2019 IEEE 58th Conference on Decision and Control (CDC)
Palais des Congrès et des Expositions Nice Acropolis
Nice, France, December 11-13, 2019

Connections Between Adaptive Control and Optimization in Machine Learning

Joseph E. Gaudio, Travis E. Gibson, Anuradha M. Annaswamy, Michael A. Bolender, and Eugene Lavretsky

***Abstract*— This paper demonstrates many immediate connections between adaptive control and optimization methods commonly employed in machine learning. Starting from common output error formulations, similarities in update law modifications are examined. Concepts in stability, performance, and learning, common to both fields are then discussed. Building on the similarities in update laws and common concepts, new intersections and opportunities for improved algorithm analysis are provided. In particular, a specific problem related to higher order learning is solved through insights obtained from these intersections.**

I. 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 [1]–[6]. 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. speed in automotive systems, 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 [7]–[11]. 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 based on a gradient descent-like algorithm [11]. 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.

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.

J. E. Gaudio and A. M. Annaswamy are with the Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139. {jegaudio,aanna}@mit.edu

T. E. Gibson is with the Department of Pathology, Harvard Medical School, Boston, MA 02115. tgibson@mit.edu

M. A. Bolender is with the Air Force Research Laboratory, WPAFB, OH 45433.

E. Lavretsky is with The Boeing Company, Huntington Beach, CA 92647.

II. 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 and the additional structure for $W(s)$ in order for (3) to hold, refer to [12]. In this section we consider the question: What common modifications to the update laws in (3) and (4) have been developed?

A. σ-Modification, e-Modification, and 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.

1) Adaptive Control: Historically the adaptive update law in (3) has been modified to ensure robustness in the presence of 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 tunable parameter that scales the extra term $\mathcal{G}$. Common choices for $\mathcal{G}$ include the σ-modification $\mathcal{G} = \theta$ [13], and the e-modification $\mathcal{G} = \|e_y\|\theta$ [14].

2) Machine Learning: 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 [1]–[5], [15]–[17]: $\bar{L}(\theta) = L(\theta) + \sigma\mathcal{R}(\theta)$ where $\sigma > 0$ is a tunable 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 [17] 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_p regularization in machine learning of $\mathcal{R} = (1/p)\|\theta\|_p^p$ with $p = 2$, (as in ridge regression), coincides with the σ-modification [13], as then $\nabla_\theta\mathcal{R} = \mathcal{G}$.

B. Deadzone Modification and Early Stopping

1) Adaptive Control: Another method employed to increase robustness in the presence of bounded disturbances is to employ a "dead zone" [18], for the update in (3) as

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

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.

2) Machine Learning: The training processes is often stopped in machine learning applications as a method to deal with overfitting [2]–[5], [19], [20]. 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 [19]. Early stopping is often seen to be needed for training neural networks due to their large number of parameters [2]–[5] and can act as regularization [20].

C. 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.

1) Adaptive Control: A continuous projection algorithm is commonly employed to provide for robustness of the adaptive update law in the presence of unmodeled dynamics [21]–[23]. One such implementation is

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

where Ω, $\theta_{i,\max}$, $\theta'_{i,\max}$ define a user-specified boundary layer region inside of Θ, a compact convex set (see [22]). The update law in (3) may then be modified as

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

2) Machine Learning: The following projection operation commonly used in online learning [15]–[17], [24], [25] finds the closet 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

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

D. Adaptive Gains and Stepsizes

1) Adaptive Control: The following parameter update law for the algebraic error model (1) is one example which alters the gain of the standard update law (3) as a function of the time-varying regressors ϕ [7], [10]:

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

where $\Upsilon \geq 0$ is a *forgetting factor* and $\mathcal{N}(t)$ is a *normalizing signal*, with common choice $\mathcal{N}(t) = (1 + \mu\phi^T(t)\phi(t))$ for $\mu > 0$ chosen appropriately (see for example [10] for a discussion of the choice of parameters). It can be seen that the update for Γ may be integrated and used in the update for θ to result in a gain adaptive to the regressor ϕ.

2) Machine Learning: Adaptive step size methods [26]–[29] 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 [29] can then be seen to be similar to (11) as

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

where m_k and V_k are functions of previous gradients, which can be compared to normalization by the regressor in (12) (c.f. discussion of the functions m_k and V_k in [29]).

