Generalisation Analysis of Deep Frank-Wolfe

Suraj Sudhakar
Saarland University, Germany

Abstract—**Stochastic Gradient Descent (SGD) is a ubiquitous algorithm due to the ability of the algorithm to a. solve a wide variety of problems and b. produce solutions that generalise well. But it suffers from the drawback of requiring a handcrafted scheduler. To overcome these problems, "Frank-Wolfe" can be used. However, traditional Frank-Wolfe is computationally more expensive than SGD, so a variant, "Deep Frank-Wolfe" (DFW) with computational efficiency similar to SGD was proposed by the authors [1]. DFW calculates the optimal step size in a closed manner thus not requiring a scheduler. However, for it to be considered a viable alternative to SGD and its adaptive variants, the algorithm must achieve a similar generalisation as SGD. One of the ways to test an algorithm generalisation is to study its performance under differing hyper-parameter values. Though studied to an extent in the original paper, this paper further tests & extends the claim of generalisation under this paradigm but is not limited to it.**

I. Introduction

Frank Wolfe is a first order, iterative, constrained optimisation algorithm with the potential to replace SGD for various situations. SGD requires separate schedulers for its learning rate, moreover, they are problem dependant, which only worsens the issue. Adaptive variants were introduced to solve the issue, but they produce solutions that do not generalise as well. Deep Frank Wolfe algorithm can be used to alleviate both the issues in an efficient manner with only one hyper-parameter, i.e. initial learning rate. DFW exploits the composite structure of deep neural networks to design an optimization algorithm that leverages efficient convex solvers while handling the decay of learning rate. The inherent decaying step makes the algorithm more robust.

While the algorithm has been tested against different initial learning rates in the original paper, DFW is an approximation of Frank-Wolfe. This approximation is sensitive to regularisation term used in the loss function as claimed in the original paper. This necessitates further experimentation in this direction, testing the sensitivity of algorithm for different regularisation's and regularisation parameters.

SGD has gained such a high popularity due to its robustness. The ability to efficiently optimise on pre-trained models and transfer learning scenarios is very much desired. DFW algorithm should also be investigated in this regard.

Experiments were designed to test the DFW algorithm for the aforementioned scenarios. All experiments were conducted against classic image classification problem. The accuracy of the model analysed & tabulated.

The rest of the paper is organised as follows, first, discuss the formulation of DFW from Frank-Wolfe. Followed by the design of Experiments, and discussion of Results & Conclusion.

II. The Deep Frank-Wolfe Algorithm

To exploit the composite nature of Deep Neural Networks, proximal framework is utilised and then extended to get DFW.

A. PROXIMAL APPROACH

SGD algorithm can be formulated as follows:

$$w_{t+1} = argmin_w\{\frac{1}{2\eta_t}\|w-w_t\|^2+T_{wt}\rho(w)+T_{wt}[L_j(f_j(w))]\} \tag{1}$$

where, w_t is the weights on 't'th iteration. η denotes the initial learning rate & T_{wt} represents the Taylor linearisation of the function.

Loss preserving Linearisation: Proximal problem is used by only linearising the inner function without the loss as follows:

$$w_{t+1} = argmin_w\{\frac{1}{2\eta_t}\|w-w_t\|^2+T_{wt}\rho(w)+L_j(T_{wt}(f_j(w)))\} \tag{2}$$

By using the formulation in equation (2), dual can be obtained for Frank-Wolfe which will have similar formulation and cost as SGD.

Proposition 1: Problem (2) with a hinge loss is amenable to optimization with Frank-Wolfe in the dual, which yields an optimal step-size in closed-form at each iteration.

To obtain the direction of the conditional gradient,

Proposition 2: If a single step is performed on the dual of (2) its conditional gradient is given by $-\delta(\rho(w)+L_y(f_x(w)))$

Computational cost per iteration is same as that for SGD, since the update only requires only SGD gradients.The algorithm has no theoretical proof of convergence The conditional gradient provided by the second proposition is exact only if $\rho = 0$, in all other cases it is an approximation. That is one of the reasons to test it against different Regularisation parameters.

III. Setup

In order to test the DFW algorithm on the discussed scenarios, multiple runs of the model have to be performed. So, a simple CNN model was used rather than Wide Resnet/DenseNet (as in the original paper). The CNN model consisted of 3 Convolution layers each with filters of size 5*5 with stride = 1 & padding to maintain the same size output. Number of filters used was 16, 32, 32 in that order. ReLU activation, and Max Pooling operations were performed after each convolution block & a dropout layer was incorporated after the third convolution layer.

