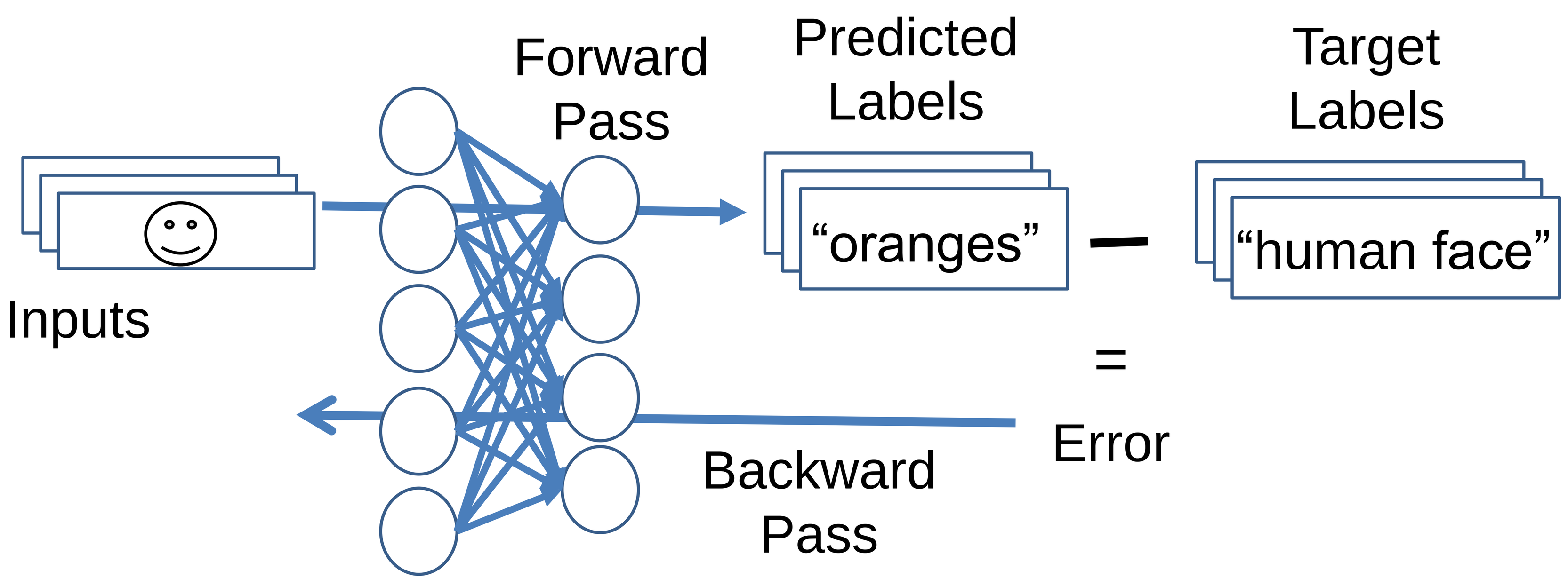
Precomputing Outputs of Hidden Layers to Speed Up Deep Neural Network Training

Sohil Lal Shrestha

Introduction and Motivation

- Training a deep neural network(DNN) is time consuming and resource intensive task
- State-of-the-art model [1] for image classification took 5-6 days to train on two 3GB GPUs
- Andrew et al. [2]'s Freezeout technique reported 3% loss in accuracy saving 20% in training time
- Our PreFast technique, built on top of FreezeOut, shows up to 20-25% further improvement in training time with no loss in accuracy.

Background and Related Work



Deep neural network training consists of series of forward and backward passes through hidden layers

Training neural networks involves two steps:

(1) Forward Pass: propagates input forward through the network

(2) Backward Pass: Propagate errors backward to compute the weight gradients

Goal of training: Search weight parameters of each layer

Approaches to optimizing training time of DNN can be generalized into two category:

(1) Algorithm approach exploits training algorithm.

(2) System approach uses parallelized training.

Algorithm Technique

(1) Each layer (L_i)'s is initialized to single learning rate value (α)

(2) Learning rate anneals to zero at t_i, where t_i is linearly spaced between a user-selected t_0 and the total number of iterations t for training.

