Deep Empirical Risk Minimization in finance: looking into the future

A. Max Reppen* H. Mete Soner†

September 25, 2022

Abstract

Many modern computational approaches to classical problems in quantitative finance are formulated as empirical loss minimization (ERM), allowing direct applications of classical results from statistical machine learning. These methods, designed to directly construct the optimal feedback representation of hedging or investment decisions, are analyzed in this framework demonstrating their effectiveness as well as their susceptibility to generalization error. Use of classical techniques shows that over-training renders trained investment decisions to become anticipative, and proves overlearning for large hypothesis spaces. On the other hand, non-asymptotic estimates based on Rademacher complexity show the convergence for sufficiently large training sets. These results emphasize the importance of synthetic data generation and the appropriate calibration of complex models to market data. A numerically studied stylized example illustrates these possibilities, including the importance of problem dimension in the degree of overlearning, and the effectiveness of this approach.

Key words: Deep learning, ERM, Overlearning, Dynamic hedging, Bias-variance trade-off.
Mathematics Subject Classification: 91G60, 49N35, 65C05.

1 Introduction

Recent advances in the training of neural networks make high-dimensional numerical studies feasible for decisions under uncertainty, and in particular, classical hedging and pricing problems in quantitative finance. Although simulation-based methods have been widely used in stochastic (optimal) control for several decades [12], only recently [30], [32] combine it with deep neural networks for the offline construction of optimal feedback actions for sequential decision problems. This approach, which we call (*dynamic*) *deep empirical risk minimization* (ERM), assumes that a training set is either readily available or can be simulated through an assumed model, and then an appropriate empirical average over this training data is used to construct a loss function to be minimized over the network parameters. A near-minimizer is the trained network that approximates the optimal investment actions. This technique is remarkably flexible and tractable. It can handle complex realistic dynamics with ease, does not require a Markov structure, and can be completely data-driven when a sufficiently large training set is available.

*Questrom School of Business, Boston University, Boston, MA, 02215, USA, email: amreppen@bu.edu. Reppen was partly supported by the Swiss National Science Foundation grant SNF 181815.

†Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ, 08540, USA, email: soner@princeton.edu. Research of Soner was partially supported by the National Science Foundation grant DMS 2106462.

Readily this method has been adopted in many studies [2, 9, 10, 17, 18, 29, 34, 36, 49] to study various problems in quantitative finance and we discuss them later in this introduction. We also refer the readers to the recent excellent surveys [24, 37, 50] and the references therein for more information. Our goal is to provide both a computational and a theoretical assessment of this promising new methodology. We use the setting of general stochastic control for optimal sequential investments to study them in a unified manner.

As always, the generalization of the trained network is the key property that is of paramount importance. We study this central question in a general setting with a fixed training set to also explore the feasibility of purely data-driven implementations. Our key observation is to carefully formulate the general stochastic optimal control problem by expressing the pathwise cost of a feedback policy as a function of the randomness whose expected value is the performance of this strategy, so that the loss function used in the training is its empirical average. A central structural assumption this approach makes is that, the reward function for all actions can be computed once a trajectory of the randomness is given. This structure which is pervasive in quantitative finance, is further elaborated in Remark 3.3 below. Thus, these recent approaches can be viewed as empirical risk minimization, enabling us to directly apply classical results from statistical machine learning with ease and provide useful estimates and tools for their analysis. The explicit dependence on the randomness also clarifies how overfitting causes non-adaptedness on the training data. To articulate this maybe not so well-known but potentially conducive reformulation with clarity, we avoid technical constructs and assumptions, and focus on emphasizing the fundamental structures and connections.

It is well-known that optimal feedback controls or investment strategies are determined by the conditional expectations of the value function evaluated at future controlled random states [25], and the above method essentially uses a regression estimate of these expectations. It is therefore natural that it implicitly faces the classical bias-variance trade-off as articulated in [27, 38, 42]. Indeed, optimal decisions depend on a time-varying estimate of the randomness driving the dynamics of the state, which is basically the return process in finance, and in some applications the available training data is limited in size. Hence, as opposed to interesting recent studies [11, 55] arguing the benefits of more complex networks and interpolation, in this dynamic setting, overfitting causes the loss of the most salient restriction of the problem, namely, adaptedness of decisions to the information flow.

The above numerical procedure that we outline in Section 5, uses a hypothesis space $\mathcal{N}_k$ of feedback actions and minimizes the empirical loss function in this space. Therefore, the training is among adapted processes, as feedback controls use only the current information. Still, the global minima of the empirical loss functions are achieved by actions whose coefficients depend on the whole random path including the future, as proved in Theorem 7.1. Therefore, as shown in Section 7, sufficiently large networks may at every time *overlearn* the future randomness instead of estimating them, generating output investment decisions that anticipate the future. This renders the trained feedback actions *on the training set* to be *non-adapted* to the filtration generated by the observable variables, and thus in-sample, they overperform the original control problem by implicitly circumventing the essential restriction of the adaptedness of the decisions. Consequently, feedback actions constructed by large enough hypothesis spaces do not always generalize and perform poorly out-of-sample. Examples of Section 7.1 clearly illustrate the concept of overlearning and the consequent non-adaptedness of the actions in non-technical settings. Theorem 7.1 considers the over-parametrized limit of the trained actions or, equivalently, the limit as the hypothesis spaces gets denser. It is shown that in the limit the performance of naturally adapted trained feedback controls is equal to the performance of the strictly smaller value given by the minimization over the anticipative controls.

