Connections Between Optimization in Machine Learning and Adaptive Control

Joseph E. Gaudio [1] **Travis E. Gibson** [2] **Anuradha M. Annaswamy** [1] **Michael A. Bolender** [3] **Eugene Lavretsky** [4]

Abstract

This paper explores many immediate connections between adaptive control and machine learning, both through common update laws as well as common concepts. Adaptive control as a field has focused on mathematical rigor and guaranteed convergence. The rapid advances in machine learning on the other hand have brought about a plethora of new techniques and problems for learning. This paper elucidates many of the numerous common connections between both fields such that results from both may be leveraged together to solve new problems. In particular, a specific problem related to higher order learning is solved through insights obtained from these intersections. This version is an extended abstract; refer to (Gaudio et al., 2019b) for the full paper.

1. Introduction

The fields of adaptive control and machine learning have evolved in parallel over the past few decades, with a significant overlap in goals, problem statements, and tools. Machine learning as a field has focused on computer based systems that improve through experience (Duda et al., 2001; Bishop, 2006; Hastie et al., 2009; Efron & Hastie, 2016; Goodfellow et al., 2016; Jordan & Mitchell, 2015). Often times the process of learning is encapsulated in the form of a parameterized model, whose parameters are learned in order to approximate a function. Optimization methods are commonly employed to reduce the function approximation error using any and all available data. The field of adaptive control, on the other hand, has focused on the process of controlling engineering systems in order to accomplish regulation and tracking of critical variables of interest (e.g. position and force in robotics, Mach number and altitude in aerospace systems, frequency and voltage in power systems) in the presence of uncertainties in the underlying system models, changes in the environment, and unforeseen variations in the overall infrastructure (Narendra & Annaswamy, 1989; Sastry & Bodson, 1989; Åström & Wittenmark, 1995; Ioannou & Sun, 1996; Narendra & Annaswamy, 2005). The approach used for accomplishing such regulation and tracking in adaptive control is the learning of underlying parameters through an online estimation algorithm. Stability theory is employed for enabling guarantees for the safe evolution of the critical variables, and convergence of the regulation and tracking errors to zero.

Learning parameters of a model in both machine learning and adaptive control occurs through the use of input-output data. In both cases, the main algorithm used for updating the parameters is often based on a gradient descent-like algorithm. Related tools of analysis, convergence, and robustness in both fields have a tremendous amount of similarity. As the scope of problems in both fields increases, the associated complexity and challenges increase as well. Therefore it is highly attractive to understand these similarities and connections so that the two communities can develop new methods for addressing new challenges.

2. Connections: Update Law

Two types of error models are common in machine learning and adaptive control, where output errors e_y may be related to regressors (features) ϕ and parameter errors $\tilde{\theta}$ as

$$e_y(t) = \tilde{\theta}^T(t)\phi(t) \tag{1}$$

$$e_y(t) = W(s)[\tilde{\theta}^T(t)\phi(t)] \tag{2}$$

where $W(s)$ denotes a dynamic operator and $\tilde{\theta} = \theta - \theta^*$ (θ^* unknown). Our goal with both perspectives will be to adjust a parameter θ with knowledge of the regressor ϕ and output error e_y, such that a loss function $L(\theta; e_y)$ is minimized. For the adaptive control perspective we present solutions in terms of gradient flow in continuous time t while the machine learning updates are presented as gradient descent in discrete time steps indexed by k, i.e.,

$$\dot{\theta}(t) = -\gamma \nabla_\theta L(\theta(t)) \tag{3}$$

$$\theta_{k+1} = \theta_k - \gamma_k \nabla_\theta L(\theta_k) \tag{4}$$

where $\gamma > 0$ is the learning rate in gradient flow and γ_k is the step size in gradient descent. For a more detailed discussion of the problem statements see Appendix A. In this

[1]Massachusetts Institute of Technology [2]Harvard Medical School [3]Air Force Research Laboratory [4]The Boeing Company. Correspondence to: Joseph E. Gaudio <jegaudio@mit.edu>.

Proceedings of the 1st Adaptive & Multitask Learning Workshop, Long Beach, California, 2019. Copyright 2019 by the author(s).

section we consider the question: What common modifications to the update laws in (3) and (4) have been developed?

2.1. σ-Modification, e-Modification, Regularization

While the update laws in (3) and (4) are designed primarily to reduce the output error e_y, there are several secondary reasons to modify these update laws from robustness considerations due to perturbations stemming from disturbances, noise, and other unmodeled causes. Historically the adaptive update law in (3) has been modified to ensure robustness to bounded disturbances as

$$\dot{\theta}(t) = -\gamma \left[\nabla_\theta L(\theta(t)) + \sigma \mathcal{G}(\theta(t), e_y(t))\right] \tag{5}$$

where $\sigma > 0$ is a tuneable parameter that scales the extra term $\mathcal{G}$. Common choices for $\mathcal{G}$ include the σ-modification $\mathcal{G} = \theta$ (Ioannou & Kokotovic, 1984), and the e-modification $\mathcal{G} = \|e_y\|\theta$ (Narendra & Annaswamy, 1987b).