(3) Learning rate is decreased using cosine annealing scheme. Thus each layer's learning rate at iteration t is

$$\alpha_i(t) = 0.5 * \alpha_i(0)\ (1 + \cos(\frac{\pi t}{t_i}))$$

(4) Freeze the layer once the learning rate reaches zero.

Possible UseCase and Limitation

Whether tradeoff between validation error and training time is acceptable to user or not is entirely up to user's use case:

- Helpful for users for prototyping different neural network design and quickly assessing which one performs better
- If the users goal is to get a high performing neural network design, the technique is not applicable.

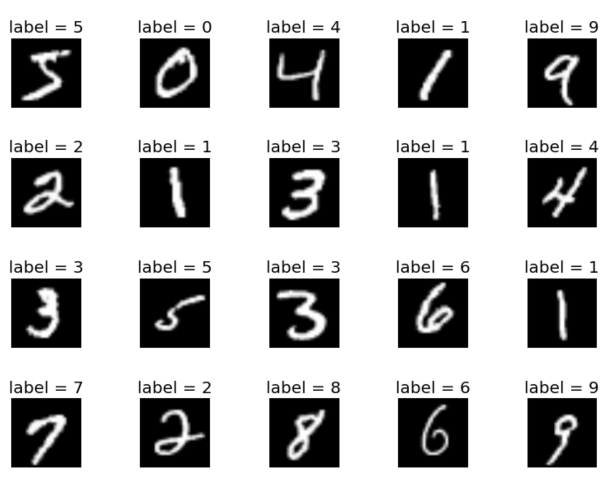


Image taken from [5]

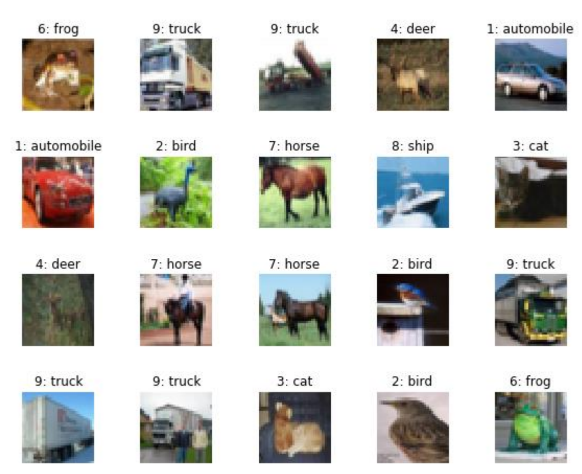


Image taken from [3]

