Weighted Basis Images : Learning Alternative Data Representations

Harsh Menon*
menon.harsh@gmail.com

Abstract

Recent work on tiny machine learning (TinyML) has shown promising results on deploying machine learning models on extremely resource constrained devices such as microcontrollers. Existing approaches for model compression focus on quantization and making smaller models. In this paper, we present a novel approach for model compression that leverages data to generate efficient models. Specifically, we use a pre-trained network to generate a weighted combination of basis images on which it can achieve accuracy comparable to using the original images. The weights in the combination are generated by a student network and contain class-specific information that can be leveraged for image classification in lieu of the original images by a relatively small extension network. On CIFAR-10, our approach results in a reduction of 4x in parameters, 4x in FLOPS and an increase in 0.3% accuracy relative to a ResNet-110 teacher model. On the speech commands dataset, our approach generates a reduction of 8x in parameters, 8x in FLOPS with an increase in accuracy of 2% relative to a ResNet-56 teacher model. On the Visual Wake Words dataset, with image sizes ranging from 96-160 pixels, our approach is able to generate accuracy within 0.5% of a MobilenetV2-0.35 teacher network with a 10x reduction in parameters and 1.45x reduction in FLOPS.

1 Introduction

Model optimization is a fundamental problem in machine learning where a given model is optimized to meet some criteria such as fitting within the budgets of a resource constrained device. Typical examples include visual wake word detection which is a binary image classification problem where the device must wake if it detects a person in its field of view. Current methods for model optimization seek a Pareto-optimal front trading accuracy for FLOPS [4], representational efficiency [10] or some other metric. A variety of approaches have been proposed to solve this problem ranging from pruning [6], quantization [5], mixed precision [9], knowledge distillation [3] and neural architecture search [2].

In this paper, we propose a novel approach to model optimization that is based on knowledge distillation [3] and focus on the task of image classification. We start with a pre-trained teacher model, but rather than train a student model to learn to classify like the teacher, we train a set of images and a student network that predicts weights for these images. We refer to these images as *basis images* as we map each original image to a linear combination of the weighted basis images. Therefore these images can be loosely thought of as representing a basis that we project the original images onto.

Typically during training the network weights are updated while the input data remains constant. Our approach can be considered as addressing the inverse of that process where we fix the weights of

*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

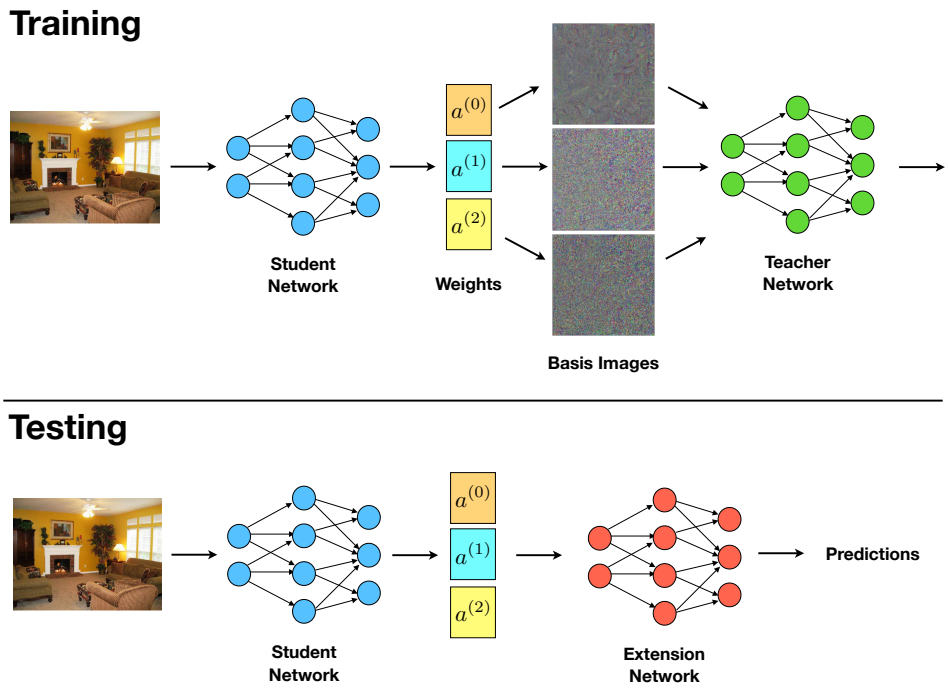


Figure 1: Overview of proposed approach

the teacher network and allow it to choose an alternative representation of the data by learning the masked basis images. The motivation here being that if the network were allowed to choose its own representation of the data it would come up with a much more efficient representation than the original.

Our contributions in this paper are

- We present a novel approach to model compression that leverages data for model compression and is able to produce efficient models
- Our approach produces a privacy-preserving representation of the data that can be used for privacy-preserving training/inference tasks

2 Related Work

Our work touches on several areas of research ranging from model compression to dataset distillation. In this section, we highlight some approaches that are most relevant to our work.

2.0.1 Model Compression

Model compression approaches range from quantization [5] to knowledge distillation [3]. In knowledge distillation, a teacher model is used to train a student model to achieve comparable performance on a given task. Our approach also uses a teacher model to train a student model. However, rather than train both teacher and student to classify using a specialized loss, we leverage an alternative data representation to transfer knowledge from the teacher to the student.