In data-driven applications, this capability of the artificial networks to overlearn, necessitates effective data enrichment when the available set is not sufficiently large. As these methods are data-hungry, this remark applies to a large class of problems coming from quantitative finance. The recent survey [1] provides a thorough introduction to this important question and outlines numerous approaches from statistics. A related problem is the calibration of complex models to market data. Modern optimization tools can also be utilized in this context allowing us to use more complex models and even artificial neural networks to achieve this goal. Currently there is extensive and far-reaching research on this topic. An example is the exciting recent paper [20] that use this approach to calibrate a local volatility model to market data. For further information we refer the reader to the references in [20]. Regularization is also widely used to reduce over-learning as we discuss further in Remark 7.7, below. Finally, even with these, when overlearning renders the data need computationally infeasible, one must use alternate methods such as the dynamic programming based *Deep Galerkin* [52] method widely used for Markovian models.

Despite this potential hurdle of overlearning, many papers that are closely related to dynamic deep empirical risk minimization report impressive numerical results in problems with a large number of states. Central financial problems of hedging and portfolio management are the foci of the pioneering papers [17, 18]. These studies demonstrate that complex and high-dimensional stock dynamics, and also market details such as transaction costs and market impact are within easy reach of this approach. In particular, they study multidimensional factor models, delivering convincing evidence for the flexibility and the scope of the algorithm, particularly in high-dimensions. [2, 36] develop several effective algorithms including the one discussed here as well as hybrid ones that use the Markovian structure together with dynamic programming. The convergence analysis of [2] complements the high-dimensional numerical experiments of [36]. The recent paper [19] provides an extensive and informative numerical analysis of the impact of the network architecture. Another exciting series of papers [9, 10] consider the difficult problem of optimal stopping. These papers solve numerous examples of practical interest in dimensions up to 100 and show that dynamic deep empirical risk minimization yields feedback actions achieving values very close to the upper bounds computable through their duals. The computation of the free boundary is studied in [47]. For further studies and more information, we refer the reader to the excellent survey papers [24, 37, 50] and the references therein.

The expediency of our formulation reveals itself by enabling a plethora of techniques from statistical machine learning [51], as evidenced by the non-asymptotic upper bounds proved in Section 8 via empirical Rademacher complexity [7, 8, 41]. In particular, Theorem 8.5 and the performance error estimate (8.4) are analytic manifestations of the bias-variance, or more precisely, bias-complexity trade-off (cf. [51], Chp. 5) in this context. Indeed, for fixed training sets the complexity increases with the size of the networks allowing for possible generalization error. On the other hand, as the complexity of a fixed network gets smaller with larger training data, for an appropriate combination of network structure and data, so does overlearning. Specifically, Corollary 8.6 shows that for sufficiently large training sets, actions constructed by appropriately wide or deep neural networks are close to the desired solutions and overlearning is negligible. Hence, through classical concepts, we obtain efficient estimates yielding structured convergence proofs. The style of these results complements the comprehensive analysis of [36] for controlled Markov processes and the convergence analysis of dynamic deep empirical risk minimization carried out in [31] for backward stochastic differential equations.

Our numerical experiments[1] support these theoretical observations as well. In Section 9, we analyze a stylized Merton utility maximization problem of Example 6.1. Like the previous papers,

[1]These experiments were carried out on personal computers. The code and the logs, including random number generator seeds, are available at `https://gitlab.com/mreppen/dderm` for their full reproduction.

our results also demonstrate the effectiveness of dynamic deep empirical risk minimization in handling high-dimensional problems and the convergence of the algorithm. Potential overlearning and its clear dependence on the dimension of the randomness driving the dynamics is shown by comparing the in-sample and out-of-sample performance of the trained networks. Although we employ early stopping based on out-of-sample performance, there is always some amount of overlearning whose level strongly depends on the state dimension. Our experiments with a training set of size $100,000$ and three hidden layers of width 10 show in-sample to out-of-sample performance differences ranging from 1.5% in 10 dimensions to 24% in 100 dimensions. This dependence is further corroborated by experiments controlling for the number of network parameters. Moreover, more aggressive minimization—beyond our conservative early stopping—results in substantial overlearning. In 100 dimensions, the trained network soon reaches 30% over-performance over the known true solution in 100–200 epochs, and more would be possible with further iterations. In these cases, the out-of-sample performance deteriorates rapidly.

It is well documented in the literature and proven by our estimates that the size of the training data is central to the performance of this approach. In 100 dimensions, we achieve a remarkable improvement in the accuracy of our numerical computations by increasing the size of training data. In studies in which data is simulated from a model, one does not create an initial training set, but rather simulate new data for each batch, essentially creating a large enough training set to obtain accurate results. Therefore, as discussed earlier, simulation ability that is consistent with the market data is key for this method.

Structurally we assume that the random process driving the state is uncontrolled, which is the case when it is given by the stock returns. Although most control problems are formulated differently, if their dynamics is known, with little effort many can reformulated to have this structure. This is demonstrated for time-discretized controlled diffusions in Section 3.2 and for all Markov decision processes in Appendix B.

The paper is organized as follows. The problem is defined in Section 3 and reformulated in Section 4. Section 5 describes dynamic the deep empirical risk minimization. Two motivating examples are given in Section 6 and overlearning is introduced and proved in Section 7. Error estimates based on Rademacher complexity and convergence are proved in Section 8. Section 9 outlines the specifics of the network structure, the optimization algorithm, and the experiments. After concluding remarks, Appendix A provides a generalization of the overlearning theorem and Appendix B formulates the classical Markov decision problem in our framework.

2 Notation and Conventions

This is a brief summary of our notation and conventions for quick reference. Precise definitions are given in subsequent sections.

Whenever possible, we use capital letters for random variables (with exceptions for the time of maturity T and the utility function U), lower case letters for deterministic quantities, and sets are denoted by calligraphic letters. In particular, action space $\mathcal{A}$, state space $\mathcal{X}$, and perturbation space $\mathcal{Z}$ are closed subsets of Euclidean spaces with the usual Euclidean norm. We assume all functions defined on these sets to be continuous. For a set $\mathcal{Y}$ and a positive integer t, $\mathcal{Y}^t$ is the Cartesian product of t copies of $\mathcal{Y}$.