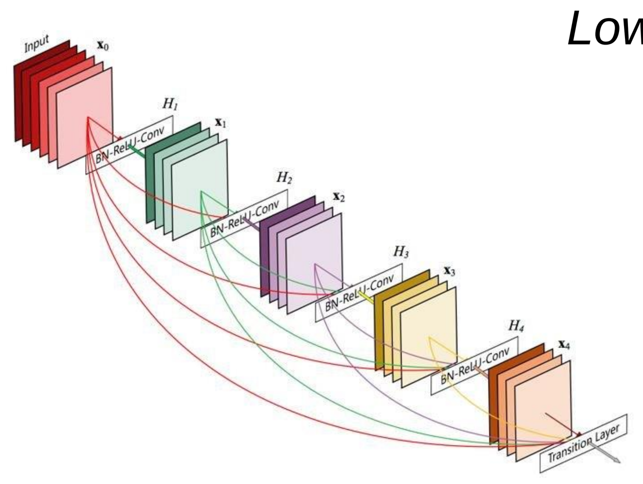
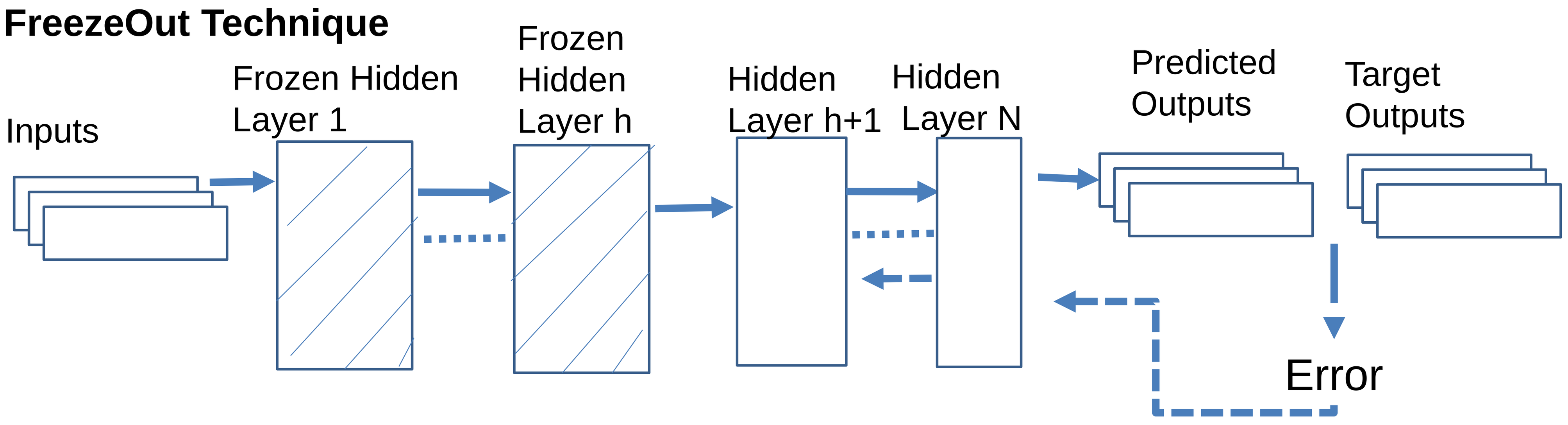


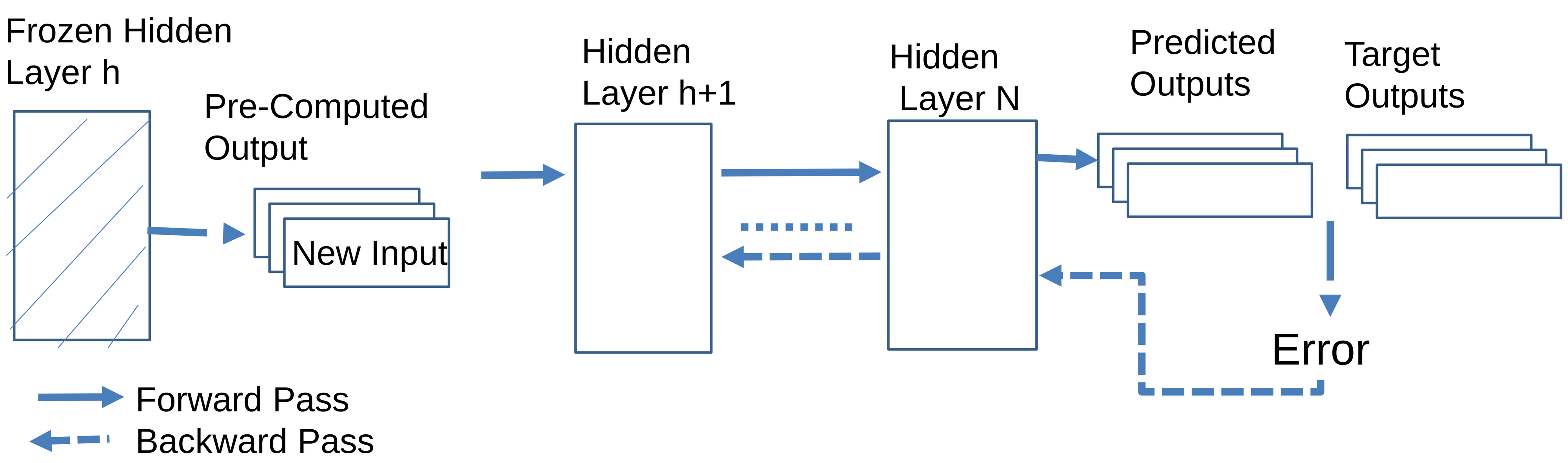
Image taken from [4]

