Efficient Inference on Deep Neural Networks by Dynamic Representations and Decision Gates

Mohammad Saeed Shafiee[*]
Faculty of Electrical Engineering
Shahid Rajee University
Tehran, Iran
`ms.shafiee@srttu.edu`

Mohammad Javad Shafiee, Alexander Wong
Vision & Image Processing Research Group
Waterloo AI Institute, University of Waterloo
DarwinAI Corp., Waterloo, Ontario, Canada
`{mjshafiee,a28wong}@uwaterloo.ca`

Abstract

The current trade-off between depth and computational cost makes it difficult to adopt deep neural networks for many industrial applications, especially when computing power is limited. Here, we are inspired by the idea that, while deeper embeddings are needed to discriminate difficult samples, a large number of samples can be well discriminated via much shallower embeddings. In this study, we introduce the concept of **decision gates** (d-gate), modules trained to decide whether a sample needs to be projected into a deeper embedding or if an early prediction can be made at the d-gate, thus enabling the computation of dynamic representations at different depths. The proposed d-gate modules can be integrated with any deep neural network and reduces the average computational cost of the deep neural networks while maintaining modeling accuracy. Experimental results show that leveraging the proposed d-gate modules led to a ∼38% speed-up and ∼39% FLOPS reduction on ResNet-101 and ∼46% speed-up and ∼36% FLOPS reduction on DenseNet-201 trained on the CIFAR10 dataset with only ∼2% drop in accuracy.

1 Introduction

Past studies such as [16] have shown that deeper architectures often lead to better modeling performance; however, deeper architectures also pose a number of issues. Besides becoming more prone to overfitting and becoming more difficult to train, the trade-off between depth and computational cost makes it difficult to adopt deeper architectures for many industrial applications.

He *et al.* [7] tackled the former issue of degradation in learning deeper neural networks (e.g., vanishing gradient) by introducing the concept of residual learning, where learning is based on the residual mapping rather than directly on the unreferenced mapping. Following that, Xie *et al.* [19] took advantage of the inception idea (i.e, split-transform-merge strategy) within a residual block structure to provide better subspace modeling while resolving the degradation problem at the same time, resulting in a ResNext architecture with improved modeling accuracy. To tackle the issue of computational cost, a wide variety of methods have been proposed, including: precision reduction [10], model compression [6], teacher-student strategies [8], and evolutionary algorithms [13, 14].

More recently, conditional computation [1, 4, 12, 18, 2] and early prediction [17] methods have been proposed to tackle this issue, which involve the dynamic execution of different modules within a network. Conditional computation methods have largely been motivated by the idea that residual networks can be considered as an ensemble of shallower networks. As such, these methods take advantage of skip connections to determine which residual modules are necessary to be executed, with most leveraging reinforcement learning.

In this study, we explore the idea of early prediction but instead draw inspiration from the soft-margin support vector [3] theory for decision-making. Specifically, we introduce the concept of decision gates (d-gate), modules trained to decide whether a sample needs to be projected into a deeper embedding

---

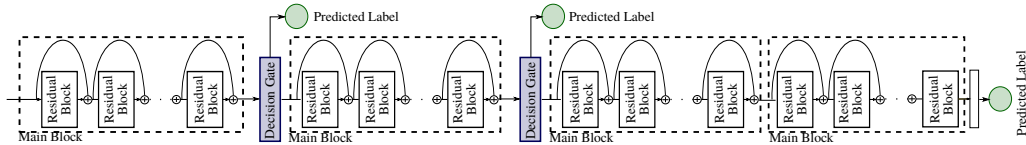[*]The work has been done as a visiting scholar at University of Waterloo

Figure 1: Decision gates (d-gate) are integrated directly into deep neural networks, and are trained to predict whether decisions can be made at the d-gate or require projecting into deep embeddings.

or if an early prediction can be made at the d-gate, thus enabling the conditional computation of dynamic representations at different depths. The proposed d-gate modules can be integrated with any deep neural network without the need to train networks from scratch, and thus reduces the average computational complexity of the deep neural networks while maintaining modeling accuracy.

2 Methodology

Deeper neural network architectures have been demonstrated to provide a better subspace embedding of data when compared to shallower architectures, resulting in better discrimination of data space and better modeling accuracy. Inspired by soft-margin support vector [3] theory, we make a hypothesis that while deeper subspace embeddings are necessary to discriminate samples that lie close to the decision boundaries in the lower embedding space, their effect on samples that already lie far from decision boundaries in the shallower embedding space may be insignificant and unnecessary. Therefore, an effective yet efficient mechanism for determining the distance between samples and the decision boundaries in the lower layers of the network would make it possible to perform early predictions on these samples without projecting them into a deeper embedding space. This approach would reduce the average computational cost of prediction significantly. However, devising an efficient way to determine whether a sample is a boundary sample is a challenging problem.