Maturity is a finite positive integer T. A stochastic process Y taking values in $\mathcal{Y}$ is a finite sequence of $\mathcal{Y}$-valued random variables $(Y_0, Y_1, \dots, Y_T)$. A *trajectory* represents one realization of this process and is a deterministic sequence $y \in \mathcal{Y}^{T+1}$. We use parenthesized subscripts to denote

the sequence up to a time t:

$$y_{(t)} := (y_0, \dots, y_t) \in \mathcal{Y}^{t+1} \quad \text{or} \quad Y_{(t)} := (Y_0, \dots, Y_t).$$

A *feedback action* is a Borel-measurable map $a : \mathcal{T} \times \mathcal{X} \times \mathcal{Z} \to \mathcal{A}$, and for such a, the corresponding controlled state is denoted by $X^a \in \mathcal{X}^{T+1}$, with initial condition $X_0^a = x \in \mathcal{X}$. The initial condition x is considered fixed, and is included in $X_{(t)}^a = (x, X_1^a, \dots, X_t^a) \in \mathcal{X}^{t+1}$, but otherwise it is omitted in the notation. The set of all bounded, continuous feedback actions is denoted by $\mathcal{C}$. The set $\mathcal{B}$ of all Borel measurable functions $g : \mathcal{Z}^T \to \mathcal{A}^T$ is related to *anticipative controls*.

3 Optimal Feedback Controls

All investments problems we consider can be formulated as sequential decision problems under uncertainty, or, equivalently, stochastic optimal control problems in discrete-time with a finite horizon of T. Thus, we study this more general problem in which actions are taken at time points in the set

$$\mathcal{T} := \{0, 1, \dots, T-1\}.$$

We assume that a stochastic process Z, which is the stock returns in most applications, drives the dynamics of the problem. Each component Z_t is a random variable on a probability space Ω taking values in $\mathcal{Z}$, a closed subset of a Euclidean space. We always set $Z_0 = 0$ and for $t > 0$, with abuse of notation, we write

$$Z_{(t)} := (Z_1, \dots, Z_t), \quad \text{and} \quad Z := (Z_1, \dots, Z_T).$$

Let $\mathbb{F} = (\mathcal{F}_t)_{t=0,1,\dots,T}$ be the filtration generated by the process Z, i.e., for $t = 1, \dots, T$, $\mathcal{F}_t$ is the smallest sigma-algebra so that $Z_{(t)} : (\Omega, \mathcal{F}_t) \to \mathcal{Z}^t$ is Borel measurable, and $\mathcal{F}_0 = \{\emptyset, \Omega\}$ is trivial.

3.1 Dynamics and Performance

At times $t \in \mathcal{T}$, investors choose an action A_t with values in $\mathcal{A}$, a closed subset of a Euclidean space. The control process $A = (A_0, \dots, A_{T-1})$ is adapted to the filtration $\mathbb{F}$, and the resulting state process denoted by X^A takes values in the *state-space* $\mathcal{X}$, again a closed subset of a Euclidean space. In the applications, the mark-to-market value of the portfolio is always included in the state as well as other relevant quantities depending on the modeling. Given an initial condition $X_0^A = x \in \mathcal{X}$, the controlled state X^A solves the simple random difference equation

$$X_{t+1}^A = F_{t+1}(X_{(t)}^A, A_t), \qquad t \in \mathcal{T},$$

where $X_{(t)}^A = (X_0^A, \dots, X_t^A)$, and $F_{t+1} : \Omega \times \mathcal{X}^{t+1} \times \mathcal{A} \mapsto \mathcal{X}$ is $\mathcal{F}_{t+1}$-measurable (as customary, we use Borel subsets of the Euclidean spaces). The performance of the action A is measured by,

$$v(A) := \mathbb{E}[\Phi(X^A, A)], \tag{3.1}$$

where $\Phi : \Omega \times \mathcal{X}^T \times \mathcal{A}^T \mapsto \mathbb{R}$ is $\mathcal{F}_T$-measurable. The optimization problem is to minimize the above performance function over a class of feedback actions discussed in the next section.

As $\mathcal{F}_t$ is generated by $Z_{(t)}$, the dependence of any $\mathcal{F}_t$ measurable random variable on the randomness is given entirely through $Z_{(t)}$. Hence, there are functions,

$$f(t, \cdot) : \mathcal{X}^{t+1} \times \mathcal{A} \times \mathcal{Z}^{t+1} \to \mathcal{X}, \quad \text{and} \quad \varphi : \mathcal{X}^{T+1} \times \mathcal{A}^T \times \mathcal{Z}^T \to \mathbb{R},$$

so that

$$F_{t+1}(X_{(t)}^A, A_t) = f(t, X_{(t)}^A, A_t, Z_{(t+1)}), \quad \text{and} \quad \Phi(X^A, A) = \varphi(X^A, A, Z).$$

3.2 Admissible Actions

We restrict the investment decisions to be a function of the current state, hence, a *feedback action*. That is, the investors determine their actions through a bounded, continuous function of their choice

$$a : \mathcal{T} \times \mathcal{X} \times \mathcal{Z} \to \mathcal{A}.$$

Indeed, for a chosen feedback function a, one first recursively defines a process X^a by the equations,

$$X_0^a = x, \quad \text{and} \quad X_{t+1}^a = f(t, X_{(t)}^a, a(t, X_t^a, Z_t), Z_{(t+1)}), \qquad t \in \mathcal{T}. \tag{3.2}$$

Then, the corresponding control process is given by $A_t^a := a(t, X_t^a, Z_t)$. It is clear that A^a is adapted to $\mathbb{F}$ and the process X^a is equal to the state process X^{A^a} given by the control process A^a. Hence, $A_t^a = a(t, X_t^{A^a}, Z_t)$. Let $\mathcal{C}$ be the set of all bounded, continuous *feedback actions* $a : \mathcal{T} \times \mathcal{X} \times \mathcal{Z} \to \mathcal{A}$. In our notation, we use A^a and a interchangeably and write

$$v(a) := v(A^a) = \mathbb{E}[\varphi(X^a, A^a, Z)].$$

Feedback actions are easily implementable and are therefore widely used in practice.