2.0.2 Dataset Distillation

Dataset distillation [11, 12] approaches aim to condense the knowledge from a large dataset into a smaller one so that models trained on the smaller dataset can achieve accuracy comparable to the original dataset. Our approach differs from these approaches in that rather than distill the dataset into a smaller set of unique samples, we distill the dataset into a common set of basis images and represent each sample as a linear combination of these basis images.

3 Weighted Basis Images

3.1 Approach

Our approach is to use a fixed pre-trained teacher network to learn a linear combination of weighted basis images on which the teacher's accuracy is comparable to the original images. The weights are generated by a student network which takes in an original image and outputs the weights for that given image. Once we have trained the student network, we use its weights to train a smaller extension network that learns to classify using the weights rather than the original images. Finally, during inference we use the student and extension model to produce class predictions.

3.2 Problem Formulation

Consider a dataset $\mathcal{U}$ with samples $\left\{\left(x^{(i)}, y^{(i)}\right), i = 1, \ldots, N\right\}$, where $x^{(i)} \in \mathbb{R}^D, y^{(i)} \in \mathbb{R}$. Assume we have a model T that is parameterized by weights W that takes in a data sample $x^{(i)}$ and returns an output $T(W, x^{(i)}) \in \mathbb{R}^K$. We train f by minimizing the loss $\ell(T(W, x^{(i)}), y^{(i)})$ where the outputs are compared to the true labels $y^{(i)}$ using a loss criterion. For image classification, we use cross entropy as our loss criterion. After training, we obtain an optimal set of weights $\overline{W}$. We refer to T as the teacher model.

Our goal is to approximate the input samples $x^{(i)} \in \mathcal{U}$, with a weighted linear combination of $M (< D < N)$ samples $z^{(i)} \in \mathbb{R}^D, i = 1, \ldots, M$ that minimizes the loss of T with weights $\overline{W}$ but operates on these samples instead of the original samples. More specifically, we solve the following problem

$$\min_{B_M, W_S} \quad \frac{1}{N} \sum_{i=1}^{N} \ell(f(\overline{W}, B_M^{(i)}), y^{(i)}) \tag{1}$$

$$\text{where} \quad B_M = AZ \tag{2}$$

$$A = S(X) \tag{3}$$

Here S is the student model (parameterized by weights W_S) that takes in the original samples ($X \in \mathbb{R}^{N \times D}$) and returns the coefficients $A \in \mathbb{R}^{N \times M}$. We refer to each of the $z^{(i)}$ as basis images and $B_M \in \mathbb{R}^{N \times D}$ is the weighted sum of the M basis images for each of the input samples. $Z \in \mathbb{R}^{M \times D}$ is a matrix whose i-th row is $z^{(i)}$. We denote the (i, j) entry of A as $a^{(ij)}$ and constrain the entries to satisfy $|a^{(ij)}| \leq 1$.

Having obtained A and Z, we can now transform the input samples $x^{(i)}$ into an M dimensional space where each input sample is represented as an M dimensional vector $a^{(i)}$. It should be noted that in general,

$$x^{(i)} \neq B_M^{(i)} = \sum_{i=1}^{M} a^{(ij)} z^{(i)} \tag{4}$$

so we have generated an alternative representation of the dataset that for the given weights $\overline{W}$. With this lower-dimensional representation of the data, we now train a smaller model E (the "extension") that is parameterized by weights W_E and takes in an M dimensional coefficient vector $a^{(i)} \in \mathbb{R}^M$ and returns an output $E(W_E, a^{(i)})$ that minimizes

$$\min_{W_E} \quad \frac{1}{N} \sum_{i=1}^{N} \ell(E(W_E, a^{(i)}), y^{(i)}) \tag{5}$$

We now use the student model and its extension to do inference instead of the original teacher model. We refer to the combination of the student model and its extension as *basisnet*.

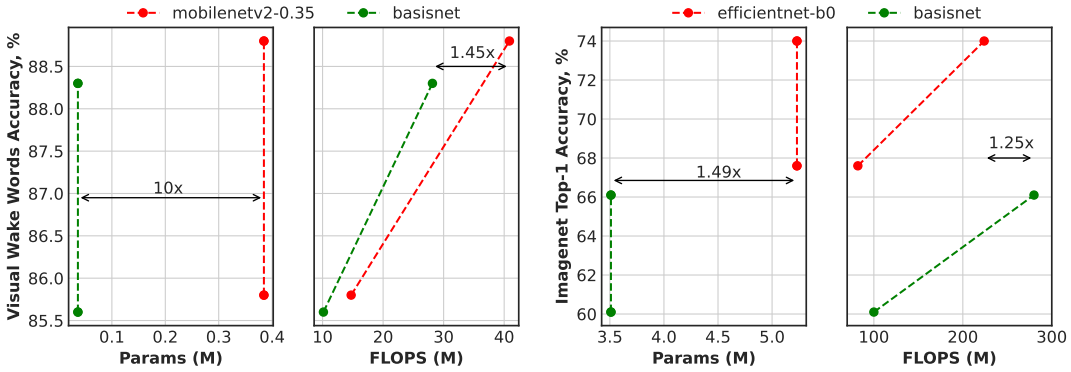


Figure 2: Accuracy of BasisNet on VWW and Imagenet datasets

3.3 Similarity Loss

In order to encourage the weights of similar classes to lie closer together, we introduce a similarity loss which is defined below.

