Weight sharing for single-channel LMS

Shamahil Ibunu, Karl Moore and Clive Cheong Took
Department of Electronic Engineering
Royal Holloway, University of London
TW20 0EX United Kingdom
{phee004, pjee001}@live.rhul.ac.uk, clive.cheongtook@rhul.ac.uk

Danilo Mandic
Department of Electrical and Electronic Engineering
Imperial College London
SW7 2BX United Kingdom
d.mandic@ic.ac.uk

***Abstract*—Constraining a group of taps of an adaptive filter to a single value may seem like a futile task, as weight sharing reduces the degree of freedom of the algorithm, and there are no obvious advantages for implementing such an update scheme. On the other hand, weight sharing is popular in deep learning and underpins the success of convolutional neural networks (CNNs) in numerous applications. To this end, we investigate the advantages of weight sharing in single-channel least mean square (LMS), and propose weight sharing LMS (WSLMS) and partial weight sharing LMS (PWS). In particular, we illustrate how weight sharing can lead to numerous benefits such as an enhanced robustness to noise and a computational cost that is *independent* of the filter length. Simulations support the analysis.**

I. Introduction

Weight sharing is a common practice in machine learning due to its numerous benefits such as minimising the risk of overfitting to the data and easier hyperparameter tuning due to a reduction in the number of parameters. However, this low computational complexity advantage is not the main objective for weight sharing. The main motivation for weight sharing is that it enforces the algorithm to detect *common features* scattered across the data [1]. As such, this particular property is not exhibited by LMS-type algorithms designed to have low computational complexities such as [2], the partial update LMS [3], and the online censoring algorithms [4], [5].

In fact, properties such as robustness to noise, convergence and stability are usually of high interest to the signal processing community including adaptive filtering algorithms. As such, the main aim for reducing the degrees of freedom of algorithms is to alleviate the computational cost at the expense of these benefits. Weight sharing (WS) goes beyond the issue of computational complexity. It was introduced for multichannel least mean square (MLMS) algorithms and the desirable properties of WS were also investigated in our work in [6]. In particular, weight sharing has been shown to lead to i) an improved stability of the adaptive algorithms especially for high condition numbers; ii) an enhanced robustness to noise; and iii) its ability to cope with large number of channels such as in massive multiple inputs multiple outputs (MIMO) applications.

Our previous work implemented weight sharing across different channels as shown in Section II-A. This present work, instead, considers weight sharing *across time* for single channel processing. Moreover, *partial* weight sharing is also considered to enhance the learning 'capacity' for more complex signals/systems by relaxing the strict requirements of full weight sharing [6].

II. Weight sharing for Adaptive filtering

A. Weight Sharing for Multichannel LMS

The weight sharing multichannel least mean square (WS-MLMS) algorithm was introduced to illustrate how weight sharing can be incorporated into the class of LMS algorithms. In the context of multichannel adaptive processing, weights $\mathbf{w}(n)$ can be shared across data channels as [6]

$$y_q(n) = \mathbf{w}^{\mathrm{T}}(n)\mathbf{x}_q(n) \qquad q = 1, \dots, Q \tag{1}$$

where $x_q(n)$ and $y_q(n)$ represent the qth input and output channel respectively and $(\cdot)^T$ is the transpose operator. Weight sharing mechanism can be expressed as

$$\begin{bmatrix} y_1(n) \\ y_2(n) \\ \vdots \\ y_Q(n) \end{bmatrix} = \sum_{\ell=1}^{L} w_\ell(n) \begin{bmatrix} x_1(n-\ell+1) \\ x_2(n-\ell+1) \\ \vdots \\ x_Q(n-\ell+1) \end{bmatrix} \tag{2}$$

where $w_\ell(n)$ denotes the weight for the ℓth lag of the tap input. This work addresses weight sharing *across time* and introduces the weight sharing LMS (WSLMS) and the partial weight sharing (PWS) LMS.

B. Proposed Weight Sharing LMS (WSLMS)

In weight sharing LMS (WSLMS), a *single* coefficient filtering $w(n)$ can be shared amongst all taps as

$$y(n) = w(n) \sum_{\ell=1}^{L} x(n-\ell+1) \tag{3}$$

Based on the instantaneous quadratic error $e^2(n)$, it can be shown that the weight update for WSLMS is given by

$$w(n+1) = w(n) + \mu e(n) \sum_{\ell=1}^{L} x(n-\ell+1) \tag{4}$$

Eq. (3), however, imposes a strict form of weight sharing in WSLMS, and may not be appropriate in models where the coefficients are significantly different from each other. To address this shortcoming, a more flexible model of weight sharing is proposed in the subsequent algorithm referred as partial weight sharing (PWS). PWS allows some taps to share the same coefficient, whilst also enabling the other taps to have different values. In this way, PWS can benefit from the additional degrees of freedom, whilst exhibiting the advantageous properties of weight sharing as mentioned in the introduction.