Regularization is often included in a machine learning optimization problem in order to help cope with overfitting by including constraints on the parameter, thus resulting in an augmented loss function (Hastie et al., 2009; Bubeck, 2015): $\bar{L}(\theta) = L(\theta) + \sigma\mathcal{R}(\theta)$ where $\sigma > 0$ is a tuneable parameter, often referred to as a Lagrange multiplier. The gradient descent update (4) for this augmented loss function is often referred to as the "regularized follow the leader" algorithm in online learning (Hazan, 2016) and may be expressed as

$$\theta_{k+1} = \theta_k - \gamma_k \left[\nabla_\theta L(\theta_k) + \sigma \nabla_\theta \mathcal{R}(\theta_k)\right]. \tag{6}$$

The common choice of ℓ_2 regularization in machine learning of $\mathcal{R} = (1/2)\|\theta\|_2^2$, can be seen to coincide with the σ-modification (Ioannou & Kokotovic, 1984), as $\nabla_\theta \mathcal{R} = \mathcal{G}$.

2.2. Dead-Zone Modification and Early Stopping

This subsection details common modifications of the update laws in both fields adopted to cease updating the parameter estimate after sufficient tuning. Another method in adaptive control employed to increase robustness in the presence of bounded disturbances is to employ a "dead-zone" (Peterson & Narendra, 1982), for the update in (3):

$$\dot{\theta}(t) = \begin{cases} -\gamma \nabla_\theta L(\theta(t)), & \mathcal{D}(e_y) > d_0 + \epsilon \\ 0, & \mathcal{D}(e_y) \leq d_0 + \epsilon \end{cases} \tag{7}$$

where $d_0 > 0$ is the dead-zone width that may correspond to an upper bound on the disturbance, and $\epsilon > 0$ is a small constant. The function $\mathcal{D}$ is a non-negative metric on the output error to stop adaptation in desired regions of the output space. A common choice is $\mathcal{D} = \|e_y\|$ such that adaptation stops after a small output error is achieved above a noise level with upper bound d_0.

The training processes is often stopped in machine learning applications as a method to deal with overfitting (Hastie et al., 2009). This may be done by using multiple data sets and stopping the parameter update process (4) when the loss computed for a validation data set starts to increase (Prechelt, 1998). Early stopping is often seen to be needed for training neural networks due to their large number of parameters (Goodfellow et al., 2016) and can act as regularization (Sjöberg & Ljung, 1995).

2.3. Projection

It is often desirable to define a compact region a priori for the parameters θ, such that during the learning process the parameters are not allowed to leave that region. In physical systems there are natural constraints which may aid in the design of that region, and for non-physical systems, the constraints are often engineered by the algorithm designer. The continuous projection algorithm, commonly employed in adaptive control for increased robustness to unmodeled dynamics (Kreisselmeier & Narendra, 1982; Lavretsky et al., 2012; Hussain, 2017), is defined as

$$\text{Proj}(\theta_i, \zeta_i) = \begin{cases} \frac{\theta_{i,\max}^2 - \theta_i^2}{\theta_{i,\max}^2 - \theta_{i,\max}'^2}\zeta_i, & \theta_i \in \Omega_i \wedge \theta_i\zeta_i > 0 \\ \zeta_i, & \text{otherwise} \end{cases} \tag{8}$$

where Ω, $\theta_{i,\max}$, $\theta'_{i,\max}$ define a user-specified boundary layer region inside of a compact convex set Θ. The update law in (3) may then be modified as

$$\dot{\theta}(t) = -\gamma \text{Proj}\left[\theta(t), \nabla_\theta L(\theta(t))\right]. \tag{9}$$

The following projection operation commonly used in online learning (Zinkevich, 2003; Hazan et al., 2007; 2008; Hazan, 2016) finds the closest point in a convex set

$$\Pi_\Theta(\bar{\theta}) \triangleq \arg\min_{\theta \in \Theta} \|\theta - \bar{\theta}\| \tag{10}$$

which may be employed in the update law (4) modified as

$$\bar{\theta}_{k+1} = \theta_k - \gamma_k \nabla_\theta L(\theta_k), \qquad \theta_{k+1} = \Pi_\Theta(\bar{\theta}_{k+1}). \tag{11}$$

2.4. Adaptive Gains and Stepsizes

The following parameter update law is one example which alters the gain of the standard update law (3) as a function of the time varying regressors ϕ (Narendra & Annaswamy, 1989; Ioannou & Sun, 1996):

$$\begin{aligned} \dot{\theta}(t) &= -\gamma \Gamma(t) \nabla_\theta L(\theta(t)) \\ \dot{\Gamma}(t) &= \begin{cases} \Upsilon\Gamma(t) - \frac{\Gamma(t)\phi(t)\phi^T(t)\Gamma(t)}{\mathcal{N}(t)}, & \|\Gamma(t)\| \leq \Gamma_0 \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{12}$$

where $\Upsilon \geq 0$ is a *forgetting factor* and $\mathcal{N}(t)$ is a *normalizing signal*, with common choice $\mathcal{N}(t) = (1 + \mu\phi^T(t)\phi(t))$. It can be seen that the update for Γ may be used in the update for θ to result in a gain adaptive to the regressor ϕ.

