identical between the cluttered and the plain set, with the backgrounds for the cluttered set taken from the Cubism subset from PeopleArt [3] dataset. See Figure 4. The network structure and training is similar to that for CIFAR10/100, and a batch size of 40 is used.

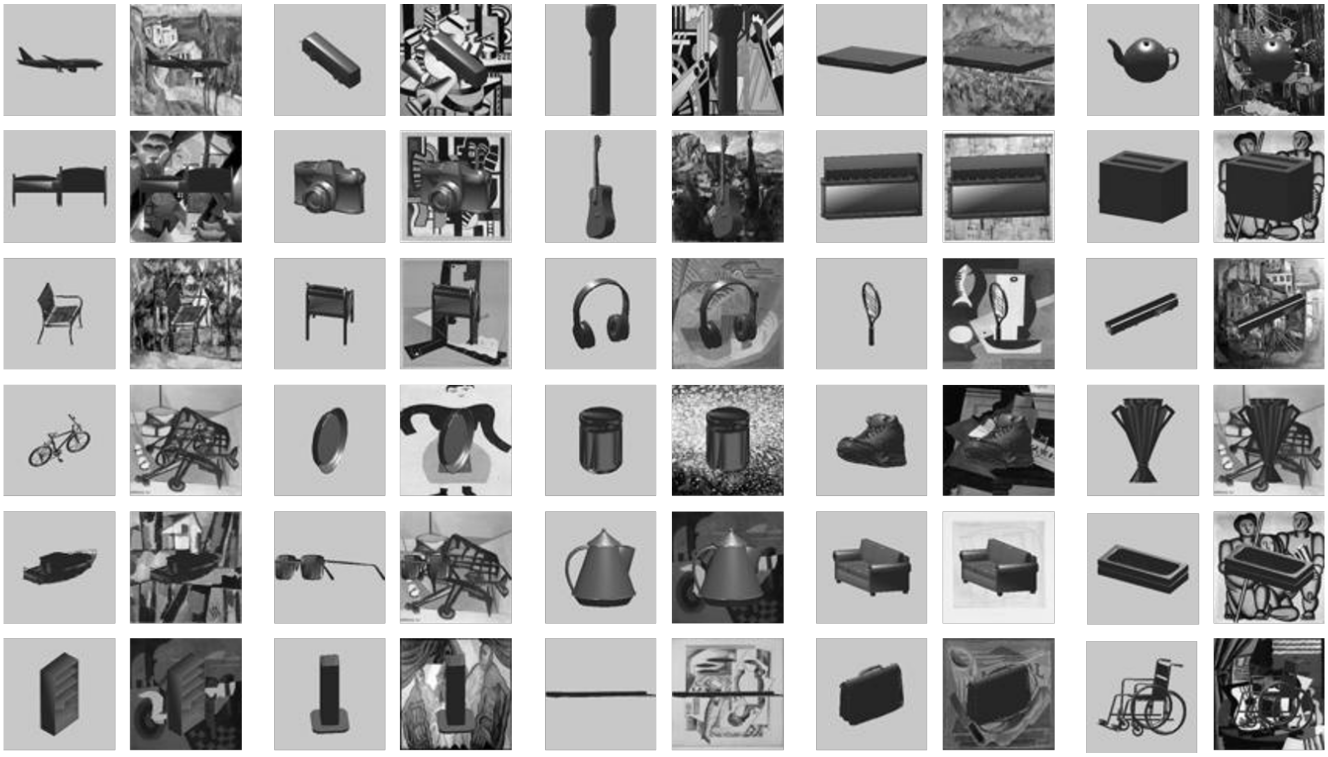

Figure 4. Unaugmented and augmented renderings of the subset of 30 classes from ObjectNet3D [4] employed to gauge the effect of task difficulty on implicit filter sparsity. The rendered images are 64x64 and obtained by randomly sampling (uniformly) the azimuth angle between -180 and 180 degrees, and the elevation between -15 and +45 degrees. The renderings are identical between the augmented and the unaugmented set and only differ in the background. The background images are grayscale versions of the Cubism subset from PeopleArt [3] dataset.

On TinyImageNet, both VGG-16 and BasicNet follow similar schemes. Using a mini-batch size of 40, the gradient descent method specific base learning rate is used for 250 epochs, and scaled down by 10 for an additional 75 epochs and further scaled down by 10 for an additional 75 epochs, totaling 400 epochs. When the mini-batch size is adjusted, the number of epochs are appropriately adjusted to ensure the same number of iterations.

On ImageNet, the base learning rate for Adam is 1e-4. For *BasicNet*, with a mini-batch size of 64, the base learning rate is used for 15 epochs, scaled down by a factor of 10 for another 15 epochs, and further scaled down by a factor of 10 for 10 additional epochs, totaling 40 epochs. The epochs are adjusted with a changing mini-batch size. For VGG-11, with a mini-batch size of 60, the total epochs are 60, with learning rate transitions at epoch 30 and epoch 50. For VGG-16, mini-batch size of 40, the total number of epochs are 50, with learning rate transitions at epoch 20 and 40.

Table 4. Convolutional filter sparsity in *BasicNet* trained on CIFAR10/100 for Adamax, AMSGrad and RMSProp with L2 regularization. Shown are the % of non-useful / inactive convolution filters, as measured by activation over training corpus (max act. $< 10^{-12}$) and by the learned BatchNorm scale ($|\gamma| < 10^{-03}$), averaged over 3 runs. See Table 1 in main paper for other combinations of regularization and gradient descent methods.

		CIFAR10			CIFAR100		
		% Sparsity		Test	% Sparsity		Test
	L2	by Act	by γ	Error	by Act	by γ	Error
AMSGrad	1e-02	93	93	20.9	95	95	71.9
	1e-04	51	47	9.9	20	13	35.6
	1e-06	0	0	11.2	0	0	40.2
Adamax	1e-02	75	90	16.4	74	87	51.8
	1e-04	49	50	10.1	10	10	39.3
	1e-06	4	4	11.3	0	0	39.8
RMSProp	1e-02	95	95	26.9	97	97	78.6
	1e-04	72	72	10.4	48	48	36.3
	1e-06	29	29	10.9	0	0	40.6

References

[1] CS231n convolutional neural networks for visual recognition. `http://cs231n.github.io/neural-networks-1/`.

[2] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *ICLR*, 2018.

[3] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. In *Advances in Neural Information Processing Systems*, pages 2074–2082, 2016.

[4] Yu Xiang, Wonhui Kim, Wei Chen, Jingwei Ji, Christopher Choy, Hao Su, Roozbeh Mottaghi, Leonidas Guibas, and Silvio Savarese. Objectnet3d: A large scale database for 3d object recognition. In *European Conference Computer Vision (ECCV)*. 2016.