3.3 Problem

The stochastic optimal control problem in discrete time—or an investment problem—is to

$$\text{minimize } a \in \mathcal{C} \ \mapsto \ v(a) = \mathbb{E}[\varphi(X^a, A^a, Z)], \tag{3.3}$$

where X^a is the solution of (3.2) and the expectation is over the distribution of Z. We assume that $f(t, \cdot) : \mathcal{X}^{t+1} \times \mathcal{A} \times \mathcal{Z}^{t+1} \to \mathcal{X}$ determining the dynamics in (3.2) and the cost function $\varphi : \mathcal{X}^T \times \mathcal{A}^T \times \mathcal{Z}^T \to \mathbb{R}$ in (3.3) are given and known.

Remark 3.1. For Markovian models, the restriction to feedback controls causes no loss of generality. Indeed, let $\mathcal{A}_{ad}$ be the set of all adapted and bounded processes $A \in \mathcal{A}^T$, and suppose that Z is a Markov process, $f(t, \cdot)$ depends only on Z_{t+1} and not on $Z_{(t+1)}$, and φ is given by

$$\varphi(X^A, A, Z) = \hat{\varphi}(X_T^A) + \sum_{t \in \mathcal{T}} \psi(t, X_t^A, A_t, Z_t),$$

for some given functions ψ and $\hat{\varphi}$. Then, under reasonable assumptions, one can show that among all adapted processes there are near-maximizers that are of feedback form.

Remark 3.2. The above model can also be obtained as an appropriate discretization of finite-horizon, continuous-time problems. As an example, consider the classical optimal control of diffusion processes with a finite horizon T_0 with the admissible controls $\widehat{\mathcal{A}}_{ad}$ given as the set of all adapted and bounded processes $A : [0, T_0] \to \mathcal{A}$. Then, the problem is to

$$\text{minimize } A \in \widehat{\mathcal{A}}_{ad} \mapsto \ v(A) := \mathbb{E}\bigg[\int_0^{T_0} \tilde{\psi}(u, \widehat{X}_u^A, A_u) \mathrm{d}u + \tilde{\varphi}(\widehat{X}_{T_0}^A) \bigg],$$

subject to dynamics

$$d\widehat{X}_u^A = \mu(t, \widehat{X}_u^A, A_u) \mathrm{d}u + \sigma(u, \widehat{X}_u^A, A_u) \mathrm{d}W_u,$$

where W is a Brownian motion and $\tilde{\psi}, \tilde{\varphi}$ are functions independent of randomness. Euler–Maruyama discretization of this model is a discrete-time decision problem with

$$\varphi(X^A, A, Z) = \sum_{t\in\mathcal{T}} \tilde{\psi}(t\Delta t, X_t^A, A_t)\Delta t + \tilde{\varphi}(X_T^A),$$

$$f(t, X_{(t)}^A, A_t, Z_{(t+1)}) = X_t^A + \mu(t\Delta t, X_t^A, A_t)\Delta t + \sigma(t\Delta t, X_t^A, A_t)Z_{t+1},$$

where $A_t = A_{t\Delta t}$, $Z_{t+1} = \Delta W_{t\Delta t} := W_{(t+1)\Delta t} - W_{t\Delta t}$. If the original function μ, σ, $\tilde{\psi}$, $\tilde{\varphi}$ also depend on the randomness in an adapted manner, this would introduce dependencies of φ and f on the past Brownian increments as well. We refer to the classical text book [22] for related results.

Classical Markov decision processes are discussed in Appendix B.

Remark 3.3. One important but subtle property of this problem is that, given data, any policy can be evaluated without the need for further data collection or interaction with the system. For instance, a small investor does not impact stock dynamics when trading, and can therefore observe the returns and reason ex post about what would have happened under other trading strategies. This is in stark contrast to many engineering applications where controls alter the physical trajectories, and a change in the control requires a new observation or simulation. This structure, which is pervasive in financial literature, is what allows direct optimization of the strategy (which we formulate as empirical risk minimization in this paper) in lieu of more general reinforcement learning methods. However, the optimization used in the algorithm requires large training sets necessitating the construction of data-driven market models for simulations. In this exciting new area of research, reinforcement learning may play a central role.

4 Reformulation

In this section, we provide a reformulation that enables us to write it as a problem of empirical risk minimization in Section 5. For a feedback action $a \in \mathcal{C}$, an initial value $x \in \mathcal{X}$, and a (deterministic) trajectory $z = (z_1, z_2, \dots, z_T)$ with $z_0 = 0$, we define the controlled state values $x^a = (x_0^a, x_1^a, \dots, x_T^a)$ recursively by the equations

$$x_0^a = x, \quad \text{and} \quad x_{t+1}^a = f(t, x_{(t)}^a, a(t, x_t^a, z_t), z_{(t+1)}), \quad t \in \mathcal{T},$$

where f is as in (3.2), $x_{(t)}^a = (x_0^a, \dots, x_t^a)$, $z_{(t+1)} = (z_1, \dots, z_{t+1})$. The above solution, denoted by $x^a(z) = (x_1^a(z), \dots, x_T^a(z))$, is a function of the trajectory z and is called the *state function*. Then, for each process $Z \in \mathcal{Z}^T$, the unique solution of (3.2) is given by $X^a = x^a(Z)$. Further, let $\alpha^a(z) := (\alpha_t^a(z), \dots, \alpha_{T-1}^a(z))$ be given by, $\alpha_t^a(z) := a(t, x^a(z), z_t)$ for $t \in \mathcal{T}$ so that $A^a = \alpha^a(Z)$.