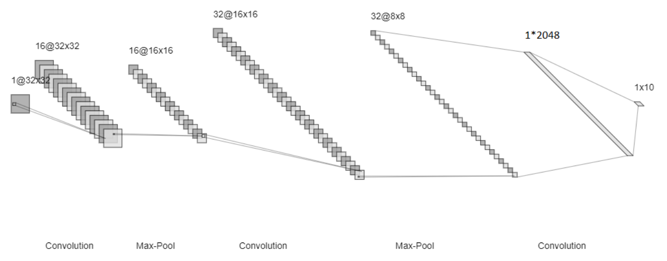


Fig. 1. Model architecture

Note, the architecture is kept simple as the aim is not to replicate SOTA results, instead to try and test the optimisation algorithm.

Mainly CIFAR10 dataset was used, with MNIST dataset being used to test only the transfer learning scenario. For all runs, a fixed batch size of 64, Number of epochs, 50, was used. All of the experiments were conducted on both SGD and DFW for comparison purposes. The SGD algorithm was subjected to a learning rate decay of 0.85 for every 3 epochs while the DFW handled the decay internally. **Regularisation:** Data augmentation, Dropouts with factor of '0.2' were optionally used and regularisation options were L2/L1/No reg. Objective was the same in all the runs, Image classification.

IV. Experiments & Results

Experiments were carried out to test different claims regarding the regularisation. The experimental setup and result for each is discussed in a separate subsection for each experiment.

A. *VARYING LEARNING RATE*

Authors claim that higher initial learning rate leads to better generalisation even if the final validation loss is slightly lesser for the run with lower initial learning rate. Both SGD and DFW were trained with two learning rates 1e-3 & 1e-1. The results are tabulated in Table 1

Algorithm	Learning rate	Validation Score	Test Accuracy
SGD	1e-1	0.035	69.75%
	1e-3	0.02	51.52%
DFW	1e-1	0.008	70.37%
	1e-3	0.01	61.24%

Table 1

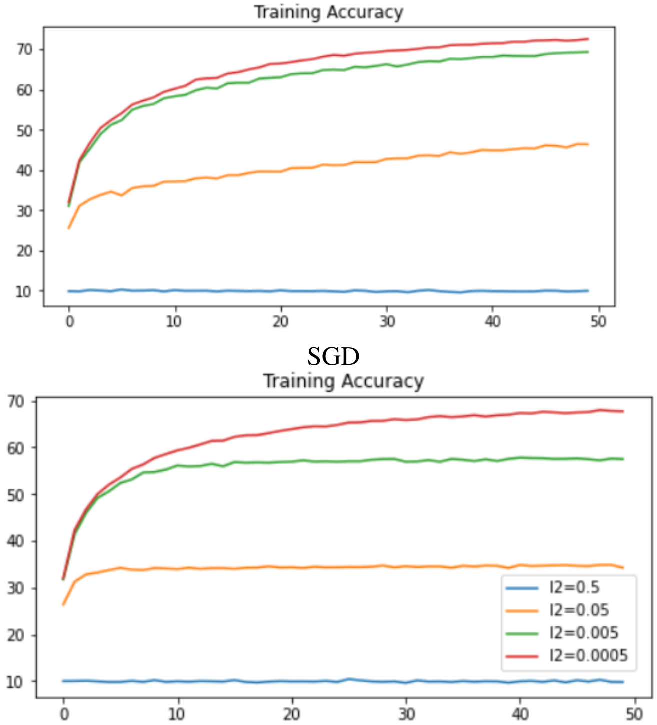
The results obtained re-affirm the claims made by the authors, also establishing a suitable baseline for my further experiments. Since we found that learning rate 1e-1 was giving good results, all further experiments were conducted with this initial learning rate.

B. *L2 regularisation*

As mentioned, if the DFW algorithm is to be considered robust, it should produce results which are stable for different values of regularisation parameter. 4 different values of regularisation parameter were used and results are tabulated in Table 2.