Here, we formulate the early prediction problem as a risk minimization problem, and introduce a set of single-layer feed-forward networks (which we will refer to as **decision gates** (d-gate)) that are integrated directly into a deep neural network (see Fig. 1). The goal of d-gate modules is to not only decide whether a sample requires projection into a deep embedding space, but also minimize the risk of early wrong classifications as well. Specifically, we train d-gate modules that are integrated into a deep neural network via a hinge loss [5] that minimizes the risk of early misclassification in lower embeddings while deciding whether the sample is a boundary sample:

$$\ell(y) = \max\left(0, 1 - y \cdot \hat{y}\right) \quad \text{s.t.} \quad \hat{y} = w^T x - b \tag{1}$$

where y is the ground truth label for the input data x and $\hat{y}$ is the predicted class label via the d-gate module with the set of weights w and biases b. The set of weights w has a dimensionality of $f \times c$, where f denotes the number of input features to the d-gate module and c denotes the number of class labels in the classification task. This d-gate module provides an important benefit where the result of $w^T x - b$ provides the distances of the sample to the corresponding decision boundary of each class label in the embedding space. Training the d-gate module in this way provides a linear classifier where samples that do not require deeper embeddings for discrimination are those with larger distances (i.e., with the positive sign) from the decision boundary. It is important to note that single layer nature of d-gate modules is designed to account for efficiency.

The d-gate module is trained via the training data utilized to train the deep neural network and the objective for each d-gate module is to minimize the classification error on the training data. Therefore, the loss function on the training data can be formulated as:

$$\mathcal{L}(Y, \hat{Y}; w, b) = \frac{1}{n} \sum_{i=1}^{n} \max\left(0, 1 - y_i(w^T x_i - b)\right) - \lambda \|w\| \tag{2}$$

where Y denotes the set of ground truth labels for all training data. What is most interesting about $\mathcal{L}(Y, \hat{Y}; w, b)$ is the fact that $\mathcal{L}(\cdot)$ is a convex function of w and b, and as such can be optimized via gradient descent. Therefore, traditional gradient descent can be adapted here, where a step is taken in the direction of a vector selected from the function's sub-gradient [15] to find the optimized values. As a result, the d-gate can be trained within a mini-batch training framework, which makes it is very convenient for utilizing in the training of deep neural networks with large datasets.

In essence, the proposed d-gate module can compute the distance of each sample to the decision boundaries based on $w^T x - b$; the calculated distances are compared with the decision threshold t of each d-gate to determine whether early prediction on the sample can be performed at the d-gate or have the sample moved to a deeper stage of the deep neural network to project into a better embedding space for improved prediction. The samples that are far from the decision boundaries result to output larger values in $w^T x - b$; therefore, if the d-gate distance for a sample satisfies the d-gate decision threshold, the class corresponding to the largest distance is assigned as the predicted class label for the sample in this early prediction step.
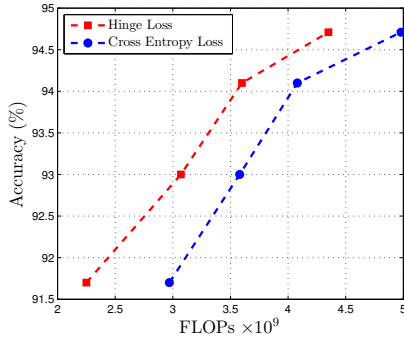
2

Figure 2: Accuracy vs. number of FLOPs: The performance of the network with decision gates trained via the proposed hinge loss is compared with decision gates trained via a conventional cross-entropy approach. It can be observed that the decision gates trained via the hinge loss provide greater computational efficiency with higher accuracy than if cross-entropy loss is leveraged.

Table 1: Experimental results for ResNet-101 and DenseNet-201 with different d-gate configurations. The average number of FLOPS and accuracy for each configuration are compared with that of the original networks. $dg(t1, t2)$ denotes network with two d-gate modules configured with decision thresholds $t1$ and $t2$, respectively.

ResNet-101				DenseNet-201			
Model	ACC.	FLOPS (G)	speed (ms)	Model	ACC.	FLOPS (G)	speed(ms)
dg(1.0, 0.5)	91.7%	2.25	4.245	dg(1.0,1.0)	92.3%	1.55	7.013
dg(1.0, 2.0)	93.0%	3.07	5.397	dg(1.0,2.0)	93.1%	1.75	8.909
dg(1.5, 2.0)	94.1%	3.60	6.898	dg(1.5,2.5)	94.3%	1.96	10.755
dg(2.5, 2.5)	94.7%	4.35	7.747	dg(4.5,4.5)	95.3%	2.52	15.130
Original	94.7%	5.02	8.724	Original	95.3%	2.72	16.466

3 Results and Discussion

The efficacy of the proposed d-gate modules is examined with two different network architectures (ResNet101 [7] and DenseNet201 [9]) on the CIFAR10 dataset. A key benefit of the proposed d-gate modules is that it enables fine control over the trade-off between modeling accuracy and computational cost by adjusting the d-gate decision thresholds. By decreasing the d-gate decision thresholds, the number of samples undergoing early prediction increases, thus reducing the average computational cost of network predictions greatly. For this study, we integrated two d-gate modules in ResNet-101 (after the first and second main blocks) and DenseNet-201 (after the first and second dense blocks), and explore different d-gate configurations. The networks are implemented in the Pytorch framework and the prediction speeds are reported based on single Nvidia Titan Xp GPU.