Set

$$\ell(a, z) := \varphi(x^a(z), \alpha^a(z), z), \qquad a \in \mathcal{C},\ z \in \mathcal{Z}^T, \tag{4.1}$$

where φ is as in (3.3). As $X^a = x^a(Z)$ and $A^a = \alpha^a(Z)$, the performance function $v(a)$ of (3.3) is equal to $\mathbb{E}[\ell(a, Z)]$. Hence, the dynamic decision problem (3.3) is equivalent to

$$\text{minimize } a \in \mathcal{C} \ \mapsto\ v(a) = \mathbb{E}[\ell(a, Z)]. \tag{4.2}$$

Its *optimal value* is given by

$$v^* := \inf_{a\in\mathcal{C}} v(a) = \inf_{a\in\mathcal{C}} \mathbb{E}[\ell(a, Z)]. \tag{4.3}$$

Precisely this structure leads to empirical risk minimization and is quite conducive to analysis.

4.1 Adapted and Anticipative controls

This reformulation of the decision problem is not restricted to feedback actions. Indeed, let $\mathcal{A}_{nt}$ be the set of all $\mathcal{A}^T$-valued random vectors that are $\mathcal{F}_T$ measurable. Elements of $\mathcal{A}_{nt}$ can be parametrized by the set $\mathcal{B}$ of all Borel measurable functions $g = (g_0, \dots, g_{T-1}) : \mathcal{Z}^T \to \mathcal{A}^T$:

$$\mathcal{A}_{nt} = \{A^g \ : \ g \in \mathcal{B}\}, \qquad \text{where} \qquad A^g_t = g_t(Z), \quad t \in \mathcal{T}. \tag{4.4}$$

Proceeding as above, we construct a state function $\hat{x}^g : \mathcal{Z}^T \to \mathcal{X}^{T+1}$ so that $X^{A^g} = \hat{x}^g(Z)$ and define α^g similarly. Hence, the cost function $\varphi(X^{A^g}(Z), \alpha^g(Z), Z)$ is a function of g and Z. Set

$$\ell(g, z) := \varphi(\hat{x}^g(z), \alpha^g(z), z), \qquad g \in \mathcal{B}, \ z \in \mathcal{Z}^T.$$

with which another empirical risk minimization problem also can be formulated with anticipative controls in $\mathcal{A}_{nt}$.

Note that processes in $\mathcal{A}_{nt}$ are *not adapted* to $\mathbb{F}$ and at any time they may use all available information. As they may anticipate and use the future, we refer to them as *anticipative actions*. Clearly, $\mathcal{A}_{nt}$ is strictly larger than the set $\mathcal{A}_{ad}$ of actions *adapted* to $\mathbb{F}$ and therefore,

$$v^*_{nt} := \inf_{A \in \mathcal{A}_{nt}} v(A) \le v^*_{ad} := \inf_{A \in \mathcal{A}_{ad}} v(A) \le v^* := \inf_{a \in \mathcal{C}} v(a).$$

In all non-trivial control problems, $v^*_{nt} < v^*_{ad}$ and as discussed in Remark 3.1 above, in Markovian models we usually have $v^*_{ad} = v^*$.

5 Dynamic Deep Empirical Risk Minimization

In this section, we outline the approach of [30, 32] which can be seen as empirical risk minimization in view of the above reformulation.

The *training set* is a collection of n observations of the random process Z,

$$\mathcal{L}_n = \left\{ Z^{(1)}, Z^{(2)}, \dots, Z^{(n)} \right\} \quad \text{where} \quad Z^{(i)} = (Z^{(i)}_1, Z^{(i)}_2, \dots, Z^{(i)}_T) \in \mathcal{Z}^T.$$

On this set, in correspondence with (4.2), the empirical *loss function* for $a \in \mathcal{C}$ is defined by,

$$L(a; \mathcal{L}_n) := \frac{1}{n} \sum_{i=1}^{n} \ell(a, Z^{(i)}). \tag{5.1}$$

As we assume that $Z^{(i)}$ are drawn independently from their distribution, $L(a; \mathcal{L}_n)$ is an approximation of $v(a) := \mathbb{E}[\ell(a, Z))]$.

We consider a sequence of *hypothesis spaces*

$$\mathcal{N}_k := \left\{ h(\cdot; \theta) \ : \ \theta \in \mathcal{O}_k \right\}, \qquad k = 1, 2, \dots,$$

where for each parameter θ, $h(\cdot; \theta) : \mathcal{T} \times \mathcal{X} \times \mathcal{Z} \to \mathcal{A}$ is a feedback action. We assume that the sequence of parameter sets $\mathcal{O}_k \subset \mathbb{R}^{d(k)}$ are compact subsets with increasing dimensions $d(k)$ and that h is a continuous function of its variables. In our numerical experiments, we use an artificial neural network with several hidden layers as our hypothesis space. However, for theoretical considerations, the only requirement we impose on the sequence $\mathcal{N}_k$ is that they satisfy the below assumption which can be seen as the approximation capability or being asymptotically "pointwise" dense in the set of continuous functions. It is well known that sequences of neural networks have this property as proved by [21], [35].

Assumption 5.1 (Pointwise Density)**.** *We assume that for any bounded continuous function $\hat{a} : \mathcal{T} \times \mathcal{X} \times \mathcal{Z} \to \mathcal{A}$, there exists a sequence $\{a_k\}_{k\in\mathbb{N}}$ such that $a_k \in \mathcal{N}_k$ for each $k \in \mathbb{N}$, and a_k converges to $\hat{a}$ pointewise, i.e., $\lim_{k\to\infty} a_k(t, x, z) = \hat{a}(t, x, z)$ for every $(t, x, z) \in \mathcal{T} \times \mathcal{X} \times \mathcal{Z}$.*

We also make the following simplifying regularity assumption on the coefficients.