$$L_{sim}(x) = |\max_{x' \in C} d(x, x') - \min_{x' \notin C} d(x, x') - \epsilon| \tag{6}$$

We compute this for every batch where $d(x, x')$ is the Euclidean distance between the weights, $\epsilon > 0$ is a learnable margin threshold and C is the class that x belongs to. Previous work [10] has defined a similar loss (triplet loss) but the difference here is that we don't use any anchor points and the margin is learned. This loss is then combined with the cross-entropy loss as

$$L = L_{cross-entropy} + \lambda L_{sim} \tag{7}$$

where λ is determined empirically.

3.4 Extending the weights

Finally, rather than just multiply each basis image with a single weight, we allow the each basis image to have multiple weights with each weight acting on a separate part of the image. We split the image into a $P \times Q$ grid and assign a separate weight for each sub-grid.

4 Datasets

We evaluate our approach on two image classification datasets - Visual Wake Words [1] and ImageNet [8]. Visual Wake Words is a person detection dataset based off the COCO dataset [7] where the labels have been modified to support binary classification. This dataset is motivated by microcontroller use-cases where the device needs to be woken when it detects a person. ImageNet is a large scale benchmark for image classification. We use the Visual Wake Words dataset to understand how our approach works in the binary classification task and then use ImageNet to test how well it scales to larger datasets and multiple labels.

5 Experiments

5.1 Results

Figure 2 shows how basisnet performs on the Visual Wake Words and Imagenet dataset. Our approach is able to significantly reduce the number of parameters and FLOPS on VWW. On Imagenet, we are able to get within 8% of the teacher model with a slight increase in FLOPS and decrease in parameters. It should be noted that for these results, we determined the student network architecture by starting with the teacher architecture and removing layers or modifying other hyperparameters manually. We would expect replacing this manual search with neural architecture search to produce

Table 1: BasisNet-VWW

Operator	Input	Channels	Layers
Conv2d	$160 \times 160 \times 3$	16	1
Bottleneck	$80 \times 80 \times 16$	8	1
Bottleneck	$40 \times 40 \times 8$	8	2
Bottleneck	$20 \times 20 \times 8$	16	3
Bottleneck	$10 \times 10 \times 16$	24	3
AvgPool	$10 \times 10 \times 24$		1
Conv2d	$1 \times 1 \times 24$	M	1
FC	$M \times 1$	20	1
FC	20×1	20	1
FC	20×1	K	1

Table 2: BasisNet-Imagenet

Operator	Input	Channels	Layers
Conv2d	$160 \times 160 \times 3$	32	1
MBConv1	$80 \times 80 \times 32$	16	1
MBConv6	$80 \times 80 \times 16$	40	2
MBConv6	$40 \times 40 \times 40$	80	2
MBConv6	$20 \times 20 \times 80$	112	3
MBConv6	$10 \times 10 \times 112$	192	5
MBConv6	$5 \times 5 \times 192$	320	1
MBConv6	$5 \times 5 \times 320$	128	1
AvgPool	$5 \times 5 \times 128$		1
Conv2d	$1 \times 1 \times 128$	M	1
FC	$M \times 1$	800	1
FC	800×1	800	1
FC	800×1	K	1

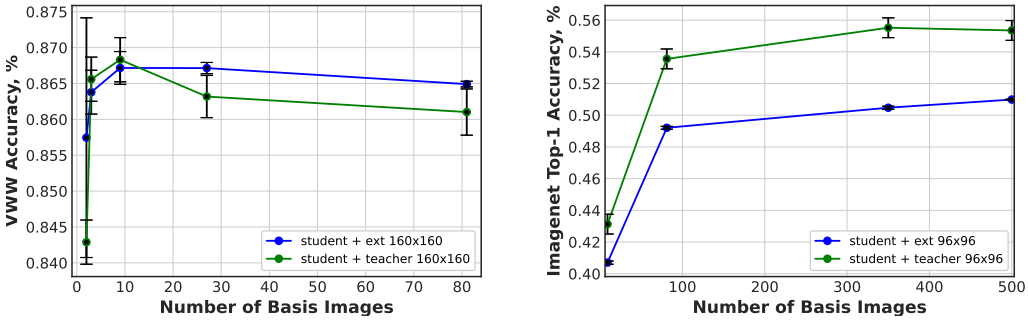


Figure 3: Variation of accuracy with number of basis images

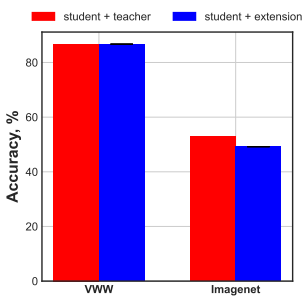
superior results. For the Visual Wake Words dataset, our architecture closely follows the teacher network. Our network consists of the first 10 layers of MobileNetV2-0.35 followed by a 3 layer MLP whose details are shown in Table 1. For the Imagenet dataset, our architecture follows the efficientnet-b0 network with a few modifications, followed by a 3 layer MLP. The details are shown in Table 2.