Adaptive step size methods (Duchi et al., 2011; Zeiler, 2012; Kingma & Ba, 2017; Reddi et al., 2018) have seen widespread use in machine learning problems due to their ability to handle sparse and small gradients by adjusting the step size as a function of features as they are processed online. A common update law for adaptive step size methods (Reddi et al., 2018) can then be seen to be similar to (11) as

$$\bar{\theta}_{k+1} = \theta_k - \gamma_k m_k / V_k^{1/2}, \qquad \theta_{k+1} = \Pi_\Theta(\bar{\theta}_{k+1}) \quad (13)$$

where m_k and V_k are functions of previous gradients, which can be compared to normalization by the regressor in (12).

3. Connections: Tools and Concepts

In this section we consider concepts and tools common to machine learning and adaptive control.

3.1. Lyapunov Functions and Regret

Stability and convergence tools in adaptive control and online machine learning are analyzed in this section. Suppose we consider the error model in (2) where $W(s) = c(sI - A)^{-1}b$, and a corresponding state space representation of the form (Narendra & Annaswamy, 1989)

$$\dot{e}(t) = Ae(t) + b\tilde{\theta}^T(t)\hat{\phi}(t) + b\theta^{*T}\tilde{\phi}(t), \; e_y(t) = ce(t) \quad (14)$$

The term $\tilde{\phi}$ is due to exponentially decaying terms in the regressor ϕ. That is, $\tilde{\phi} = \hat{\phi} - \phi$ and $\dot{\tilde{\phi}} = \Lambda\tilde{\phi}$ for a Hurwitz matrix $\Lambda \in \mathbb{R}^{N\times N}$.[1] Stability is often proven in adaptive control by the use of a Lyapunov function V, such as

$$V = \gamma^{-1}\tilde{\theta}^T\tilde{\theta} + e^T P e + \alpha\tilde{\phi}^T\bar{P}\tilde{\phi}. \quad (15)$$

Note that the last two terms in V are not needed for the algebraic error model in (1). The time derivative of the Lyapunov function may then be stated using the update law in (3) and the KYP lemma as $\dot{V} = -e^T Q e - \alpha\tilde{\phi}^T\bar{Q}\tilde{\phi} + 2e^T Pb\theta^{*T}\tilde{\phi}$, where $\dot{V} \leq 0$ for $\alpha > (4\|Pb\|^2\|\theta^*\|^2/(\min eig(Q)\cdot\min eig(\bar{Q}))$. It can be shown that $\delta(t) \triangleq 2e^T Pb\theta^{*T}\tilde{\phi}$ is exponentially decaying with $\tilde{\phi}, e \in \mathcal{L}_2 \cap \mathcal{L}_\infty$. By integrating $\dot{V}$ from t_0 to T:

$$\int_{t_0}^{T} e^T Q e dt - \int_{t_0}^{T}\delta(t)dt \leq -\int_{t_0}^{T}\dot{V}dt = V(t_0) - V(T). \quad (16)$$

Given that $\dot{V} \leq 0$, $V(t_0) - V(T) \leq V(t_0) < \infty$.

[1] This formulation is common in the design of non-minimal adaptive observers (Narendra & Annaswamy, 1989). It can be noted that $\hat{\phi} \to \phi$ as $t \to \infty$ as Λ is Hurwitz. Also for $\hat{\phi} = \phi$, (14) is the same as (2). A Hurwitz matrix Λ implies the existence of a positive definite matrix $\bar{P} = \bar{P}^T \in \mathbb{R}^{N\times N}$ and $0 < \bar{Q} = \bar{Q}^T \in \mathbb{R}^{N\times N}$ such that: $\Lambda^T\bar{P} + \bar{P}\Lambda = -\bar{Q}$.

In online learning, efficiency of an algorithm is often analyzed using the notion of "regret" as

$$\text{regret}_T = \sum_{k=1}^{T}\mathcal{C}_k(\theta_k) - \min_{\theta\in\Theta}\sum_{k=1}^{T}\mathcal{C}_k(\theta) \quad (17)$$

where regret can be seen to correspond to the sum of the time varying convex costs $\mathcal{C}_k$ associated with the choice of the time varying parameter estimate θ_k, minus the cost associated with the best static parameter estimate choice, over a time horizon of T steps (Zinkevich, 2003; Hazan et al., 2007; 2008; Hazan, 2016). Suppose we consider a quadratic cost $\mathcal{C}_k = e_k^T Q e_k$, $Q = Q^T > 0$. A continuous time limit of (17) leads to an integral of the form

$$\text{continuous regret}_T = \int_{t_0}^{T} e^T Q e dt - \int_{t_0}^{T}\bar{\delta}(t)dt \quad (18)$$

where $\bar{\delta}(t)$ is an exponentially decaying signal which is due to nonzero initial conditions in (2) or similarly in (14). A strong similarity can thus be seen between (16) and (18).

It is desired to have regret grow sub-linearly with time, such that average regret, $(1/T)\text{regret}_T$, goes to zero in the limit $T \to \infty$, to provide for an efficient algorithm (Hazan, 2016). For adaptive control, convergence of state/output errors is shown from a similar integral which is akin to *constant* regret upper bounded by $V(t_0)$ in (16).

3.2. Unmodeled Dynamics and Generalization

Models used to design adaptive controllers, including the examples of (1) and (2), are approximations with a certain amount of modeling errors. As such, they may only hold about an operating point and need to contend with unmodeled dynamics. This implies that any stabilizing controller must be designed to not only adapt to parametric uncertainties, but also be robust to unmodeled dynamics. In addition, constraints on the state and input may also be present in adaptive control problems (Karason & Annaswamy, 1994; Annaswamy & Kárason, 1995). Analysis becomes more complicated when considering unmodeled dynamics and constraints, resulting in non-global guarantees. Many of the update law modifications in adaptive control from Section 2 were initially derived to ensure robustness in such cases.