It can be observed from Table 1 that the computational cost of ResNet network can be reduced by 67 MFLOPS while maintaining the same level of accuracy as to the original ResNet-101 by integrating two d-gate modules with decision thresholds of $(t1, t2) = (2.5, 2.5)$. The integration of d-gate modules can reduce the computational cost of ResNet-101 network by ~39% (i.e., lower by 1.95 GFLOPS) with 1.7% drop in accuracy compared to the original ResNet-101 (with distance thresholds $(t1, t2) = (1.0, 2.0)$ in d-gate1 and d-gate2), resulting in a ~38% speed-up. The experiments for DenseNet-201 show that it is possible to reduce the number of FLOPs by 970 MFLOPs (~36% reduction) with only a ~2% drop in accuracy, leading to a ~46% speed-up. Furthermore, a 2.3× speed-up can be achieved with d-gate modules compared to the original DenseNet-201 within a 3% accuracy margin. Based on the experimental results, the proposed d-gate modules lead to a significant increase in prediction speed, making it well-suited for industrial applications.

In addition to the d-gate modules being proposed, one of the key contributions of this paper is the introduction of a hinge loss for training the d-gate modules. Past studies [11] have argued that cross-entropy results in a small margin between the decision boundaries and the training data. As such, it is very difficult to trust the confidence values of the Softmax layer to decide about the sample since there is no valuable information in the Softmax output. To demonstrate the effectiveness of the hinge loss leveraged in the proposed d-gates compared to the cross-entropy loss, an additional comparative experiment was conducted. More specifically, two decision gates were added to ResNet101 in the same way as reported. However, rather than train using the proposed hinge loss, the decision gates were instead trained via a cross-entropy loss. This enables us to compare the effect of hinge loss vs. cross-entropy loss on decision gate functionality. Fig. 2 demonstrates the accuracy vs. number of FLOPs for the network where the decision gates were trained based on the proposed hinge loss approach compared to trained using a regular cross-entropy training procedure. It can be observed that, with the same number of FLOPs in the network, the network where the decision gates were trained based on the proposed hinge loss provides much higher modeling accuracy compared to that trained via cross-entropy loss. The accuracy gap increases exponentially when the decision gates are configured such that the network uses fewer number of FLOPs. What this illustrates is the aforementioned issue with the use of cross-entropy loss and decision boundaries.

References

[1] Emmanuel Bengio, Pierre-Luc Bacon, Joelle Pineau, and Doina Precup. Conditional computation in neural networks for faster models. *arXiv preprint arXiv:1511.06297*, 2015.

[2] Tolga Bolukbasi, Joseph Wang, Ofer Dekel, and Venkatesh Saligrama. Adaptive neural networks for efficient inference. *arXiv preprint arXiv:1702.07811*, 2017.

[3] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[4] Ludovic Denoyer and Patrick Gallinari. Deep sequential neural network. *arXiv preprint arXiv:1410.0510*, 2014.

[5] Ürün Dogan, Tobias Glasmachers, and Christian Igel. A unified view on multi-class support vector classification. *Journal of Machine Learning Research*, 17(45):1–32, 2016.

[6] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[8] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[9] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, volume 1, page 3, 2017.

[10] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. *arXiv preprint arXiv:1712.05877*, 2017.

[11] Xuezhi Liang, Xiaobo Wang, Zhen Lei, Shengcai Liao, and Stan Z Li. Soft-margin softmax for deep classification. In *International Conference on Neural Information Processing*, pages 413–421. Springer, 2017.

[12] Lanlan Liu and Jia Deng. Dynamic deep neural networks: Optimizing accuracy-efficiency trade-offs by selective execution. *arXiv preprint arXiv:1701.00299*, 2017.

[13] M. Shafiee, A. Mishra, and A. Wong. Deep learning with darwin: Evolutionary synthesis of deep neural networks. *arXiv:1606.04393*, 2016.

[14] M. Shafiee and A. Wong. Evolutionary synthesis of deep neural networks via synaptic cluster-driven genetic encoding. In *NIPS Workshop*, 2016.

[15] Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical programming*, 127(1):3–30, 2011.

[16] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[17] Surat Teerapittayanon, Bradley McDanel, and HT Kung. Branchynet: Fast inference via early exiting from deep neural networks. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 2464–2469. IEEE, 2016.

[18] Zuxuan Wu, Tushar Nagarajan, Abhishek Kumar, Steven Rennie, Larry S Davis, Kristen Grauman, and Rogerio Feris. Blockdrop: Dynamic inference paths in residual networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8817–8826, 2018.

[19] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5987–5995. IEEE, 2017.