III. Connections: Tools and Concepts

This section details concepts and tools common to both machine learning and adaptive control.

A. Lyapunov Functions and Regret

Stability and convergence tools in adaptive control and online machine learning are analyzed in this section.

1) Adaptive Control: 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 [7]

$$\begin{aligned} \dot{e}(t) &= Ae(t) + b\tilde{\theta}^T(t)\hat{\phi}(t) + b\theta^{*T}\tilde{\phi}(t) \\ e_y(t) &= ce(t). \end{aligned} \tag{14}$$

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

[1]This formulation is common in the design of non-minimal adaptive observers [7]. 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: $\Lambda^T\bar{P} + \bar{P}\Lambda = -\bar{Q}$.

control by the use of a Lyapunov function V, such as

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

It should be noted 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 [7] as $\dot{V} = -e^T Q e - \alpha\tilde{\phi}^T\bar{Q}\tilde{\phi} + 2e^T P b\theta^{*T}\tilde{\phi}$, where $\dot{V} \leq 0$ for $\alpha > (4\|Pb\|^2\|\theta^*\|^2/(\min eig(Q)\cdot\min eig(\bar{Q}))$. It can be shown [7] that $\delta(t) \triangleq 2e^T P b\theta^{*T}\tilde{\phi}$ is an exponentially decaying signal with $\tilde{\phi}, e \in \mathcal{L}_2 \cap \mathcal{L}_\infty$. By integrating $\dot{V}$ from t_0 to T, we obtain

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

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

2) Machine Learning: In online learning, efficiency of an algorithm is often analyzed using the notion of "regret" as

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

where regret can be seen to correspond to the sum of 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 [15], [17], [24], [25]. 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 as

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

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 [17]. 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).

B. Unmodeled Dynamics and Generalization

1) Adaptive Control: 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 controllers 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 [30], [31]. Analysis becomes more complicated when considering such unmodeled dynamics and constraints, resulting in non-global guarantees. Many of the update law modifications in adaptive control from Section II were initially derived to ensure robustness in such cases.

2) Machine Learning: This same notion of robustness to modeling errors exists in machine learning in which an estimator $\hat{y}$ 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 [32], [33].

C. Persistent Excitation and Stochastic Perturbations

1) Adaptive Control: 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 [34]. 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 [7]. The PE condition essentially corresponds to certain spectral conditions being satisfied by the regressor [35], [36]. A detailed exposition of system identification and parameter convergence in both deterministic and stochastic cases can be found in [37]–[41]. 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/state error once transients decay to zero.

2) Machine Learning: 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 [41]. For example [42] 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 which hold 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.

D. Tracking vs Exploration

The concept of exploration can be viewed as the opposite of tracking, with the former often employed in machine learning while the latter is one of the main control goals.

1) Adaptive Control: The goal of adaptive control is to adjust the parameter θ in such a way to minimize the output error e_y in (1) and (2). It can be seen from the error models in (1) and (2) with the update in (3), that as the output error e_y goes to zero, learning becomes less and less, and that it is possible for a large parameter error to remain even with zero output (or tracking) error. That is, in many adaptive control applications, stability and tracking are successfully accomplished even without parameter convergence.

2) Machine Learning: In many machine learning methods, including reinforcement learning, there exist explicit modifications to update laws to promote exploration of the parameter space. These modifications include restarting trajectories with random initial conditions, adding random perturbations to algorithms, and driving the system towards non-zero error regions [42]–[44]. This preference of exploration and learning over stability is motivated by the desire to find optimal parameters of a system. Stability is not always crucial as models are often trained with offline data on a computer, allowing for many iterations without the financial cost of failure present in physical systems (i.e. a nonzero probability of failure p_f may be acceptable).

E. Convergence Guarantees

1) Adaptive Control: Adaptive control problems are often parameterized in a specific way such that e_y goes to zero asymptotically as in (1) and (2). Parameter convergence is shown to occur in these cases with a persistent excitation condition (see Section III-C.1). The specific parameterizations in the output space ensure that a global minimum of $e_y = 0$ exists and is unique. In the absence of PE, standard adaptive control algorithms converge to one of the many local minima in the parameter space (i.e. $\dot{\tilde{\theta}} \to 0$ but $\tilde{\theta} \neq 0$) [7].