C. Proposed Partial Weight Sharing LMS (PWS)

In PWS, the first K taps are allowed to have different weight coefficients, whilst the last $L-K$ taps share the same weight. As such, the output $y(n)$ can be computed as

$$y(n) = \sum_{\ell=1}^{K} w_\ell(n) x(n-\ell+1) + w(n) \sum_{\ell=K+1}^{L} x(n-\ell+1) \quad (5)$$

For the first K taps, the ℓth coefficient $w_\ell(n)$ can updated as

$$w_\ell(n+1) = w_\ell(n) + \mu_{\text{Partial1}} [e(n) x(n-\ell+1)] \qquad \ell = 1, ..., K \quad (6)$$

The last $L-K$ taps share the same weight $w(n)$, which can be updated as

$$w(n+1) = w(n) + \mu_{\text{Partial2}} [e(n) \sum_{\ell=K+1}^{L} x(n-\ell+1)] \quad (7)$$

Observe that (6) and (7) requires two different stepsizes, μ_{Partial1} and μ_{Partial2} which is discussed in the next section.

III. Comparative analyses

This section analyses and compares the proposed algorithms WSLMS and PWS with LMS from a practical perspective. In particular, their robustness against noise, together with their computational and convergence properties are next analysed.

A. Robustness against noise

While LMS is known to be robust against noise compared to the recursive least squares at low signal-noise-ratio (SNR) values [7], its weight update relies on the raw values of the input $x(n)$. Its weight update does not have any inbuilt mechanism to combat the effect of the noise. Unlike the LMS, the update for WSLMS (4) averages/smoothes its input as:

$$\begin{aligned} w(n+1) &= w(n) + \mu e(n) \sum_{\ell=1}^{L} x(n-\ell+1) \\ &= w(n) + \eta e(n) \underbrace{\frac{1}{L} \sum_{\ell=1}^{L} x(n-\ell+1)}_{\text{Moving average of inputs}} \quad (8) \end{aligned}$$

Similarly, PWS benefits from the moving average mechanism in (7), which mitigates the effects of noise (albeit to a lesser extent). As such, it is expected that PWS performs better than LMS in noisy settings, yet worse than WSLMS.

Step	Algorithm	Addition	Multiplication
Estimate $y(n)$	LMS	$L-1$	L
	WSLMS (3)	$L-1$	1
	PWS (5)	$L-1$	$K+1$
Error $e(n)$	LMS	1	0
	WSLMS	1	0
	PWS	1	0
Update $\mathbf{w}(n+1)$	LMS	L	$L+1$
	WSLMS (4)	$L+1$	2
	PWS (6) and (7)	L	$K+2$
Total	LMS	O(L)	O(L)
	WSLMS	O(L)	O(1)
	PWS	O(L)	O(K)

TABLE I
Computational complexities of LMS, WSLMS, and PWS algorithms, where L represents the filter length, and K the number of non-shared coefficients in PWS.

B. Computational Complexities

Table II compares the computational cost of each algorithmic step of LMS, WSLMS, and PWS. They all have the same computational cost in terms of additions.

However, the computational cost for multiplication is significantly different for each algorithm. For WSLMS, the computational cost is constant, and does not depend on the filter length. For PWS, the computational cost is dependent on the number of non-shared weights. For example, if only 50% of the weights are updated, the computational cost is reduced by half - a non-trivial benefit for long filters.

C. Convergence issues

An optimal stepsize plays a fundamental role in ensuring the convergence of an adaptive algorithm. To this end, optimal stepsizes are derived based on the one-step ahead error prediction $e(n+1)$. A first order approximation of a Taylor series expansion of a function $f(n)$ can be expressed as

$$f(n+\Delta n) \approx f(n) + \Delta n f'(n) \quad (9)$$

where $f'(n)$ denotes the first order derivative. Thus, the one-step ahead error prediction $e(n+1)$ can be approximated as

$$e(n+1) \approx e(n) + \sum_{\ell=1}^{L} \frac{\partial e(n)}{\partial w_\ell(n)} \Delta w_\ell(n) \quad (10)$$

which provides the basis for deriving optimal stepsizes.