Assumption 5.2 (Regularity)**.** *We assume that f, φ are uniformly bounded and continuous.*

As an immediate consequence of the regularity assumption, there exist a constant c^*, so that ℓ defined in (4.1) satisfies

$$|\ell(a, z)| \leq c^*, \qquad \forall\ z \in \mathcal{Z}^T,\ a \in \mathcal{C}.$$

Moreover, for any a_n converging pointwise to a feedback action $\hat{a}$, $\lim_{n\to\infty} \ell(a_n, z) = \ell(\hat{a}, z)$ for every $z \in \mathcal{Z}$. In particular, by dominated convergence, $\lim_{n\to\infty} v(a_n) = v(\hat{a})$.

These assumptions easily imply that the sequence $\mathcal{N}_k$ can approximate the optimal value. Further convergence results are proved in Section 8. Let v^* be as in (4.3) and set

$$v_k^* := \inf_{a\in\mathcal{N}_k} v(a) = \inf_{\theta\in\mathcal{O}_k} v(h(\cdot, \theta)).$$

Lemma 5.3. *Suppose that the above density and regularity assumptions hold. Then,*

$$\lim_{k\to\infty} v_k^* = v^*.$$

Proof. Fix $\epsilon > 0$ and let $a_\epsilon^* \in \mathcal{C}$ be an ϵ-minimizer of v: $v(a_\epsilon^*) \leq v^* + \epsilon$. In view of the density assumption, there exists a sequence $a_k \in \mathcal{N}_k$ such that a_k converges to a_ϵ^* pointwise. Then, by the regularity assumption, $\limsup_{k\to\infty} v_k^* \leq \lim_{k\to\infty} v(a_k) = v(a_\epsilon^*) \leq v^* + \epsilon$. As the opposite inequality $v^* \leq v_k^*$ holds trivially for every k, we conclude that v_k^* converges to v^*. □

Training. We fix the training set $\mathcal{L}_n$ and the hypothesis space $\mathcal{N}_k$, and

$$\text{minimize} \quad \theta \in \mathcal{O}_k \ \mapsto\ L(h(\cdot; \theta); \mathcal{L}_n). \tag{5.2}$$

As L, h are continuous and $\mathcal{O}_k$ is compact, there exists a minimizer $\theta_{k,n}^* \in \mathcal{O}_k$. Then, the continuous function $A_{k,n}^* := h(\cdot; \theta_{k,n}^*)$ is the *trained feedback action* that could be constructed by $\mathcal{N}_k$ using $\mathcal{L}_n$, and

$$V_k^*(\mathcal{L}_n) := L(A_{k,n}^*; \mathcal{L}_n) = \inf_{a\in\mathcal{N}_k} L(a; \mathcal{L}_n) \tag{5.3}$$

is the *optimal in-sample performance* of the hypothesis space $\mathcal{N}_k$ on the given data $\mathcal{L}_n$.

The effectiveness of this algorithm depends on the size of the training set $\mathcal{L}_n$, the architecture of the hypothesis space $\mathcal{N}_k$, and on their interactions, and our main goal is to study these. Since numerically one can only construct an approximation of the above minimizer, details of the approximating optimization procedure are an essential part of the algorithm. In our experiments, we use a standard stochastic gradient descent variant with early stopping based on the test-set performance (cf. Section 9.1 for implementation details). In studies with simulated data, one does not create an initial training set, but rather simulate new data for each batch until training stalls or a stopping rule is satisfied.

In the literature several alternatives to direct empirical risk minimization has been discussed and their use might be beneficial. We refer the reader to a recent paper [4] and the references therein. As our main goal is to analyze the original algorithm, we do not consider these alternatives in this manuscript.

6 Examples

We briefly outline two classes of problems to clarify the model and the notation. Further examples can be found in the forthcomig paper [48].

6.1 Merton Problem

Here we only outline a simple portfolio management problem in a financial market with d many assets. Although this example does not include many important modeling details, it must be clear that by appropriately choosing $Z, \mathcal{A}, X^a$ and the dynamics, one can cover essentially all Merton type utility maximization, portfolio management, and hedging problems studied in the literature. Also problems with different structures such as free boundary problems studied in [9] and the hedging problems with frictions in [17, 18] can be included in our framework.

Let $S_t \in \mathbb{R}^d_+$ denote the stock price process and assume that one-period interest rate r is constant. The control variable $\pi_t = (\pi_t^1, \dots, \pi_t^d) \in \mathbb{R}^d$ is the amount of money to be invested in each of the stock. Classically, it is assumed that π_t^i could take any value. Starting with initial wealth of $x > 0$, the self-financing wealth dynamics for the portfolio choice π_t are given by

$$X_{t+1} = X_t + \pi_t \cdot Z_{t+1} + r(X_t - \pi_t \cdot \mathbf{1}) = (1+r)X_t + \pi_t \cdot (Z_{t+1} - r\mathbf{1}) \quad t \in \mathcal{T},$$

where $X_0 = x$, $\mathbf{1} = (1, \dots, 1) \in \mathbb{R}^d$ and the return process Z is given by

$$Z_{t+1} = \frac{S_{t+1} - S_t}{S_t} \in \mathbb{R}^d, \quad t \in \mathcal{T}.$$

We consider feedback controls $\pi_t = a(t, X_t^a, Z_t)$ and let X^a be the corresponding wealth process. Then, the classical problem is to maximize $v(a) := \mathbb{E}[U(X_T^a)]$ with a given utility function U.

Example 6.1. In Section 9, we numerically study the following stylized example with an explicit solution in detail, to illustrate the convergence of the algorithm, potential overlearning, and the influence of the dimension on them. We take the initial wealth $X_0 = x = 0$, $T = 2$, and use an exponential utility $U(x) = 1 - e^{-\lambda x}$ where $\lambda > 0$ is the risk-aversion parameter.