2) Machine Learning: The field of machine learning has rapidly grown in recent years, as demonstrated by well attended conferences such as ICML and NeurIPS, where papers typically focus on empirical performance on large scale problems. A notable exception is a body of work that is emerging which consists of optimization-centric problem formulations, and the examination of the loss landscape, where recent results have shown that in certain classes of problems, local minimums are nearly equivalent to global minimums in terms of performance on test data [45]–[47].

F. Neural Networks

1) Adaptive Control: 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 [48] in flight control [49], and resource allocation problems [50] (see [51] for a brief history). Since then, the use of neural networks in control systems has expanded to include stabilizing nonlinear dynamical systems [52]. Design and analysis of stable controllers based on neural networks was taken up by the adaptive control community in the 1990s due to the the similarities of gradient-like parameter update laws [52]–[56].

2) Machine Learning: 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 [5], [57], [58]. Recurrent neural networks [59]–[61], while often similar in structure to nonlinear dynamical systems, have historically been trained in a manner similar to feed-forward neural networks [62] using back propagation through time [63]. Hebbian learning [64] based approaches have also been considered. The machine learning community has worked to rigorously analyze subclasses of deep neural network architectures such as deep linear networks [65], [66]. The update laws employed in training deep neural networks often include selections of modifications of the update laws as discussed in Section II. For an overview of the history of neural networks see [67].

G. Other Parameterization Schemes

1) Adaptive Control: Adaptive control schemes often consider the case where an unknown parameter occurs linearly with respect to a regressor vector ϕ and may be related to an output error e_y algebraically (1) or dynamically (2). Often times the vector ϕ is a nonlinear function of the state of the system or reference model in order to approximate a more general nonlinear function D as: $D(x) = \theta^{*T}\phi(x)$ [68]. Common parameterizations for unknown nonlinearities include Gaussian radial basis functions [68]. Another class of parameterizations consist of nonlinearly parameterized uncertainty $D(\theta^*, \phi)$ in dynamical systems, for which there exists stabilizing adaptive control methods [69], [70].

2) Machine Learning: Parametric methods are common in machine learning as well, and are useful in many regression and classification based tasks [1]–[5]. However, Bayesian based approaches are also widespread in areas such as topic models [71], clustering [72] and graphical models [73]. Additionally, new results in high dimensional statistics are increasingly being considered in which the model may be of higher dimension than the sample size [74].

IV. Advantageous Combinations of Machine Learning and Adaptive Control Tools

Given the enormous number of similarities in problem statements, tools, concepts, and algorithms, it is natural to examine what the benefits are that accrue by combining insights obtained in these two communities. Two examples of such an exercise is delineated in this section.

A. 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 in the machine learning community is when accelerated learning can occur for higher-order learning methods [57], [58], [75]. In particular, Nesterov's accelerated method [76] 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 accelerated method may be stated as

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

where $\beta > 0$ is a design parameter that weighs the effect of past parameters. Continuous time problem formulations have been explored in [77], [78], with rate-matching discretizations established in [79]–[81]. Many of these methods however become inadequate for time-varying inputs.

Adaptive update laws which include additional levels of integration appeared in the "higher order tuners" in [82], [83], and take the form

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

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), as well as in adaptive control with error model as in (2), and may be derived from a common variational perspective [84].

B. Improved Algorithm Performance Bounds

Regret analysis common in online machine learning (see Section III-A.2) can result in overly conservative bounds for the performance of an algorithm. In particular, in online projected gradient descent (11) for regression (1) with squared output error cost $\mathcal{C} = (1/2)e_y^2$, regret analysis guarantees $\text{regret}_T = O(\sqrt{T})$ (cf. [17]). For the same regret cost function for regression as in (1), one can guarantee $\text{regret}_T = O(1)$ (constant regret), using adaptive control methods, which is the best bound (up to constants).

V. Conclusions

This paper explored many connections between adaptive control and machine learning, 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 was written to elucidate numerous common connections between both fields such that results from both may be leveraged to solve new problems.

VI. Acknowledgements

The authors acknowledge Dr. Michael I. Jordan for several useful discussions.

References

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

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

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

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

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

[6] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, jul 2015.

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

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

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

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

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

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

[13] P. A. Ioannou and P. V. Kokotovic, "Robust redesign of adaptive control," *IEEE Transactions on Automatic Control*, vol. 29, no. 3, pp. 202–211, mar 1984.