5.1.1 Effects of increasing the number of basis images

Figure 3 shows the variation of the accuracy of the student and extension network as well as the student and teacher network with increasing number of basis images for different images sizes on the two datasets. For VWW, we see that increasing the number of basis images increases the overall accuracy upto a certain threshold after which there is a shallow dropoff. Furthermore, this trend holds for both the student and teacher combination and the student and extension combination. On Imagenet, we see a similar trend for both the student and teacher combination and student and extension model.

5.1.2 Accuracy of the student and extension networks

Figure 4 shows how the accuracy drops for a fixed student and teacher combination on the VWW and ImageNet datasets for one of the data points shown in 3. We see that for VWW the student and extension retains most of the accuracy of the original network with only a 0.1% decrease in accuracy. On ImageNet, we see a greater reduction in accuracy of about 5%.



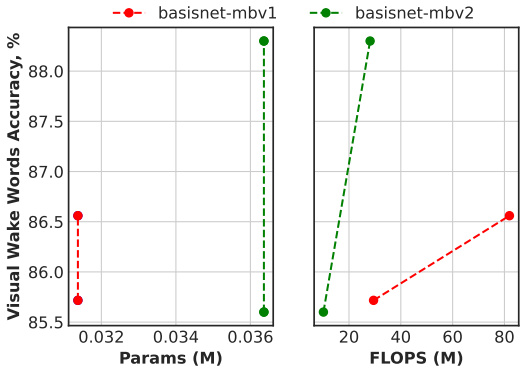


Figure 5: Comparison of basisnet with mobilenetv1 and mobilenetv2 student architecture

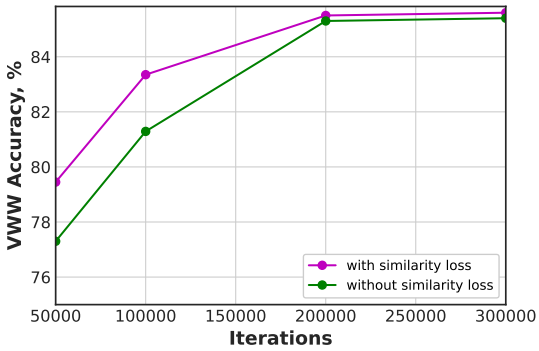


Figure 6: Variation of VWW validation accuracy during training with and without similarity loss

5.1.3 Effects of not learning the basis images

5.1.4 Effects of architecture variation in the student network

Figure 5 shows the performance of basisnet if we replace the student architecture shown in 1 with a MobilenetV1 based architecture shown in 3. Here we see that with this new architecture we are able to achieve a significant reduction in parameters as well with comparable accuracy as the MobilenetV2 version. However, this comes at a increase in FLOPS. This highlights the importance of the architecture of the student network in determining an optimal tradeoff between the different parameters.

5.1.5 Effects of the similarity loss

Figure 6 shows how the accuracy of the student and teacher network varies as we train on 96x96 images of the VWW dataset. We see that the similarity loss provides faster convergence and a slight increase in accuracy.

5.1.6 Effects of adding more weights per basis image

Figure 7 shows the results of training 3 models on Imagenet. The first has 32 basis images, the second 128 basis images and the third has 32 basis images but each basis image has 4 weights (with each weight being applied to a different quadrant of the image) resulting in a total of 128 weights. From the figure, we see that the accuracy of the $32 \times 2 \times 2$ model is comparable to that of using just a single weight but 128 basis images, implying that the total number of basis images is what determines the final accuracy of the model rather than the number of weights per basis image.

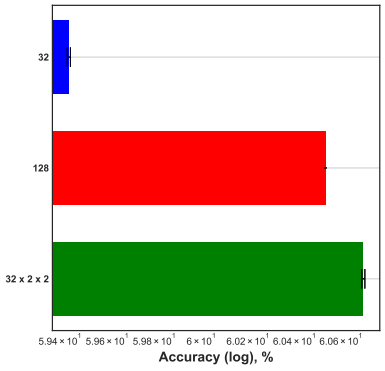


Figure 7: Variation of VWW accuracy with multiple weights per basis image

5.1.7 Visualizing basis images and their reconstructions

In this section, we visualize the basis images as well the reconstructions of the original images. Figure 8 shows the original images on Imagenet and their reconstructions using basis images. From these images, we see that the reconstructions look nothing like the original images and appear randomized.

Table 3: BasisNet-VWW (MobilenetV1)

Operator	Input	Channels
Conv2d	$160 \times 160 \times 3$	32
DW + PW Conv	$80 \times 80 \times 32$	64
DW + PW Conv	$80 \times 80 \times 64$	128
DW + PW Conv	$40 \times 40 \times 128$	8
AvgPool	$40 \times 40 \times 128$	
Conv2d	$1 \times 1 \times 128$	M
FC	$M \times 1$	20
FC	20×1	20
FC	20×1	K