1) Optimal Stepsize for LMS: For the LMS algorithm, observe that both $\frac{\partial e(n)}{\partial w_\ell(n)}$ and $\Delta w_\ell(n)$ depends on the *individual* input $x(n-\ell+1)$, i.e.

$$\frac{\partial e(n)}{\partial w_\ell(n)} = -x(n-\ell+1) \quad (11)$$

$$\Delta w_\ell(n) = \mu x(n-\ell+1) e(n) \quad (12)$$

Replacing (11)-(12) into (10) yields the one-step ahead error prediction, given by

$$e(n+1) \approx e(n) - \mu e(n) \sum_{\ell=1}^{L} x^2(n-\ell+1) \quad (13)$$

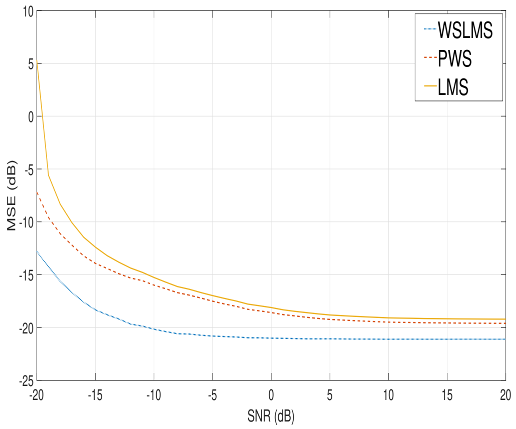

Fig. 1. Evaluation the robustness of LMS, WSLMS, and PWS over a range of signal-to-noise ratio values.

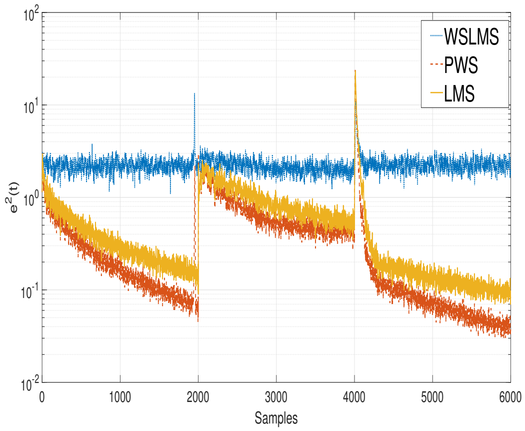

Fig. 2. Transient performance of algorithms considered for the identification a time-varying system.

Factorising the common terms in (13) leads to

$$e(n+1) \approx e(n)\left[1-\mu\sum_{\ell=1}^{L}x^2(n-\ell+1)\right] \tag{14}$$

For the one-step ahead quadratic error to go to zero, we have

$$\left[1-\mu\sum_{\ell=1}^{L}x^2(n-\ell+1)\right]=0 \tag{15}$$

which leads to the well-known normalised stepsize

$$\mu_{\text{LMS}}=\frac{1}{\sum_{\ell=1}^{L}x^2(n-\ell+1)} \tag{16}$$

2) Optimal Stepsize for WSLMS: Observe that both $\frac{\partial e(n)}{\partial w_\kappa(n)}$ and $\Delta w_\kappa(n)$ depend on the *sum* of inputs, i.e.

$$\frac{\partial e(n)}{\partial w_\kappa(n)}=-\sum_{\kappa=1}^{L}x(n-\kappa+1) \tag{17}$$

$$\Delta w_\kappa(n)=\mu\sum_{\kappa=1}^{L}x(n-\kappa+1)e(n) \tag{18}$$

Replacing (17)-(18) into (10) yields the one-step ahead error:

$$\begin{aligned} e(n+1) &\approx e(n)-\sum_{\ell=1}^{L}\mu e(n)\left[\sum_{\kappa=1}^{L}x(n-\kappa+1)\right]^2 \\ &\approx e(n)-\mu e(n)L\left[\sum_{\kappa=1}^{L}x(n-\kappa+1)\right]^2 \end{aligned} \tag{19}$$

Factorising the common error term in (19) leads to

$$e(n+1)\approx e(n)\left[1-\mu L\left(\sum_{\kappa=1}^{L}x(n-\kappa+1)\right)^2\right] \tag{20}$$

$$\therefore \quad \mu_{\text{WSLMS}}=\frac{1}{L\left(\sum_{\kappa=1}^{L}x(n-\kappa+1)\right)^2} \tag{21}$$

Algorithms	LMS μ_{LMS}
Weight Sharing (4) μ_{WS}	$1/L$
Partial Weight Sharing (6) μ_{Partial1}	L/K
Partial Weight Sharing (7) μ_{Partial2}	$L/(L-K)^2$

TABLE II
COMPARATIVE ANALYSIS IN TERMS OF OPTIMAL STEPSIZES.