Overview and Design

FreezeOut Technique



PreFast Technique



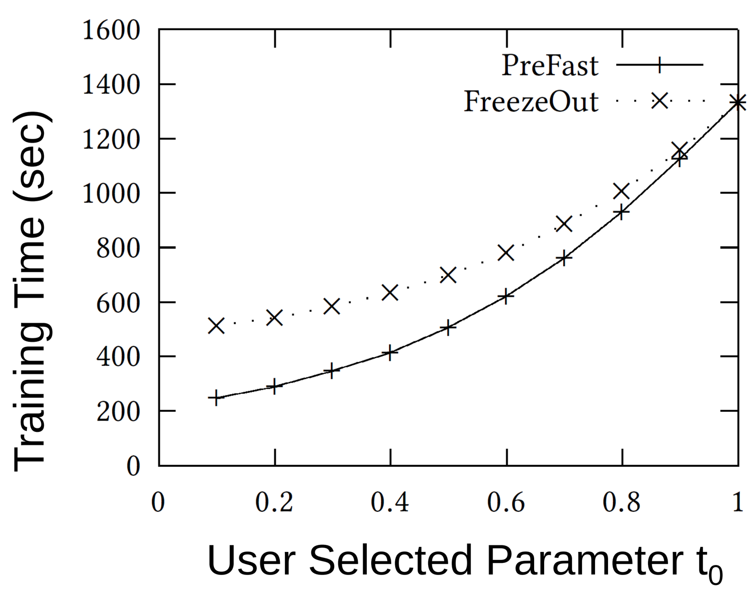
PreFast replaces forward pass through frozen layer with pre-computed outputs

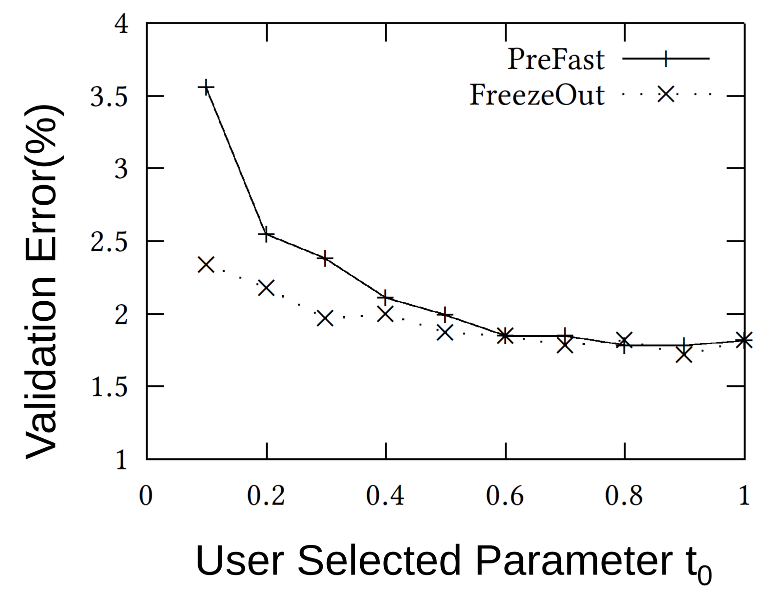
- FreezeOut excludes frozen layer only through backward pass
- PreFast precomputes the output from frozen layer
- Frozen layer's output can be computed as new input to subsequent layers
- Bypasses the frozen layer completely