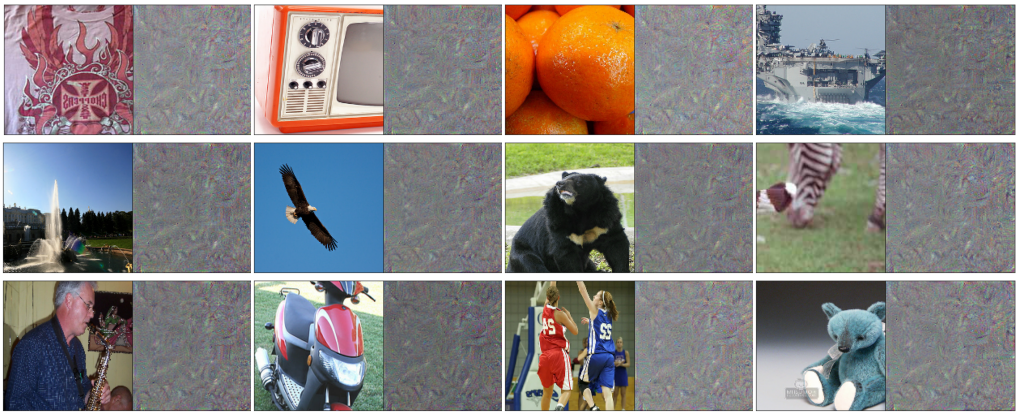


Figure 8: Sample images from Imagenet (left) and their counterparts (right) using basis images

6 Conclusions and Future Work

In this work, we introduced a novel approach that learns an alternative data representation for model compression and produces a privacy-preserving representation of the data. Our approach produces efficient models and expands the Pareto curve offering more parameters one can tweak to obtain optimal performance on a range of criteria.

As we discovered through our experiments, the architecture of the student network is crucial to performance and so one future research direction would be adding neural architecture search and see how it improves the performance of our approach. Apart from wake word detection, another common use case in tinyML is continuously running low-power monitoring devices. In the future, we could look into how to extend our approach to address the constraints of such devices. Finally, while we have demonstrated our approach only on images, we could try to extend it to other modalities such as audio or language and see how our approach generalizes to those domains.

References

[1] CHOWDHERY, A., WARDEN, P., SHLENS, J., HOWARD, A., AND RHODES, R. Visual wake words dataset. *CoRR abs/1906.05721* (2019).

[2] ELSKEN, T., METZEN, J. H., AND HUTTER, F. Neural architecture search: A survey. *J. Mach. Learn. Res. 20* (2019), 55:1–55:21.

[3] HINTON, G. E., VINYALS, O., AND DEAN, J. Distilling the knowledge in a neural network. *CoRR abs/1503.02531* (2015).

[4] HOWARD, A. G., ZHU, M., CHEN, B., KALENICHENKO, D., WANG, W., WEYAND, T., ANDREETTO, M., AND ADAM, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR abs/1704.04861* (2017).

[5] HUBARA, I., COURBARIAUX, M., SOUDRY, D., EL-YANIV, R., AND BENGIO, Y. Quantized neural networks: Training neural networks with low precision weights and activations. *J. Mach. Learn. Res. 18* (2017), 187:1–187:30.

[6] LECUN, Y., DENKER, J. S., AND SOLLA, S. A. Optimal brain damage. In *Advances in Neural Information Processing Systems 2, [NIPS Conference, Denver, Colorado, USA, November 27-30, 1989]* (1989), D. S. Touretzky, Ed., Morgan Kaufmann, pp. 598–605.

[7] LIN, T., MAIRE, M., BELONGIE, S. J., HAYS, J., PERONA, P., RAMANAN, D., DOLLÁR, P., AND ZITNICK, C. L. Microsoft COCO: common objects in context. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V* (2014), D. J. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., vol. 8693 of *Lecture Notes in Computer Science*, Springer, pp. 740–755.

[8] RUSSAKOVSKY, O., DENG, J., SU, H., KRAUSE, J., SATHEESH, S., MA, S., HUANG, Z., KARPATHY, A., KHOSLA, A., BERNSTEIN, M. S., BERG, A. C., AND LI, F. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis. 115*, 3 (2015), 211–252.

[9] SA, C. D., LESZCZYNSKI, M., ZHANG, J., MARZOEV, A., ABERGER, C. R., OLUKOTUN, K., AND RÉ, C. High-accuracy low-precision training. *CoRR abs/1803.03383* (2018).

[10] SCHROFF, F., KALENICHENKO, D., AND PHILBIN, J. Facenet: A unified embedding for face recognition and clustering. 815–823.

[11] WANG, T., ZHU, J., TORRALBA, A., AND EFROS, A. A. Dataset distillation. *CoRR abs/1811.10959* (2018).

[12] ZHAO, B., MOPURI, K. R., AND BILEN, H. Dataset condensation with gradient matching. *CoRR abs/2006.05929* (2020).

A Appendix A

To simplify notation, we will refer to $B_M^{(i)}$ as $\theta_M^{(i)}$ and $l(f(\overline{W}, \theta_M^{(i)}), y^{(i)})$ in the following as $l(\theta_M^{(i)})$. In the following, we present a few propositions that will provide the basis for our approach.

Definition A.1. Define the M term approximation error of the i-th sample $x^{(i)}$ as

$$\varepsilon_M^{(i)} = x^{(i)} - \theta_M^{(i)} \tag{8}$$

where $\theta_M^{(i)}$ is obtained by solving the problem defined in (1).