This same notion of robustness to modeling errors exists in machine learning in which an estimator is constructed from a finite training data set. It is then desired that this estimator produces a low prediction error based on a test data set consisting of unseen data. Generalization thus refers to the concept of a designed estimator having low loss when applied to new problems. In particular it can be seen that in specific cases, generalization pertains to stability, where algorithms that are stable and train in a small amount of time result in a small generalization error (Bousquet & Elisseeff, 2002; Hardt et al., 2016).

3.3. Persistent Excitation and Stochastic Perturbations

Persistent excitation (PE) of the system regressor in adaptive control is a condition that has been shown to be necessary and sufficient for parameter convergence (Jenkins et al., 2018). It can be shown that if the regressor ϕ is persistently exciting, then the parameter estimation error $\tilde{\theta}(t)$ converges to zero uniformly in time (Narendra & Annaswamy, 1989). The PE condition essentially corresponds to certain spectral conditions being satisfied by the regressor (Boyd & Sastry, 1983; 1986). A detailed exposition of system identification and parameter convergence in both deterministic and stochastic cases can be found in (Goodwin & Sin, 1984; Anderson & Johnson, 1982; Narendra & Annaswamy, 1986; 1987a; Ljung, 1987). Another way to think of the PE condition is that it leads to a perfect test error, since it provides for convergence of the parameter error to zero, and therefore zero output error once transients decay to zero.

Many machine learning problems consider the case when stochastic perturbations are present. In this context, significant improvements may be possible by leveraging well known concepts in system identification (Ljung, 1987). For example (Dean et al., 2018) purposely includes a Gaussian random input into a dynamical system in order to provide for PE by construction. Such stochastic perturbations can guarantee a PE condition only in the limit, when infinite samples can be obtained. In order to address the realistic case of finite samples, approaches in machine learning algorithms for system identification and control have attempted to obtain performance bounds with probability $1 - p_f$ for $p_f \in (0, 1)$, where the bound usually scales inversely with p_f. The probability of failure given by the choice of p_f allows for error due to the presence of finite samples.

3.4. Neural Networks

Gradient based methods to solve for estimates of unknown parameters via back propagation, in what would develop into the foundations of neural networks have been used for decades in control, with early examples consisting of finding optimal trajectories (Pontryagin, 1961) in flight control (Kelley, 1960), and resource allocation problems (Bryson, 1961) (see (Dreyfus, 1990) for a brief history). Since then, the use of neural networks in control systems has expanded to include stabilizing nonlinear dynamical systems (Miller et al., 1995). Design and analysis of stable controllers based on neural networks was taken up by the adaptive control community due to the the similarities of gradient-like update laws used in neural networks and adaptive control. The adaptive control community developed a well established literature for the use of neural networks in nonlinear dynamical systems in the 1990s (Miller et al., 1995; Narendra & Parthasarathy, 1990; 1991; Yu & Annaswamy, 1996; 1998).

The use of neural networks in the machine learning community greatly expanded as of recent due to the increase in computing power available and an increase in applications (Krizhevsky et al., 2012; Sutskever et al., 2013; Goodfellow et al., 2016). Recurrent neural networks (Hopfield, 1982; Hinton & Sejnowski, 1983; Hochreiter & Schmidhuber, 1997), while often similar in structure to nonlinear dynamical systems, have historically been trained in a manner similar to feed-forward neural networks (Rumelhart et al., 1986) using back propagation through time (Werbos, 1990). While a theoretical understanding of why deep neural networks work as well as they do for given problems has been lacking, the machine learning community has worked to rigorously analyze sub-classes of deep neural network architectures such as deep linear networks (Arora et al., 2018; 2019). The update laws employed in training deep neural networks often include selections of modifications of the update laws as discussed in Section 2 (Schmidhuber, 2015).

4. Advantageous Combinations of Tools: Higher Order Learning

Given the many similarities in problem statements, tools, concepts, and algorithms, we now demonstrate how methods from the field of adaptive control can be used to solve a new problem related to higher-order learning. Many of the update laws addressed thus far were first-order in nature, and made use of gradient-like quantities for learning. A question of increasing interest is when accelerated learning can occur for higher-order learning methods. In particular, Nesterov's accelerated method (Nesterov, 1983) was able to certify a convergence rate of $O(1/k^2)$ as compared to the standard gradient descent (4) rate of $O(1/k)$ for a class of convex functions. A parameterization of Nesterov's higher order method may be stated as

$$\theta_{k+1} = \vartheta_k - \gamma \nabla_\theta L(\vartheta_k), \quad \vartheta_k = \theta_k + \beta(\theta_k - \theta_{k-1}) \tag{19}$$

where $\beta > 0$ is a design parameter that weighs the effect of past parameters. Continuous time problem formulations have been explored in (Su et al., 2016; Wibisono et al., 2016), with rate-matching discretizations established in (Wilson et al., 2016; Wilson, 2018; Betancourt et al., 2018). Many of these methods however become inadequate for time varying inputs and features.

