Training a Neural Network with Stochastic Frank-Wolfe

Optimation for Data Science Course Project
Data Science–University of Padua

October 8, 2022

Overview

- Stochastic variants
- Regularization
- Projection free algorithm
- $\min_{x\in\Omega} F(x) = \min_{x\in\Omega} \frac{1}{m}\sum_{i=0}^{m} f_i(x)$
- $\min_{\theta\in\Omega} F(\theta) = -\frac{1}{m}\sum_{i=0}^{m}(y^i log(\hat{y}^i) + (1-y^i)log(1-\hat{y}^i))$

- L_1-ball constraint
- $C(radius) = \{x \in R^n : ||x||_1 \leq radius\}$
- $v_t = \arg\min_{v \in c} \langle \tilde{\nabla}(F_t, v) \rangle$
- Result of the LMO:

$$v_t = \begin{cases} diameter \times sign(-\nabla_{i_k} f(\theta_k)) \cdot, & \text{if } i_k = \arg\max_i |\nabla_i f(\theta_k)|. \\ 0 & \text{otherwise.} \end{cases}$$

- $\theta_{t+1} = \theta_t + \alpha(v_t - \theta_t)$

Algorithm Stochastic Frank-Wolfe method for l_1-ball

Require: Starting from a point inside the region
  for k=1,.... **do**
    Uniformly sample i.i.d. $i_1.i_2, ..., i_b$ from $[1, .., n]$
    $\tilde{\nabla} L(\theta_k) \leftarrow \frac{1}{b} \sum_{j=1}^{b} \nabla f_{i_j}(\theta_k)$
    Set $\hat{\theta}_k = diameter \times sign(-\tilde{\nabla}_{i_k} L(\theta_k))$, with $i_k = \arg\max_i |\tilde{\nabla}_i L(\theta_k)|$
    if $\hat{\theta}_k$ satisfies some specific condition, then STOP
    Set $\theta_{k+1} = \theta_k + \alpha_k(\hat{\theta}_k - \theta_k)$
  end for

Algorithm Stochastic Variance Reduced Frank-Wolfe method for l_1-ball

Require: Starting from a point inside the region
  for t=0,...,S-1 **do**
    take snapshot $\theta_0 = \theta_t$ and compute $\nabla F(\theta_0)$
    for k=1,....m-1 **do**
      Uniformly sample i.i.d. $i_1.i_2, ..., i_b$ from $[1, .., n]$
      $\tilde{\nabla} F(\theta_k) \leftarrow \nabla F(x_0) + \frac{1}{b_k}\sum_{j=1}^{b_k}(\nabla f_{i_j}(x_k) - \nabla f_{i_j}(\theta_0))$
      Set $\hat{\theta}_k = diameter \times sign(-\tilde{\nabla}_{i_k} F(\theta_k))$, with $i_k = \arg\max_i |\tilde{\nabla}_i F(\theta_k)|$
      Set $\theta_{k+1} = \theta_k + \alpha_k(\hat{\theta}_k - \theta_k)$
    end for
  $\theta_{t+1} \leftarrow \theta_{K_t}$
  end for

- Initialization of parameter
- Forward Propagation
- Backward propagation
- Updating Parameters
- Prediction function
- Cost function
- Main function

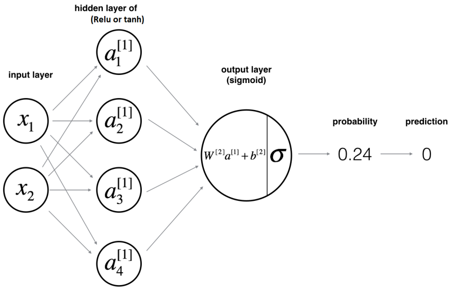


Table: SFW hyperparameters, with ReLu as the activation function

Data set	Learning rate	Batch size	l_1 ball diameter	Epochs	Hidden unit size
fashion mnist	0.001	128	5	20	32
moon	0.0008	32	3	10	16
fruit	0.001	32	20	10	64

Table: SVRF hyperparameters, with ReLu as the activation function

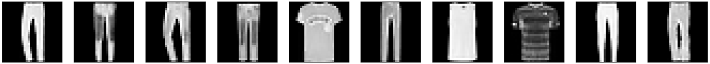
Data set	Learning rate	Batch size	l_1 ball diameter	Epochs	Hidden unit size	inner loop size
fashion mnist	0.005	128	5	20	32	20
moon	0.003	32	3	10	16	10
fruit	0.001	32	20	10	64	10