[14] K. S. Narendra and A. M. Annaswamy, "A new adaptive law for robust adaptation without persistent excitation," *IEEE Transactions on Automatic Control*, vol. 32, no. 2, pp. 134–145, feb 1987.

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

[16] S. Bubeck, "Convex optimization: Algorithms and complexity," *Foundations and Trends® in Machine Learning*, vol. 8, no. 3-4, pp. 231–357, 2015.

[17] E. Hazan, "Introduction to online convex optimization," *Foundations and Trends® in Optimization*, vol. 2, no. 3-4, pp. 157–325, 2016.

[18] B. B. Peterson and K. S. Narendra, "Bounded error adaptive control," *IEEE Transactions on Automatic Control*, vol. 27, no. 6, pp. 1161–1168, dec 1982.

[19] L. Prechelt, "Automatic early stopping using cross validation: quantifying the criteria," *Neural Networks*, vol. 11, no. 4, pp. 761–767, jun 1998.

[20] J. Sjöberg and L. Ljung, "Overtraining, regularization and searching for a minimum, with application to neural networks," *International Journal of Control*, vol. 62, no. 6, pp. 1391–1407, dec 1995.

[21] G. Kreisselmeier and K. S. Narendra, "Stable model reference adaptive control in the presence of bounded disturbances," *IEEE Transactions on Automatic Control*, vol. 27, no. 6, pp. 1169–1175, dec 1982.

[22] H. S. Hussain, "Robust adaptive control in the presence of unmodeled dynamics," Ph.D. dissertation, MIT, 2017.

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

[24] E. Hazan, A. Agarwal, and S. Kale, "Logarithmic regret algorithms for online convex optimization," *Machine Learning*, vol. 69, no. 2-3, pp. 169–192, aug 2007.

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

[26] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, Jul. 2011.

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

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

[29] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," in *International Conference on Learning Representations*, 2018.

[30] S. P. Karason and A. M. Annaswamy, "Adaptive control in the presence of input constraints," *IEEE Transactions on Automatic Control*, vol. 39, no. 11, pp. 2325–2330, 1994.

[31] A. M. Annaswamy and S. P. Kárason, "Discrete-time adaptive control in the presence of input constraints," *Automatica*, vol. 31, no. 10, pp. 1421–1431, oct 1995.

[32] O. Bousquet and A. Elisseeff, "Stability and generalization," *Journal of Machine Learning Research*, vol. 2, pp. 499–526, Mar. 2002.

[33] M. Hardt, B. Recht, and Y. Singer, "Train faster, generalize better: Stability of stochastic gradient descent," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. New York, New York, USA: PMLR, Jun. 2016, pp. 1225–1234.

[34] B. M. Jenkins, A. M. Annaswamy, E. Lavretsky, and T. E. Gibson, "Convergence properties of adaptive systems and the definition of exponential stability," *SIAM Journal on Control and Optimization*, vol. 56, no. 4, pp. 2463–2484, jan 2018.

[35] S. Boyd and S. Sastry, "On parameter convergence in adaptive control," *Systems & Control Letters*, vol. 3, no. 6, pp. 311–319, dec 1983.

[36] S. Boyd and S. S. Sastry, "Necessary and sufficient conditions for parameter convergence in adaptive control," *Automatica*, vol. 22, no. 6, pp. 629–639, nov 1986.

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

[38] B. D. Anderson and C. Johnson, "Exponential convergence of adaptive identification and control algorithms," *Automatica*, vol. 18, no. 1, pp. 1–13, jan 1982.

[39] K. S. Narendra and A. M. Annaswamy, "Robust adaptive control in the presence of bounded disturbances," *IEEE Transactions on Automatic Control*, vol. 31, no. 4, pp. 306–315, apr 1986.

[40] ——, "Persistent excitation in adaptive systems," *International Journal of Control*, vol. 45, no. 1, pp. 127–160, jan 1987.

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

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

[43] R. S. Sutton, A. G. Barto, and R. J. Williams, "Reinforcement learning is direct adaptive optimal control," *IEEE Control Systems*, vol. 12, no. 2, pp. 19–22, apr 1992.

[44] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 2018.

[45] A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun, "The loss surfaces of multilayer networks," in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, G. Lebanon and S. V. N. Vishwanathan, Eds., vol. 38. PMLR, 2015, pp. 192–204.