Adaptive update laws which include additional levels of integration appeared in the "higher order tuners" in (Morse, 1992; Evesque et al., 2003), and take the form

$$\dot{\vartheta}(t) = -\gamma \nabla_\theta L(\theta(t)), \quad \dot{\theta}(t) = -\beta(\theta(t) - \vartheta(t))\mathcal{N}_t \tag{20}$$

where $\mathcal{N}_t \triangleq (1 + \mu \phi^T(t)\phi(t))$ for a $\mu > 0$. In contrast to (19), the update law in (20) can be shown to be stable in the presence of time varying regressors as in (1) and as well as in adaptive control applications with error model as in (2) (Gaudio et al., 2019a). This solution was only possible by leveraging techniques from the field of adaptive control.

Acknowledgments

The authors acknowledge Dr. Michael I. Jordan for useful discussions. This work was supported by the Air Force Research Laboratory, Collaborative Research and Development for Innovative Aerospace Leadership (CRDInAL), Thrust 3 - Control Automation and Mechanization grant FA 8650-16-C-2642 and the Boeing Strategic University Initiative.

References

Anderson, B. D. and Johnson, C. Exponential convergence of adaptive identification and control algorithms. *Automatica*, 18(1):1–13, jan 1982. doi: 10.1016/0005-1098(82)90021-8.

Annaswamy, A. M. and Kárason, S. P. Discrete-time adaptive control in the presence of input constraints. *Automatica*, 31(10):1421–1431, oct 1995. doi: 10.1016/0005-1098(95)00059-6.

Arora, S., Cohen, N., and Hazan, E. On the optimization of deep networks: Implicit acceleration by overparameterization. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 244–253, Stockholmsmssan, Stockholm Sweden, July 2018. PMLR.

Arora, S., Cohen, N., Golowich, N., and Hu, W. A convergence analysis of gradient descent for deep linear neural networks. In *International Conference on Learning Representations*, 2019.

Åström, K. J. and Wittenmark, B. *Adaptive Control: Second Edition*. Addison-Wesley Publishing Company, 1995.

Betancourt, M., Jordan, M. I., and Wilson, A. C. On symplectic optimization. *arXiv preprint arXiv:1802.03653*, 2018.

Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, feb 2017. doi: 10.1080/01621459.2017.1285773.

Bousquet, O. and Elisseeff, A. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, March 2002.

Boyd, S. and Sastry, S. On parameter convergence in adaptive control. *Systems & Control Letters*, 3(6):311–319, dec 1983. doi: 10.1016/0167-6911(83)90071-3.

Boyd, S. and Sastry, S. S. Necessary and sufficient conditions for parameter convergence in adaptive control. *Automatica*, 22(6):629–639, nov 1986. doi: 10.1016/0005-1098(86)90002-6.

Bryson, A. E. A gradient method for optimizing multistage allocation processes. In *Proc. Harvard Univ. Symposium on digital computers and their applications*, 1961.

Bubeck, S. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8 (3-4):231–357, 2015. doi: 10.1561/2200000050.

Dean, S., Mania, H., Matni, N., Recht, B., and Tu, S. Regret bounds for robust adaptive control of the linear quadratic regulator. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 4192–4201. Curran Associates, Inc., 2018.

Dreyfus, S. E. Artificial neural networks, back propagation, and the kelley-bryson gradient procedure. *Journal of Guidance, Control, and Dynamics*, 13(5):926–928, sep 1990. doi: 10.2514/3.25422.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, July 2011.

Duda, R. O., Hart, P. E., and Stork, D. G. *Pattern Classification, 2nd Edition*. John Wiley & Sons, 2001.

Efron, B. and Hastie, T. *Computer Age Statistical Inference: Algorithms, Evidence and Data Science*. Cambridge University Press, 2016.

Evesque, S., Annaswamy, A. M., Niculescu, S., and Dowling, A. P. Adaptive control of a class of time-delay systems. *Journal of Dynamic Systems, Measurement, and Control*, 125(2):186, 2003. doi: 10.1115/1.1567755.

Gaudio, J. E., Gibson, T. E., Annaswamy, A. M., and Bolender, M. A. Provably correct learning algorithms in the presence of time-varying features using a variational perspective. *arXiv preprint arXiv:1903.04666*, 2019a.

Gaudio, J. E., Gibson, T. E., Annaswamy, A. M., Bolender, M. A., and Lavretsky, E. Connections between adaptive control and optimization in machine learning. *arXiv preprint arXiv:1904.05856*, 2019b.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016.

Goodwin, G. C. and Sin, K. S. *Adaptive Filtering Prediction and Control*. Prentice Hall, 1984.

Hardt, M., Recht, B., and Singer, Y. Train faster, generalize better: Stability of stochastic gradient descent. In Balcan, M. F. and Weinberger, K. Q. (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1225–1234, New York, New York, USA, June 2016. PMLR.

Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

Hazan, E. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016. doi: 10.1561/2400000013.

Hazan, E., Agarwal, A., and Kale, S. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, aug 2007. doi: 10.1007/s10994-007-5016-8.

Hazan, E., Rakhlin, A., and Bartlett, P. L. Adaptive online gradient descent. In Platt, J. C., Koller, D., Singer, Y., and Roweis, S. T. (eds.), *Advances in Neural Information Processing Systems 20*, pp. 65–72. Curran Associates, Inc., 2008.