Regularisation value	Test Accuracy for SGD	Test Accuracy for DFW
0.5	10.0%	10.0 %
0.05	48.73%	40.9 %
0.005	67.71%	64.73 %
0.0005	72.5%	71.86 %

Table 2



SGD

DFW

Fig. 2. Training accuracy for different l2 values

The training curves can referred to in Figure 2. The behaviour of DFW algorithm is pretty stable and comparable to that of SGD, except for second case. It can be concluded that DFW is robust against different L2 regularisation values. Also, as a secondary note, DFW algorithm seems to have a slightly faster convergence rate.

C. *L1 regularisation*

Similar experiment was conducted this time using L1 regularisation and only two variants for L1 parameter. Refer Table 3.

Regularisation value	Test Accuracy for SGD	Test Accuracy for DFW
0.005	10.0%	10.0 %
0.0005	62.2%	52.71 %

Table 3

Difference in performance is noticeable for L1 regularisation. By taking into account the second result of Table2 and second result of Table 3. It can be concluded that higher regularisation has more negative effect on DFW compared to SGD, though the difference is not too high.

D. PRETRAINED

Pretrained VGG11 network was used and the full model was trained against SGD and DFW, again for 50 epochs. The final accuracy and loss curves are compared in Table 3 and Figure 3 respectively.

Algorithm	Test Accuracy
SGD	91.72%
DFW	90.15%

Table 4

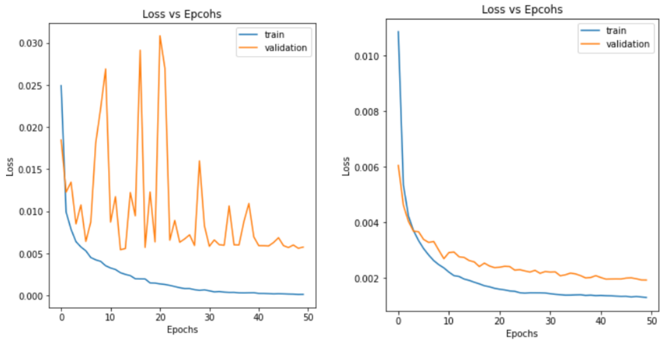


Fig. 3. Training & Validation Loss on pretrained model for SGD & DFW respectively

The curve obtained for DFW is much smoother than for SGD due to inherent learning rate decay. The decrease in spike of validation loss for SGD can be attributed to when decay in learning rate comes into effect. Though the final accuracy is comparable.

E. TRANSFER LEARNING

A model was initially trained on CIFAR10 dataset, it was then finetuned and tested on MNIST dataset. The procedure was applied to both SGD and DFW, accuracy and loss are compared in Table 5 and Figure 4

Algorithm	Test Accuracy
SGD	98.92%
DFW	98.75%

Table 5

Transfer learning is a good test of generalisation capacity of the algorithm. DFW again has results comparable to SGD.

V. Conclusion

Deep Frank Wolfe algorithm was proposed as a good alternative to SGD. DFW clearly has similar generalisation capacity as that of SGD for the problem of Image classification. Though only the generalisation related results are tabulated, it can be inferred from the loss curves that DFW also has convergence rate slighlty faster than SGD. However, the current version of the algorithm only works for SVM loss function. The empirical promise shown by DFW is sufficient to warrant further research into it and the possibility of using the optimiser to solve a wider variety of problems.

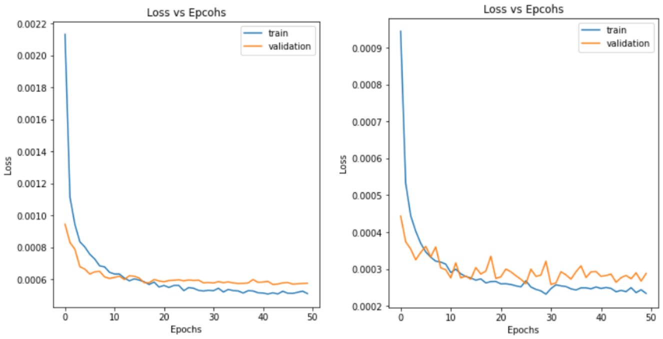


Fig. 4. Training & Validation Loss for transfer learning for SGD & DFW respectively

References

[1] A. Z. Leonard Berrada and M. P. Kumar, "Deep frank-wolfe for neural network optimization," 2019.

VI. APPENDIX

Code will be uploaded to: https://github.com/s-suraj-08/FrankWolfeDeepLearning