Proposition A.1. *For a continuous twice-differentiable loss function ℓ, the difference in loss between the M and $M-1$ term approximation is bounded by*

$$\ell(\theta_M^{(i)}) - \ell(\theta_{M-1}^{(i)}) \leq ||z^{(M)}|| \left(||g(\theta_M^{(i)})|| + \frac{1}{2}\lambda_H(\theta_{M-1}^{(i)}) \, ||z^{(M)}|| \right) \tag{9}$$

and we have

$$g(\theta_{M-1}^{(i)}) = g(\theta_M^{(i)}) \tag{10}$$

$$H(\theta_{M-1}^{(i)}) \succeq 0 \tag{11}$$

$$H(\theta_M^{(i)}) \succeq 0 \tag{12}$$

where $g(\theta_M^{(i)})$ is the gradient of the loss at $\theta_M^{(i)}$ and $H(\theta_M^{(i)})$ is the hessian of the loss at $\theta_M^{(i)}$ and $\lambda_H(\theta_M^{(i)})$ is the largest eigenvalue of the $H(\theta_M^{(i)})$.

The proof is shown in Appendix A.

Proposition A.2. *For a continuous twice-differentiable loss function ℓ, the difference in loss between the $x^{(i)}$ and the M term approximation is bounded by*

$$\ell(x^{(i)}) - \ell(\theta_M^{(i)}) \leq (D - M) \left(\max_{0 \leq k \leq D} ||z^{(k)}|| \left(||g(x^{(i)})|| + \frac{1}{2} \max_{0 \leq k \leq D} \left(\lambda_H(\theta_k^{(i)}) ||z^{(k)}|| \right) \right) \right) \tag{13}$$

and we have that

$$g(\theta_j^{(i)}) = g(x^{(i)}), \forall j \tag{14}$$

A.1 Proof of Proposition 4.1

Proof. Define $\gamma_M^{(i)}$ as a sub-optimal approximation of $\theta_M^{(i)}$ that we construct as

$$\gamma_M^{(i)} = \theta_{M-1}^{(i)} + \alpha^{(iM)} z^{(M)} \tag{15}$$

since we know that

$$\ell(\theta_M^{(i)}) \leq \ell(\gamma_M^{(i)}) \tag{16}$$

We will bound $\ell(\gamma_M^{(i)})$ and then apply those bounds to $\ell(\theta_M)$. Now we do a Taylor series expansion of the loss function around $\theta_{M-1}^{(i)}$

$$\begin{aligned}
\ell(\theta_{M-1}^{(i)} + \alpha^{(iM)} z^{(M)}) &= \ell(\theta_{M-1}^{(i)}) + \alpha^{(iM)} z^{(M)} g(\theta_{M-1}^{(i)})^T + \frac{1}{2} \alpha^{(iM)} z^{(M)} H(\theta_{M-1}^{(i)}) (\alpha^{(iM)} z^{(M)})^T \\
\implies \ell(\gamma_M^{(i)}) &= \ell(\theta_{M-1}^{(i)}) + \alpha^{(iM)} z^{(M)} g(\theta_{M-1}^{(i)})^T + \frac{1}{2} \alpha^{(iM)} z^{(M)} H(\theta_{M-1}^{(i)}) (\alpha^{(iM)} z^{(M)})^T \\
\implies \ell(\theta_M^{(i)}) &\leq \ell(\theta_{M-1}^{(i)}) + \alpha^{(iM)} z^{(M)} g(\theta_{M-1}^{(i)})^T + \frac{1}{2} \alpha^{(iM)} z^{(M)} H(\theta_{M-1}^{(i)}) (\alpha^{(iM)} z^{(M)})^T
\end{aligned} \tag{17}$$

Define $\xi_{M-1}^{(i)}$ as a sub-optimal approximation of $\theta_{M-1}^{(i)}$ that we construct as

$$\xi_{M-1}^{(i)} = \theta_M^{(i)} - \alpha^{(iM)} z^{(M)} \tag{18}$$

since we know that

$$\ell(\theta_{M-1}^{(i)}) \leq \ell(\xi_{M-1}^{(i)}) \tag{19}$$

Now we do a Taylor series expansion of the loss function around $\theta_M^{(i)}$

$$\begin{aligned}
\ell(\theta_M^{(i)} - \alpha^{(iM)} z^{(M)}) &= \ell(\theta_M^{(i)}) - \alpha^{(iM)} z^{(M)} g(\theta_{M-1}^{(i)})^T + \frac{1}{2} \alpha^{(iM)} z^{(M)} H(\theta_M^{(i)}) (\alpha^{(iM)} z^{(M)})^T \\
\implies \ell(\xi_{M-1}^{(i)}) &= \ell(\theta_M^{(i)}) - \alpha^{(iM)} z^{(M)} g(\theta_M^{(i)})^T + \frac{1}{2} \alpha^{(iM)} z^{(M)} H(\theta_M^{(i)}) (\alpha^{(iM)} z^{(M)})^T \\
\implies \ell(\theta_{M-1}^{(i)}) &\leq \ell(\theta_M^{(i)}) - \alpha^{(iM)} z^{(M)} g(\theta_M^{(i)})^T + \frac{1}{2} \alpha^{(iM)} z^{(M)} H(\theta_M^{(i)}) (\alpha^{(iM)} z^{(M)})^T
\end{aligned} \tag{20}$$