Hinton, G. E. and Sejnowski, T. J. Optimal perceptual inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 448–453, Washington, DC, June 1983.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Computation*, 9(8):1735–1780, nov 1997. doi: 10.1162/neco.1997.9.8.1735.

Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, apr 1982. doi: 10.1073/pnas.79.8.2554.

Hussain, H. S. *Robust Adaptive Control in the Presence of Unmodeled Dynamics*. PhD thesis, MIT, 2017.

Ioannou, P. A. and Kokotovic, P. V. Robust redesign of adaptive control. *IEEE Transactions on Automatic Control*, 29(3):202–211, mar 1984. doi: 10.1109/TAC.1984.1103490.

Ioannou, P. A. and Sun, J. *Robust Adaptive Control*. PTR Prentice-Hall, 1996.

Jenkins, B. M., Annaswamy, A. M., Lavretsky, E., and Gibson, T. E. Convergence properties of adaptive systems and the definition of exponential stability. *SIAM Journal on Control and Optimization*, 56(4):2463–2484, jan 2018. doi: 10.1137/15M1047805.

Jordan, M. I. and Mitchell, T. M. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, jul 2015. ISSN 0036-8075. doi: 10.1126/science.aaa8415.

Karason, S. P. and Annaswamy, A. M. Adaptive control in the presence of input constraints. *IEEE Transactions on Automatic Control*, 39(11):2325–2330, 1994. doi: 10.1109/9.333787.

Kelley, H. J. Gradient theory of optimal flight paths. *ARS Journal*, 30(10):947–954, oct 1960. doi: 10.2514/8.5282.

Kingma, D. P. and Ba, J. L. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2017.

Kreisselmeier, G. and Narendra, K. S. Stable model reference adaptive control in the presence of bounded disturbances. *IEEE Transactions on Automatic Control*, 27(6):1169–1175, dec 1982. doi: 10.1109/TAC.1982.1103093.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 25*, pp. 1097–1105. Curran Associates, Inc., 2012.

Lavretsky, E., Gibson, T. E., and Annaswamy, A. M. Projection operator in adaptive systems. *arXiv preprint arXiv:1112.4232*, 2012.

Ljung, L. *System Identification: Theory for the User*. Prentice-Hall, 1987.

Miller, W. T., Sutton, R. S., and Werbos, P. J. *Neural Networks for Control*. MIT press, 1995.

Morse, A. S. High-order parameter tuners for the adaptive control of linear and nonlinear systems. In *Systems, Models and Feedback: Theory and Applications*, pp. 339–364. Birkhuser Boston, 1992. doi: 10.1007/978-1-4757-2204-8_23.

Narendra, K. S. and Annaswamy, A. M. Robust adaptive control in the presence of bounded disturbances. *IEEE Transactions on Automatic Control*, 31(4):306–315, apr 1986. doi: 10.1109/TAC.1986.1104259.

Narendra, K. S. and Annaswamy, A. M. Persistent excitation in adaptive systems. *International Journal of Control*, 45(1):127–160, jan 1987a. doi: 10.1080/00207178708933715.

Narendra, K. S. and Annaswamy, A. M. A new adaptive law for robust adaptation without persistent excitation. *IEEE Transactions on Automatic Control*, 32(2):134–145, feb 1987b. doi: 10.1109/TAC.1987.1104543.

Narendra, K. S. and Annaswamy, A. M. *Stable Adaptive Systems*. Prentice-Hall, Inc., NJ, 1989. (out of print).

Narendra, K. S. and Annaswamy, A. M. *Stable Adaptive Systems*. Dover, 2005.

Narendra, K. S. and Parthasarathy, K. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1(1):4–27, mar 1990. doi: 10.1109/72.80202.

Narendra, K. S. and Parthasarathy, K. Gradient methods for the optimization of dynamical systems containing neural networks. *IEEE Transactions on Neural Networks*, 2(2): 252–262, mar 1991. doi: 10.1109/72.80336.

Nesterov, Y. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27:372–376, 1983.

Parks, P. C. Liapunov redesign of model reference adaptive control systems. *IEEE Transactions on Automatic Control*, 11(3):362–367, jul 1966. doi: 10.1109/TAC.1966.1098361.

Peterson, B. B. and Narendra, K. S. Bounded error adaptive control. *IEEE Transactions on Automatic Control*, 27(6): 1161–1168, dec 1982. doi: 10.1109/TAC.1982.1103112.

Pontryagin, L. *Mathematical Theory of Optimal Processes*. Routledge, may 1961. doi: 10.1201/9780203749319.

Prechelt, L. Automatic early stopping using cross validation: quantifying the criteria. *Neural Networks*, 11(4):761–767, jun 1998. doi: 10.1016/S0893-6080(98)00010-0.

Reddi, S. J., Kale, S., and Kumar, S. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, oct 1986. doi: 10.1038/323533a0.

Sastry, S. and Bodson, M. *Adaptive Control: Stability, Convergence and Robustness*. Prentice-Hall, 1989.

Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, jan 2015. doi: 10.1016/j.neunet.2014.09.003.

Sjöberg, J. and Ljung, L. Overtraining, regularization and searching for a minimum, with application to neural networks. *International Journal of Control*, 62(6):1391–1407, dec 1995. doi: 10.1080/00207179508921605.