To simplify even further, we assume that initially one dollar is borrowed and invested uniformly on all stocks. Then, $\pi_0 = (1/d, \dots, 1/d)$ and $X_1 = (Z_1 \cdot \mathbf{1})/d - r$ are uncontrolled, and the investment problem is to choose the feedback portfolio $a(Z_1) := \pi_1(X_1, Z_1) \in \mathbb{R}^d$ so as to maximize

$$v(a) = \mathbb{E}\big[U(X_2^a)\big] = \mathbb{E}\big[1 - \exp(-\lambda X_2^a)\big],$$

where $X_2^a = (1+r)X_1 + a(Z_1) \cdot (Z_2 - r\mathbf{1})$. A more standard way of comparing different utility values is the *certainty equivalent* of a utility value $v < 1$, given by

$$\mathrm{ce}(v) := \frac{1}{\lambda} \ln(1 - v) \quad \Longleftrightarrow \quad v = U(\mathrm{ce}(v)).$$

In the numerical experiments, to reduce the output noise, we fix a unit vector $\eta \in \mathbb{R}^d$ and take $Z_2 = \zeta\eta$, where the real-valued Gaussian random variable ζ is independent of Z_1 and has mean m and volatility s. Then, with $r = 0$,

$$a^*(z) = a^* = \frac{m}{\lambda s^2}\,\eta, \quad v^* = v(a^*) = 1 - \exp(-\frac{m^2}{2s^2}), \quad \mathrm{ce}(v^*) = -a^*.$$

6.2 Production Planning

The multi-stage optimization problems introduced by [13, 14] is in the above structure as well. Although not in quantitative finance, here we describe a simple example of these problems that is very similar to Example 1 in [13], to clarify many of the notions introduced in the paper.

We consider producers facing an optimal production decision. They observe the random demand Z_1, Z_2 in two stages. The production level $a(Z_1)$ is decided after observing Z_1 but before Z_2 and the second component of the random demand Z_2 is observed afterwards, at the final stage. The goal is to bring the final inventory level close to zero by properly choosing the production level at stage one. Let X^a be the inventory level. We assume the initial inventory is zero and no production is made initially. Then, $X_1^a = Z_1$, $X_2^a = X_1^a - a(Z_1) + Z_2$, and the problem is to minimize

$$v(a) = \mathbb{E}\big[\varphi(X_2^a)\big] = \mathbb{E}\big[\varphi\big(Z_1 + Z_2 - a(Z_1)\big)\big]$$

over all production functions a. The penalty function $\varphi \geq 0$ is convex and is equal to zero only at the origin. In our framework, $\mathcal{A} = [0, \infty)$, and $f(t, x, z, a) = y - a + z$.

For $\varphi(x) = x^2$ this is exactly the classical regression problem of estimating the total demand $Z_1 + Z_2$ after observing the first component Z_1. It is well-known that the optimal solution is $a^*(Z_1) = \mathbb{E}[Z_1 + Z_2 \mid Z_1]$, and this optimization problem reduces to the classical regression well-known to face the bias-variance trade-off. Although this connection may not be as explicit in other more complex models, it is always inherent to the problem.

7 Overlearning

Recall the set of anticipative controls $\mathcal{A}_{nt}$ of Section 4.1, $V_k^*(\mathcal{L}_n)$ of (5.3) and set

$$V^*(\mathcal{L}_n) := \lim_{k\to\infty} V_k^*(\mathcal{L}_n) = \lim_{k\to\infty} \inf_{a\in\mathcal{N}_k} L(a; \mathcal{L}_n).$$

The following result is proved in Section 7.2 below, under the natural assumption of distinct data and its relaxation is discussed in the Appendix A.

Theorem 7.1. *Suppose that the density and the regularity assumptions (c.f. Assumptions 5.1, 5.2) hold and the training data is distinct, i.e., for every $t \in \mathcal{T}$ and $i \neq j$, $Z_t^{(i)} \neq Z_t^{(j)}$. Then,*

$$\limsup_{n\to\infty} V^*(\mathcal{L}_n) \leq v_{nt}^* := \inf_{A\in\mathcal{A}_{nt}} v(A).$$

In all non-trivial decision problems, $v_{nt}^* < v^* := \inf_{a\in\mathcal{C}} v(a)$. Thus, sufficiently large hypothesis spaces, in-sample, overperform the optimal value v^*. In optimal control, it is centrally important that the decisions are adapted to the information flow. The above results show that hypothesis spaces circumvent this restriction by predicting the future values the data and are thus able to overperform on the training set. We emphasize that the training is done in the hypothesis class $\mathcal{N}_k$ and the elements of $\mathcal{N}_k$ are of feedback form. Hence, they are naturally adapted processes. This is further discussed in Remark 7.6 below. We refer to this possibility as *overlearning*.

7.1 Examples

We return to two examples from Section 6 to clarify the above discussion.

We first consider the production planning problem of Section 6.2. The only feedback action in that context is the production decision. For a fixed control $\alpha \in \mathbb{R}$ and given demands $z = (z_1, z_2)$, the cost function $\ell(\alpha, z) = \varphi(z_1 + z_2 - \alpha) \geq 0$ is zero at the origin. Hence, $a^*(z) = z_1 + z_2$ is the pointwise optimizer. If the training data $\mathcal{L} = \{Z^{(1)}, Z^{(2)}, \dots, Z^{(n)}\}$ is distinct, any sufficiently large $\mathcal{N}_k$ has an element a_k^* such that $a_k^*(Z_1^{(i)})$ is uniformly close to $Z_2^{(i)} + Z_2^{(i)}$ for each i. Then, the feedback action a_k^* constructed by $\mathcal{N}_k$ achieves an in-sample performance value close to zero yielding $V^*(\mathcal{L}_n) = 0$. As $v^* > 0$, this would be overlearning and a_k^* does not generalize.