Adding (17) and (20), we get

$$0 \leq \alpha^{(iM)} z^{(M)} \left(g(\theta_{M-1}^{(i)}) - g(\theta_M^{(i)}) \right)^T + \frac{1}{2} (\alpha^{(iM)})^2 z^{(M)} \left(H(\theta_{M-1}^{(i)}) + H(\theta_M^{(i)}) \right) (z^{(M)})^T \tag{21}$$

The only way that the above equation can hold for any value of $\alpha^{(iM)}, z^{(M)}$ is if

$$g(\theta_{M-1}^{(i)}) = g(\theta_M^{(i)}) \tag{22}$$

$$H(\theta_{M-1}^{(i)}) + H(\theta_M^{(i)}) \succeq 0 \tag{23}$$

Since the hessian is a symmetric matrix and every symmetric matrix can be represented as the sum of a positive semidefinite and negative semidefinite matrix, the above condition implies that

$$H(\theta_{M-1}^{(i)}) \succeq 0 \tag{24}$$

$$H(\theta_{M}^{(i)}) \succeq 0 \tag{25}$$

Thus, we can describe the difference in loss between the two approximations as

$$\ell(\theta_M^{(i)}) - \ell(\theta_{M-1}^{(i)}) \leq \alpha^{(iM)} z^{(M)} g(\theta_M^{(i)})^T + \frac{1}{2}\alpha^{(iM)} z^{(M)} H(\theta_{M-1}^{(i)})(\alpha^{(iM)} z^{(M)})^T \tag{26}$$

$$\leq \left|\alpha^{(iM)} z^{(M)} g(\theta_M^{(i)})^T\right| + \frac{1}{2}(\alpha^{(iM)})^2 z^{(M)} H(\theta_{M-1}^{(i)})(z^{(M)})^T \tag{27}$$

$$\leq \left|z^{(M)} g(\theta_M^{(i)})^T\right| + \frac{1}{2} z^{(M)} H(\theta_{M-1}^{(i)})(z^{(M)})^T \tag{28}$$

$$\leq ||z^{(M)}||\ ||g(\theta_M^{(i)})|| + \frac{1}{2} z^{(M)} H(\theta_{M-1}^{(i)})(z^{(M)})^T \tag{29}$$

$$\tag{30}$$

We can simplify the second term on the RHS as

$$z^{(M)} H(\theta_{M-1}^{(i)})(z^{(M)})^T = |z^{(M)} H(\theta_{M-1}^{(i)})(z^{(M)})^T| \tag{31}$$

$$\leq ||z^{(M)}||\ ||H(\theta_{M-1}^{(i)})\ (z^{(M)})^T|| \tag{32}$$

$$\leq ||z^{(M)}||\ ||H(\theta_{M-1}^{(i)})||\ ||z^{(M)}|| \tag{33}$$

$$= (||H(\theta_{M-1}^{(i)})||)(||z^{(M)}||^2) \tag{34}$$

$$= \lambda_H(\theta_{M-1}^{(i)})||z^{(M)}||^2 \tag{35}$$

where $\lambda_H(\theta_{M-1}^{(i)})$ is the largest eigenvalue of the Hessian H and we have used the fact that the 2-norm of the hessian is equal to the largest eigenvalue since H is symmetric. Putting this all together, we get

$$\ell(\theta_M^{(i)}) - \ell(\theta_{M-1}^{(i)}) \leq ||z^{(M)}||\ \left(||g(\theta_M^{(i)})|| + \frac{1}{2}\lambda_H(\theta_{M-1}^{(i)})\ ||z^{(M)}||\right) \tag{36}$$

□

A.2 Proof of Proposition 4.2

Proof. First, note that when $M = D$, then $\theta_D^{(i)} = x^{(i)}$ because $z^{(j)} = e^j, \alpha^{(ij)} = x^{(ij)}$. So, we can use the result from A.1 to get

$$\ell(x^{(i)}) - \ell(\theta_{D-1}^{(i)}) \leq \ \left(||g(\theta_M^{(i)})|| + \frac{1}{2}\lambda_H(\theta_{D-1}^{(i)})\right) \tag{37}$$

Now we can do this for $\theta_{D-1}^{(i)}, \ldots, \theta_{M+1}^{(i)}$ to get

$$\ell(\theta_{D-1}^{(i)}) - \ell(\theta_{D-2}^{(i)}) \leq ||z^{(D-1)}||\ \left(||g(\theta_{D-2}^{(i)})|| + \frac{1}{2}\lambda_H(\theta_{D-2}^{(i)})\ ||z^{(D-1)}||\right) \tag{38}$$

$$\ldots \tag{39}$$

$$\ell(\theta_{M+1}^{(i)}) - \ell(\theta_{M}^{(i)}) \leq ||z^{(M+1)}||\ \left(||g(\theta_M^{(i)})|| + \frac{1}{2}\lambda_H(\theta_{M}^{(i)})\ ||z^{(M+1)}||\right) \tag{40}$$

Adding all of these together, we get

$$\ell(x^{(i)}) - \ell(\theta_M^{(i)}) \leq \sum_{k=M+1}^{D} \left(||z^{(k)}||\ \left(||g(\theta_{k-1}^{(i)})|| + \frac{1}{2}\lambda_H(\theta_{k-1}^{(i)})\ ||z^{(k)}||\right)\right) \tag{41}$$