Su, W., Boyd, S., and Candès, E. J. A differential equation for modeling nesterov's accelerated gradient method: Theory and insights. *Journal of Machine Learning Research*, 17(153):1–43, 2016.

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. On the importance of initialization and momentum in deep learning. In Dasgupta, S. and McAllester, D. (eds.), *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 1139–1147. PMLR, 2013.

Vapnik, V. Principles of risk minimization for learning theory. In Moody, J. E., Hanson, S. J., and Lippmann, R. P. (eds.), *Advances in Neural Information Processing Systems 4*, pp. 831–838. Morgan-Kaufmann, 1992.

Werbos, P. J. Backpropagation through time: What it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990. doi: 10.1109/5.58337.

Wibisono, A., Wilson, A. C., and Jordan, M. I. A variational perspective on accelerated methods in optimization. *Proceedings of the National Academy of Sciences*, 113(47):E7351–E7358, nov 2016. doi: 10.1073/pnas.1614734113.

Wilson, A. *Lyapunov Arguments in Optimization*. PhD thesis, University of California, Berkeley, 2018.

Wilson, A. C., Recht, B., and Jordan, M. I. A lyapunov analysis of momentum methods in optimization. *arXiv preprint arXiv:1611.02635*, 2016.

Yu, S.-H. and Annaswamy, A. M. Neural control for nonlinear dynamic systems. In Touretzky, D. S., Mozer, M. C., and Hasselmo, M. E. (eds.), *Advances in Neural Information Processing Systems 8*, pp. 1010–1016. MIT Press, 1996.

Yu, S.-H. and Annaswamy, A. M. Stable neural controllers for nonlinear dynamic systems. *Automatica*, 34(5):641–650, may 1998. doi: 10.1016/S0005-1098(98)00012-0.

Zeiler, M. D. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

Zinkevich, M. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 928–936, 2003.

Supplementary Material: Appendix

A. Problem Statements

In this section, we state typical problems that are addressed in the areas of adaptive control and machine learning. In both cases, we illustrate the role of learning, the input-output data used, and the overall problem that is desired to be solved.

A.1. Adaptive Control

The main goal in adaptive control is to carry out problems such as estimation or tracking in the presence of parametric uncertainties. The underlying model that relates inputs, outputs, and the unknown parameters is assumed to stem from either the underlying physics or from data-driven approaches. Often these models take the form

$$y(t) = f_1(\phi(t), \theta^*) \tag{21}$$

or

$$\dot{x}(t) = f_2(x(t), u(t), \theta^*),\ y(t) = f_3(x(t), u(t), \theta^*) \tag{22}$$

where $u \in \mathbb{R}^m$ is an exogenous input, $x \in \mathbb{R}^n$ denotes the state, $y \in \mathbb{R}^p$ corresponds to output measurements, $\phi \in \mathbb{R}^N$ corresponds to measured and computed variables, and $\theta^* \in \mathbb{R}^N$ denotes the uncertain parameter. In an estimation problem, the goal is to estimate the state x in (22) and output y in both (21), (22), alongside the unknown parameter θ^* simultaneously, using all available variables. In a control problem, the goal is to determine a control input u so that the output y in (22) follows a desired output $\hat{y}$.

A typical approach taken in order to solve the estimation problem in (21) is to choose an estimator structure of the form

$$\hat{y}(t) = f(\phi(t), \theta(t)) \tag{23}$$

where $\theta \in \mathbb{R}^N$ denotes the estimate of θ^* and adjust θ so that the estimation error $e_y = \hat{y} - y$ is minimized, i.e., choose a function $g_1(e_y, \phi)$ with

$$\dot{\theta}(t) = g_1(e_y(t), \phi(t)) \tag{24}$$

so that the estimator has bounded signals, $e_y(t)$ converges to zero and $\theta(t)$ converges to θ^*. Similarly, the control problem consists of constructing an output tracking error $e_y = \hat{y} - y$, where $\hat{y}$ denotes the desired output that y is required to track. The goal is to then choose functions $g_2(e_y, \phi, \theta)$ and $g_3(e_y, \phi, \theta)$ so that the control input u and parameter estimate θ can be chosen as

$$\begin{aligned} u(t) &= g_2(e_y(t), \phi(t), \theta(t)) \\ \dot{\theta}(t) &= g_3(e_y(t), \phi(t), \theta(t)) \end{aligned} \tag{25}$$

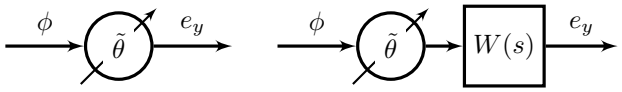

Figure 1. Error Models. **Left:** Regression (26), **Right:** Adaptive Control (27).

leading to closed-loop signals remaining bounded, $e_y(t)$ converging to zero and $\theta(t)$ converging to its true value θ^*. Denote the corresponding parameter errors as $\tilde{\theta} = \theta - \theta^*$.

In order to derive the function g_1 for the estimation problem in (21) and the functions g_2 and g_3 for the control problem in (22) so as to realize the underlying goals, a stability framework together with an error model approach is often employed in adaptive control. The error model approach consists of identifying the basic relationship between the two errors that are commonly present in these adaptive systems, which are the estimation (or tracking) error e_y and the parameter error $\tilde{\theta}$. While the estimation error is measurable and correlated with the parameter error, the parameter error is unknown but adjustable through the parameter estimate. In order to determine the update laws g_i, the relationship (error model) that relates these two errors is used as a cue.