Experimental Setup

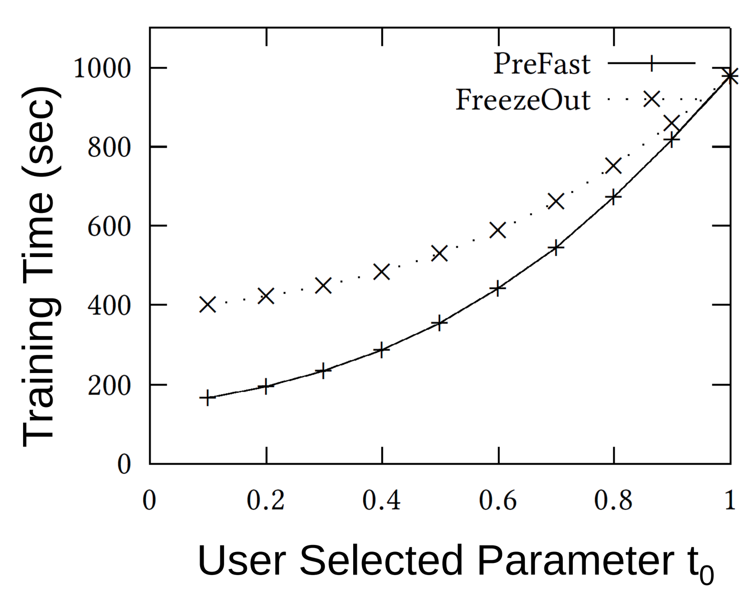
- Modified publicly available version of FreezeOut code
- Trained state-of-the-art DenseNet [4] model for image classification.
- Datasets Used: MNIST [5], CIFAR [3]
- Random sampled MNIST dataset to 4000 and CIFAR to 2000 samples due to resource constraint.
- Compared training time and validation error between PreFast and FreezeOut using different t_0 values
- Tested on Texas Advanced Computing Center Maverick (TACC) with CentOS 6.4, CPU - Intel Xeon E5-2680 v2 Ivy Bridge, 2.80 GHz, 20 CPUs/node, NVIDIA Tesla K40 GPU and Cuda 7.5.

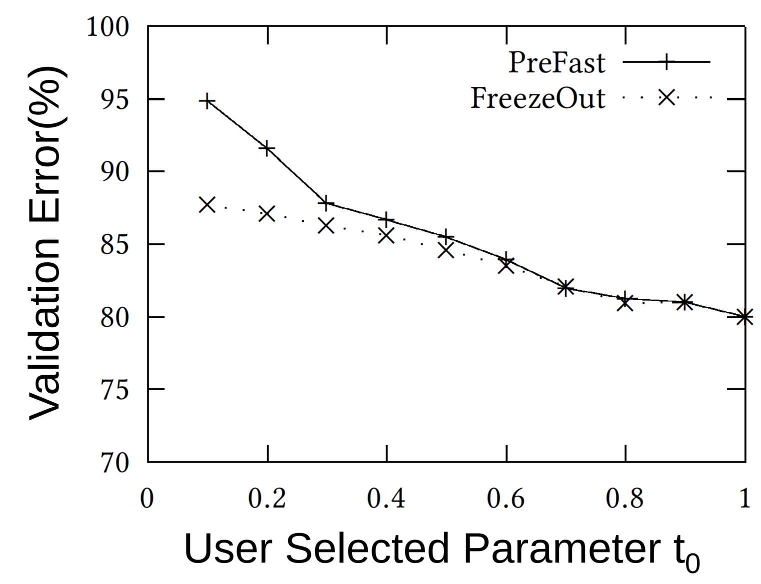
Empirical Results





Lower training time for similar validation errors on MNIST Dataset





Lower training time for similar validation errors using CIFAR-100 Dataset

[1] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). **Imagenet classification with deep convolutional neural networks**. In proc of NIPS

[2] Brock, A., Lim, T., Ritchie, J. M., & Weston, N. (2017). **FreezeOut: Accelerate Training by Progressively Freezing** , in NIPS workshop

[3] Krizhevsky, A., & Hinton, G. (2009). **Learning multiple layers of features from tiny images** (Vol. 1, No. 4, p. 7). Technical report, University of Toronto.

[4] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). **Densely Connected Convolutional Networks**. In CVPR

[5] LeCun, Y. & Cortes, C. (2010). **MNIST handwritten digit database**.

Contact Information:
Sohil Lal Shrestha sohil.shrestha@mavs.uta.edu

Software Engineering Research Center (SERC)
The University of Texas at Arlington