Remark: The normalising factor in the stepsize depends on the sum of squares in LMS in (16), whereas it is the square of the sum in WSLMS in (21). These two differing normalising factors converges to the same value for independent and identically distributed (i.i.d.) samples.

3) Optimal Stepsize for PWS: Unlike WSLMS and LMS, PWS requires two optimal stepsizes for its updates in (6) and (7). Based on Section III-C1, the optimal stepsize for (6) can be obtained as

$$\mu_{\text{Partial1}} \quad = \quad \frac{1}{\sum_{\ell=1}^{K}x^2(n-\ell+1)} \tag{22}$$

The optimal stepsize for (7) can be similarly derived as

$$\mu_{\text{Partial2}} \quad = \quad \frac{1}{(L-K)\left(\sum_{\kappa=K+1}^{L}x(n-\kappa+1)\right)^2} \tag{23}$$

Table II summarises all stepsizes in terms of that of the LMS algorithm for optimal performances of the adaptive algorithms for i.i.d samples.

IV. Simulations

The aim of the simulations is twofold. First, the robustness of the adaptive algorithms against noise was investigated. Second, the adaptive capability of the proposed algorithms to track time-varying systems/signals in noise-free settings. The filter length of $L=50$ and the number of non-shared coefficients of $K=25$ in PWS were set in all experiments. The validity of the optimal sizes in Table II was verified in Table III.

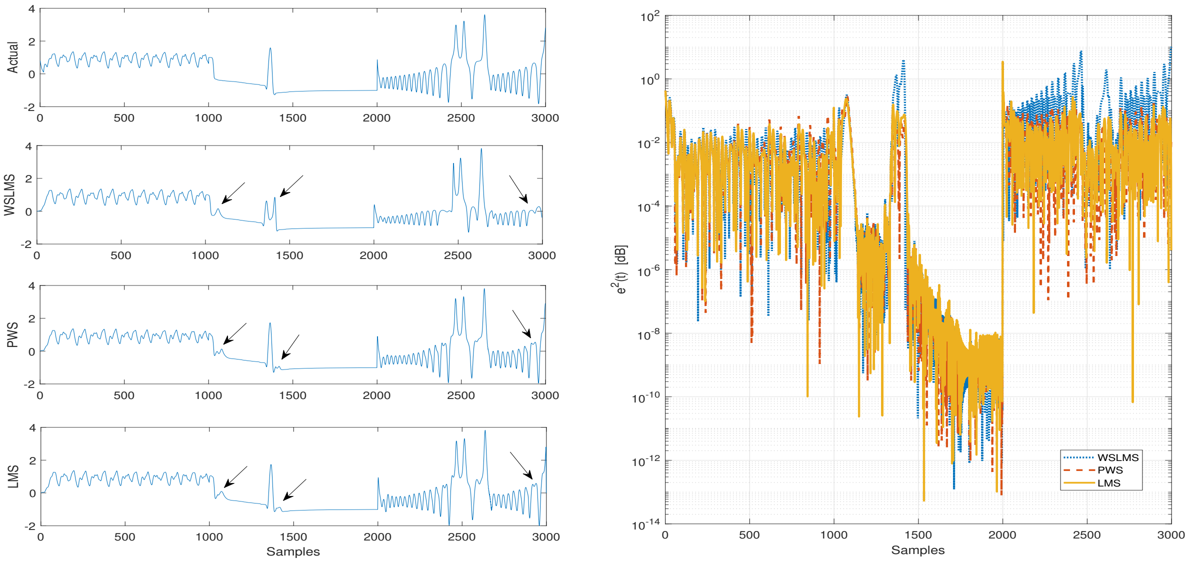

Fig. 3. Transient performances of algorithms considered in a one-step ahead prediction of three chaotic signals. The left panels show the estimated signals, with artifacts highlighted by arrows. The right panel shows the learning curves of WSLMS, PWS, and LMS.

A. Simulation 1: Robustness against noise in signal prediction

Additive white Gaussian noise was added to the Mackey-Glass signal [8] to vary the signal-to-noise ratio (SNR) from -20 dB to 20 dB. For each SNR value, 100 trials were run and averaged to yield the performance curves in mean-squared-error (MSE) sense, as shown in Fig. 1. Observe that WSLMS exhibited the lowest MSE performance consistently. On the other hand, PWS was superior to LMS only at very low SNRs, but performed similarly to LMS. K (the number of non-shared coefficients) was reduced to zero, its performance would approach that of WSLMS.