Two types of error models frequently occur in adaptive systems, and are presented below (see Figure 1). The first corresponds to the case when the relation in (21) is linear, and the underlying error model is simply of the form (cf. (Narendra & Annaswamy, 2005))

$$e_y(t) = \tilde{\theta}^T(t)\phi(t) \tag{26}$$

and as a result, the function g_1 in (24) can be determined simply using the gradient rule that minimizes $\|e_y\|^2$. The second is of the form (cf. (Narendra & Annaswamy, 2005))

$$e_y(t) = W(s)[\tilde{\theta}^T(t)\phi(t)] \tag{27}$$

where $W(s)[\zeta]$ denotes a dynamic operator operating on $\zeta(t)$. It has been shown in the adaptive control literature (Narendra & Annaswamy, 1989; Sastry & Bodson, 1989; Åström & Wittenmark, 1995; Ioannou & Sun, 1996; Narendra & Annaswamy, 2005) that for specific classes of dynamic operators $W(s)$, a stable, gradient-like rule can be determined for adjusting $\tilde{\theta}$. Most of these results apply uniformly to the case when u and y are scalars or vectors, with the latter introducing additional technicalities. In this paper we consider the case where inputs and outputs are scalars for notational simplicity, and to focus on the core of the learning problem with multi-dimensional regressors ϕ and parameter estimates θ. Often the unknown parameter θ^* is assumed to reside in a compact convex set, which we will denote as Θ.

A.2. Machine Learning

Machine learning is a broad field encompassing a wide variety of learning techniques and problems such as classification and regression. A large portion of machine learning considers supervised learning problems, where regressors ϕ and outputs y are related to one another in an unknown algebraic manner (Duda et al., 2001; Bishop, 2006; Hastie et al., 2009; Efron & Hastie, 2016; Goodfellow et al., 2016; Jordan & Mitchell, 2015). A typical approach taken in order to perform classification or regression is to choose an output estimator $\hat{y}_k$ parameterized with adjustable weights θ_k as

$$\hat{y}_k = f(\phi_k, \theta_k). \tag{28}$$

A common form of the estimator as in (28) is that of neural networks, where the parameters θ_k represent the adjustable weights in the network (Duda et al., 2001; Bishop, 2006; Hastie et al., 2009; Efron & Hastie, 2016; Goodfellow et al., 2016).

Similar to adaptive control, θ_k is often adjusted using the output error $e_{y,k} = \hat{y}_k - y_k$. A loss function $L : \Theta \to \mathbb{R}$ of $e_{y,k}$ is minimized through the adjustable weights. An example loss function for regression is ℓ_p loss (with $p \in \mathbb{N}$, $p > 0$ and even) $L(\theta_k) = (1/p)\|e_{y,k}\|_p^p$. For binary classification ($y_k \in \{-1, 1\}$) common loss functions include hinge loss $L(\theta_k) = \max(0, 1 - y_k\hat{y}_k)$, and logistic loss $L(\theta_k) = \ln(1 + \exp(-y_k\hat{y}_k))$. Additionally, as in empirical risk minimization (ERM) (Vapnik, 1992), the total loss function considered for the purpose of a parameter update may be an average of loss functions over m samples as: $(1/m)\sum_{i=1}^{m} L_i(\theta_k)$. The above descriptions make it clear that the structure of the estimation problem in both adaptive control and machine learning are strikingly similar. In the next section, we examine the nature of the adjustment of θ_k.

A.3. Common Update Laws

As previously stated, the goal in adaptive control is to design a rule to adjust θ in an online continuous manner using knowledge of ϕ and e_y such that e_y tends toward zero. Given that the output errors may be corrupted by noise, an iterative, gradient-like update is usually employed. To do so for the algebraic error model (26), consider the squared loss cost function: $L(\theta(t)) = (1/2)e_y^2(t)$. The gradient of this function with respect to the parameters can be expressed as: $\nabla_\theta L(\theta(t)) = \phi(t)e_y(t)$. The standard gradient flow update law (Narendra & Annaswamy, 1989) may be expressed as follows with user-designed gain parameter $\gamma > 0$ as

$$\dot{\theta}(t) = -\gamma\nabla_\theta L(\theta(t)) = -\gamma\phi(t)e_y(t). \tag{29}$$

For dynamical error models such as (27), a stability approach rather than a gradient based one is taken using Lyapunov methods, which leads to an adaptive law identical to (29) for a class of dynamic systems $W(s)$ that are strictly positive real (Narendra & Annaswamy, 1989; Parks, 1966).

The common update law for supervised machine learning problems, gradient descent[2], is akin to the time varying regression law (29) in discrete time, and of the form

$$\theta_{k+1} = \theta_k - \gamma_k\nabla_\theta L(\theta_k) \tag{30}$$

where the "stepsize" γ_k is usually chosen as a decreasing function of time (Hazan et al., 2007; 2008; Hazan, 2016; Bubeck, 2015; Zinkevich, 2003), a standard feature of stochastic gradient algorithms.

[2]While this is not true of all machine learning as the field is broad, (for example Bayesian methods often use sampling based techniques such as Markov Chain Monte Carlo), even in the world of probabilistic inference, gradient based methods can also be used, cf. variational inference (Blei et al., 2017).