Since we also have that $g(\theta_j^{(i)}) = g(\theta_{j-1}^{(i)}), \forall j$, we can state that

$$g(\theta_M^{(i)}) = g(\theta_{M+1}^{(i)}) = \cdots = g(x^{(i)}) \tag{42}$$

This gives us

$$\ell(x^{(i)}) - \ell(\theta_M^{(i)}) \leq \left(\sum_{k=M+1}^{D} ||z^{(k)}|| \right) ||g(x^{(i)})|| + \left(\sum_{k=M+1}^{D} \frac{1}{2} \lambda_H(\theta_{k-1}^{(i)}) \, ||z^{(k)}||^2 \right) \tag{43}$$

$$\leq (D - M) \left(\max_{0 \leq k \leq D} ||z^{(k)}|| \left(||g(x^{(i)})|| + \frac{1}{2} \max_{0 \leq k \leq D} \left(\lambda_H(\theta_k^{(i)}) ||z^{(k)}|| \right) \right) \right) \tag{44}$$

□

B Appendix B

B.1 Linear Regression

Consider a dataset with samples $\left\{ \left(x^{(i)}, y^{(i)} \right), i = 1, \ldots, N \right\}$, where $x^{(i)} \in \mathbb{R}^{D \times 1}, y^{(i)} \in \mathbb{R}$. We can express the linear regression problem as

$$\min_{\theta} \frac{1}{2} \|X\theta - y\|_2^2 \tag{45}$$

where X is a $N \times D$ matrix whose rows are $(x^{(i)})^T$, y is a N dimensional vector and θ is a D dimensional vector of weights. The solution to the above problem can be obtained by solving the normal equations shown below

$$X^T X\theta = X^T y \tag{46}$$

In our approach, we approximate the dataset with a linear combination of $M < N$ synthetic data samples

$$x^{(i)} = \sum_{j=1}^{M} \alpha^{(ij)} z^{(j)} + e^{(i)} \tag{47}$$

where the coefficients are defined as

$$\alpha^{(ij)} = (x^{(i)})^T z^{(j)} \tag{48}$$

We can express this in matrix form as

$$X = AZ + E \tag{49}$$

$$A = XZ^T \tag{50}$$

where A is a $N \times M$ matrix containing the $\alpha^{(ij)}$ coefficients, Z is a $M \times D$ matrix whose rows are the synthetic data samples $(z^{(j)})^T$ and E is a $N \times D$ error matrix whose rows are the error vectors $(e^{(i)})^T$. Putting this all together, our goal is given an optimal weight θ and the original dataset, find an optimal Z that minimizes the loss shown below.

$$\min_{Z} \quad \frac{1}{2} \|XZ^T Z\theta - y\|_2^2 \tag{51}$$

Since this is a non-convex problem, we can instead solve a semidefinite program with a change of variables $W = Z^T Z$

$$\begin{aligned} \min_{W} \quad & \frac{1}{2} \|XW\theta - y\|_2^2 \\ \text{s.t.} \quad & W \succeq 0 \\ & W = W^T \end{aligned} \tag{52}$$

where since W is symmetric positive semi-definite, we can construct Z by using the first M eigenvectors from the eigenvalue decomposition of W as shown below.

$$W = Q\Lambda Q^T = (Q\Lambda^{1/2})(Q\Lambda^{1/2})^T \tag{53}$$

$$Z^T = Q_M \Lambda_M^{1/2} \tag{54}$$

where Q is the $D \times D$ matrix of eigenvectors and Q_M is the $D \times M$ matrix containing the first M eigenvalues of Q and Λ_M is a diagonal $M \times M$ matrix containing the first M eigenvalues.

Taking the derivative w.r.t W we get

$$\nabla_W L(W) = \nabla_W \frac{1}{2}(XW\theta - y)^T(XW\theta - y) \tag{55}$$

$$= \nabla_W \frac{1}{2}\left((XW\theta)^T XW\theta - (XW\theta)^T y - y^T(XW\theta) + y^T y\right) \tag{56}$$

$$= \nabla_W \frac{1}{2}\left(\theta^T W^T X^T XW\theta - 2y^T XW\theta\right) \tag{57}$$

$$= X^T XW\theta\theta^T - X^T y\theta^T \tag{58}$$

Setting this to zero, we obtain

$$X^T XW\theta\theta^T = X^T y\theta^T \tag{59}$$

Assuming $X^T X$ is invertible, we can simplify the above as

$$W\theta\theta^T = (X^T X)^{-1} X^T y\theta^T = \theta\theta^T \tag{60}$$

Post multiplying by θ, we get

$$(W\theta\theta^T)\theta = (\theta\theta^T)\theta \tag{61}$$

$$\implies ||\theta||^2 W\theta = ||\theta||^2\theta \tag{62}$$

$$\implies W\theta = \theta \tag{63}$$

which implies that the optimal W will have θ as its eigenvector corresponding to the eigenvalue 1. This makes sense because then the loss function can be expressed as

$$||XW\theta - y|| = ||X\theta - y|| \tag{64}$$

and thus the loss using the synthetic data matches that of using the original data and is independent of M in the linear case.