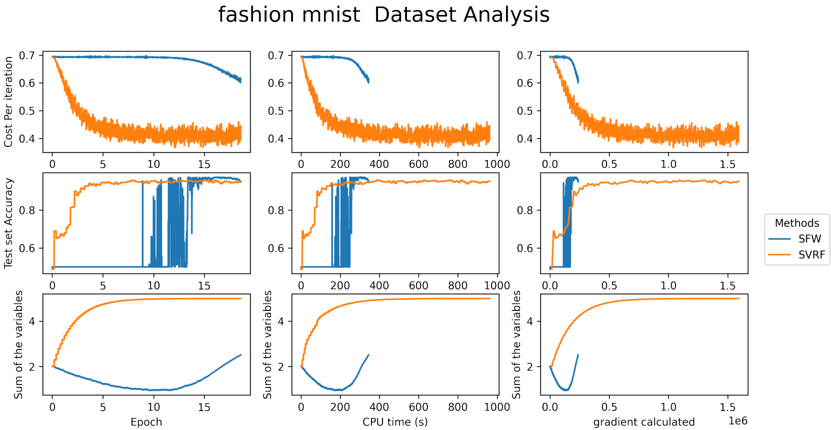
Table: Test Set Accuracy

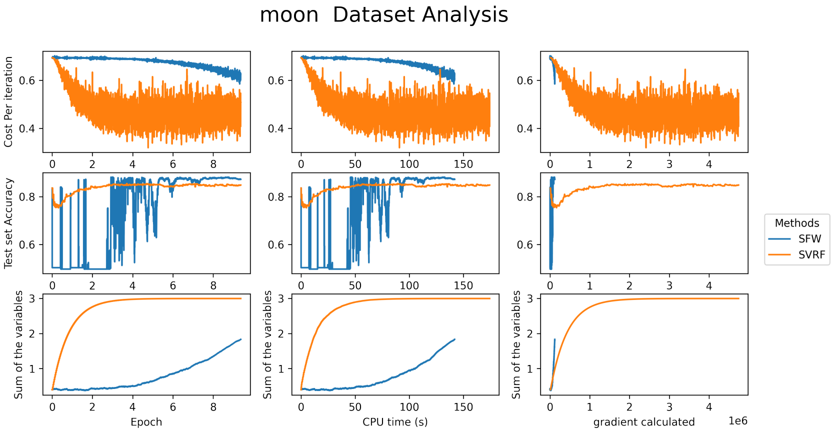
Dataset	SFW	SVRF
fashion mnist	94.9%	**95.2%**
moon	**87.3%**	84.5%
fruit	86.7%	**91.9%**

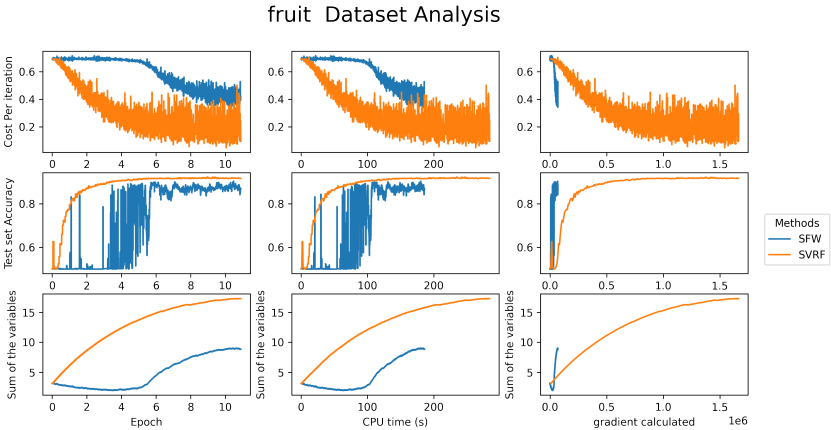
Table: Training Loss

Dataset	SFW	SVRF
fashion mnist	0.613	**0.4371**
moon	0.625	**0.508**
fruit	0.415	**0.095**




fashion mnist Dataset Analysis
Cost Per iteration
Test set Accuracy
Sum of the variables
Epoch
CPU time (s)
gradient calculated
1e6
Methods
SFW
SVRF


moon Dataset Analysis
Cost Per iteration
Test set Accuracy
Sum of the variables
Epoch
CPU time (s)
gradient calculated
1e6
Methods
SFW
SVRF


fruit Dataset Analysis
Cost Per iteration
Test set Accuracy
Sum of the variables
Epoch
CPU time (s)
gradient calculated
1e6
Methods
SFW
SVRF

Thank you