B. Simulation 2: Modelling more complex systems or signals

Sim 2a on time varying system identification: The finite impulse response $\mathbf{h}(n)$ of the system was changed every 2000 samples as follows: $\mathbf{h}_1 = [0.9, 0.3, 0.5, -0.1]$, $\mathbf{h}_2 = [0.4, 0.1, 0.2, -0.6]$, and $\mathbf{h}_3 = [0.7, 0.4, 0.3, 0.1]$. The synthetic input was generated as $x(n) = 0.8x(n-1) + \omega(n)$, where $\omega(n)$ a unit variance white Gaussian noise. Fig. 2 shows that PWS which outperformed both LMS and WSLMS. This was confirmed by its lowest MSE of -3.52 dB compared to -2.12 dB (LMS) and 3.61 dB (WSLMS).

Sim 2b on time-varying signal prediction: To assess the ability of the considered algorithms to track a time-varying signal, we used a synthetic signal made up of 3 different signals: Mackey-Glass [8], Saito [9] , Lorenz [10] - each of length 1000 samples as shown in top left panel of Fig. 3. The learning curves are shown in the right panel of Fig. 3. Observe that PWS and LMS performed similarly and outperformed WSLMS. The artifacts of the estimate by WSLMS were more pronounced than those of PWS and LMS in the left panels.

Stepsizes	μ_{LMS}	μ_{WSLMS}	μ_{Partial1}	μ_{Partial2}
Sim 1 ($\times 10^{-3}$)	5	0.2 (0.1)	6 (10)	0.3 (0.4)
Sim 2a ($\times 10^{-3}$)	1	0.04 (0.02)	2 (2)	0.08 (0.08)
Sim 2b ($\times 10^{-2}$)	1	0.02 (0.02)	1 (2)	0.04 (0.08)

TABLE III
COMPARISON BETWEEN EXPERIMENTAL AND THEORETICAL OPTIMAL STEPSIZES. THE THEORETICAL VALUES IN $(\cdot)$ WERE DERIVED FROM TABLE II.

C. On the validation of the optimal stepsizes

Table III compares the experimental values with the theoretical values. Observe that the theoretical values for μ_{WSLMS} are more accurate than those of μ_{Partial1} and μ_{Partial2}. This is because the two updates in (6) and (7) of PWS affect each other, although their optimal stepsizes were derived as if they were independent of each other. Yet, these theoretical stepsizes provide useful indicative values to initialise the stepsizes, given they were derived based on an i.i.d assumption.

V. CONCLUSION

We have proposed two novel adaptive algorithms to introduce weight sharing across time rather than across channels into the LMS family of algorithms [6]. In particular, it was shown that weight sharing provides an additional degree of robustness against noise in the form of a moving average in (8). On the other hand, weight sharing constrains the modelling capability. To leverage the advantages of both weight sharing and the standard LMS, the partial weight sharing algorithm has been introduced and shown to offer a good trade-off to implement weight sharing in adaptive filtering algorithms.

References

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[2] F. G. A. Neto and V. H. Nascimento, "A novel reduced-complexity widely linear QLMS algorithm," in *IEEE Statistical Signal Processing Workshop*, pp. 81–84, 2011.

[3] M. Godavarti and A. Hero, "Partial update LMS algorithms," *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2382–2399, 2005.

[4] D. Berberidis, V. Kekatos, and G. B. Giannakis, "Online censoring for large-scale regressions with application to streaming big data," *arXiv:1507.07536*, 2015.

[5] E. C. Mengüç, M. Xiang, and D. P. Mandic, "Online censoring based complex-valued adaptive filters," *Signal Processing*, vol. 200, p. 108638, 2022.

[6] C. Cheong Took and D. Mandic, "Weight sharing for LMS algorithms: Convolutional neural networks inspired multichannel adaptive filtering," *Digital Signal Processing*, vol. 127, p. 103580, 2022.

[7] S. Haykin, *Adaptive Filter Theory*. Prentice Hall (5th Edition), 2014.

[8] M. Mackey and L. Glass, "Oscillation and chaos in physiological control systems," *Science*, vol. 197, pp. 287–289, 1977.

[9] K. Mitsubori and T. Saito, "Torus doubling and hyperchaos in a five dimensional hysteresis circuit," *IEEE Int. Symposium on circuit and systems*, vol. 6, pp. 113–116, 1994.

[10] L. E. Norton, "Deterministic nonperiodic flow," *Journal of the Atmospheric Sciences*, vol. 20, pp. 130–141, 1963.