[46] R. Ge, J. D. Lee, and T. Ma, "Matrix completion has no spurious local minimum," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 2973–2981.

[47] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht, "Gradient descent only converges to minimizers," in *29th Annual Conference on Learning Theory*, ser. Proceedings of Machine Learning Research, V. Feldman, A. Rakhlin, and O. Shamir, Eds., vol. 49. Columbia University, New York, New York, USA: PMLR, 23–26 Jun 2016, pp. 1246–1257.

[48] L. Pontryagin, *Mathematical Theory of Optimal Processes*. Routledge, may 1961.

[49] H. J. Kelley, "Gradient theory of optimal flight paths," *ARS Journal*, vol. 30, no. 10, pp. 947–954, oct 1960.

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

[51] S. E. Dreyfus, "Artificial neural networks, back propagation, and the kelley-bryson gradient procedure," *Journal of Guidance, Control, and Dynamics*, vol. 13, no. 5, pp. 926–928, sep 1990.

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

[53] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4–27, mar 1990.

[54] ——, "Gradient methods for the optimization of dynamical systems containing neural networks," *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 252–262, mar 1991.

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

[56] ——, "Stable neural controllers for nonlinear dynamic systems," *Automatica*, vol. 34, no. 5, pp. 641–650, may 1998.

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

[58] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, S. Dasgupta and D. McAllester, Eds., vol. 28, no. 3. PMLR, 2013, pp. 1139–1147.

[59] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554–2558, apr 1982.

[60] G. E. Hinton and T. J. Sejnowski, "Optimal perceptual inference," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Washington, DC, Jun. 1983, pp. 448–453.

[61] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, nov 1997.

[62] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, oct 1986.

[63] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[64] D. O. Hebb, *The Organization of Behavior*. Wiley, 1949.

[65] S. Arora, N. Cohen, and E. Hazan, "On the optimization of deep networks: Implicit acceleration by overparameterization," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. Stockholmsmssan, Stockholm Sweden: PMLR, Jul. 2018, pp. 244–253.

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

[67] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, jan 2015.

[68] R. M. Sanner and J.-J. E. Slotine, "Gaussian networks for direct adaptive control," *IEEE Transactions on Neural Networks*, vol. 3, no. 6, pp. 837–863, 1992.

[69] A.-P. Loh, A. M. Annaswamy, and F. P. Skantze, "Adaptation in the presence of a general nonlinear parameterization: An error model approach," *IEEE Transactions on Automatic Control*, vol. 44, no. 9, pp. 1634–1652, 1999.

[70] C. Cao, A. M. Annaswamy, and A. Kojic, "Parameter convergence in nonlinearly parameterized systems," *IEEE Transactions on Automatic Control*, vol. 48, no. 3, pp. 397–412, mar 2003.

[71] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, Jan. 2003.

[72] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Sharing clusters among related groups: Hierarchical dirichlet processes," in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. MIT Press, 2005, pp. 1385–1392.

[73] M. J. Wainwright and M. I. Jordan, *Graphical Models, Exponential Families, and Variational Inference*. Now Publishers, 2007, vol. 1, no. 1–2.

[74] M. J. Wainwright, *High-Dimensional Statistics*. Cambridge University Press, feb 2019.

[75] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, jan 2009.

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

[77] W. Su, S. Boyd, and E. J. Candès, "A differential equation for modeling nesterov's accelerated gradient method: Theory and insights," *Journal of Machine Learning Research*, vol. 17, no. 153, pp. 1–43, 2016.

[78] A. Wibisono, A. C. Wilson, and M. I. Jordan, "A variational perspective on accelerated methods in optimization," *Proceedings of the National Academy of Sciences*, vol. 113, no. 47, pp. E7351–E7358, nov 2016.

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

[80] A. Wilson, "Lyapunov arguments in optimization," Ph.D. dissertation, University of California, Berkeley, 2018.

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

[82] A. S. Morse, "High-order parameter tuners for the adaptive control of linear and nonlinear systems," in *Systems, Models and Feedback: Theory and Applications*. Birkhäuser Boston, 1992, pp. 339–364.

[83] S. Evesque, A. M. Annaswamy, S. Niculescu, and A. P. Dowling, "Adaptive control of a class of time-delay systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 125, no. 2, p. 186, 2003.

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