Next, consider the utility maximization problem discussed in Example 6.1 with one stock. Then, for control $\alpha \in \mathbb{R}$ and returns $z = (z_1, z_2)$, $\ell(\alpha, z) = 1 - \exp(-\lambda[(1+r)(z_1 - r) + \alpha(z_2 - r)])$. By taking arbitrarily large positions depending on the sign of $z_2 - r$, one obtains $V^*(\mathcal{L}_n) = 1$. In financial terms, large enough hypothesis spaces anticipate the sign of the random variable $Z_2 - r$ by observing Z_1 and use it to create numerical arbitrage caused by the obvious non-adaptedness and overlearning on the training data. In particular, the trained feedback actions almost achieve a performance value of one, and thus overperform the optimal value $v^* < 1$. Additionally, in this example, the optimal value obtained by the anticipative controls is also equal to one, $v_{nt}^* = 1$, which is consistent with Theorem 7.1.

Remark 7.2. In closely related studies Pflug and Pichler analyze the dependence of optimization problems on the distribution of the randomness. In our terminology, they prove that overlearning implies that the limit of the values obtained by empirical measures do not convergence. Motivated by this observation, they carefully define the nested (or adapted) distance among probability measure which yields the continuity of the value function, cf. Proposition 1 in [46]. A similar observation is also made in Example 7.1 in [3].

7.2 Asymptotic Overlearning

We continue with an estimate used in the proof of Theorem 7.1. For any $\alpha \in \mathcal{A}^T$, we define a constant (in space) action by $A_t^\alpha := \alpha_t$. With abuse of notation, we consider $\alpha \in \mathcal{A}^T$ as an element of $\mathcal{C}$. Recall that the anticipative actions $\mathcal{A}_{nt}$ are parametrized by $\mathcal{B}$ of $\mathcal{F}_T$ measurable functions, cf. (4.4).

Lemma 7.3. *It holds that,*

$$L_n^* := \frac{1}{n}\sum_{i=1}^n \inf_{\alpha \in \mathcal{A}^T} \ell(\alpha, Z^{(i)}) \leq \frac{1}{n}\sum_{i=1}^n \ell(g, Z^{(i)}) = L(A^g; \mathcal{L}_n), \qquad \forall g \in \mathcal{B}. \tag{7.1}$$

Proof. Fix $\hat{z} \in \mathcal{Z}^T$, $g = (g_0, \dots, g_{T-1}) \in \mathcal{B}$ and let $\hat{x}^g$ be the state function defined in Section 4.1. Define a constant action $\hat{\alpha} := (\hat{\alpha}_0, \dots, \hat{\alpha}_{T-1})$ by $\hat{\alpha}_t = g_t(\hat{z})$ for $t \in \mathcal{T}$. Let $x^{\hat{\alpha}}$ be the corresponding state process. Then, by a simple induction argument on the time variable, we can show that $x^{\hat{\alpha}}(\hat{z}) = \hat{x}^g(\hat{z})$ (the process $x^{\hat{\alpha}}(z)$ is possibly not equal to $\hat{x}^g(z)$ for trajectories z other than $\hat{z}$). Therefore, $\ell(\hat{\alpha}, \hat{z}) = \ell(g, \hat{z})$ and

$$\inf_{\alpha \in \mathcal{A}^T} \ell(\alpha, \hat{z}) \leq \ell(\hat{\alpha}, \hat{z}) = \ell(g, \hat{z}) \quad \Rightarrow \quad \inf_{\alpha \in \mathcal{A}^T} \ell(\alpha, Z^{(i)}) \leq \ell(g, Z^{(i)}), \quad i = 1, \dots, n.$$

The inequality $L_n^* \leq L(A^g; \mathcal{L}_n)$ now follows directly. □

Theorem 7.4. *Under the hypotheses of Theorem 7.1, $V^*(\mathcal{L}_n) = L_n^*$ for every n.*

Proof. Fix $\mathcal{L}_n, \epsilon > 0$ and for $i = 1, \dots, n$, choose $\alpha^{(i)} \in \mathcal{A}^T$ satisfying

$$\ell(\alpha^{(i)}, Z^{(i)}) \leq \inf_{\alpha \in \mathcal{A}^T} \ell(\alpha, Z^{(i)}) + \epsilon.$$

Since $Z^{(i)}$ are distinct, there exists a bounded, smooth function $a_\epsilon : \mathcal{T} \times \mathcal{X} \times \mathcal{Z} \to \mathcal{A}$, such that

$$a_\epsilon(t, \xi, Z_t^{(i)}) = \alpha_t^{(i)}, \quad t \in \mathcal{T},\ \xi \in \mathcal{X}^T,\ \ i = 1, \dots, n.$$

For each $Z^{(i)} \in \mathcal{L}_n$, by induction over time, one can show that $x^{a_\epsilon}(Z^{(i)}) = x^{\alpha^{(i)}}(Z^{(i)})$. Therefore, $\ell(a_\epsilon, Z^{(i)}) = \ell(\alpha^{(i)}, Z^{(i)})$ for each i, and

$$L(a_\epsilon; \mathcal{L}_n) = \frac{1}{n} \sum_{i=1}^n \ell(a_\epsilon, Z^{(i)}) = \frac{1}{n} \sum_{i=1}^n \ell(\alpha^{(i)}, Z^{(i)}) \leq \frac{1}{n} \sum_{i=1}^n \inf_{\alpha \in \mathcal{A}^T} \ell(\alpha, Z^{(i)}) + \epsilon = L_n^* + \epsilon.$$

Moreover, by the density assumption (c.f. Assumption 5.1), there is a sequence $\{a_k \in \mathcal{N}_k\}_{k \in \mathbb{N}}$ (depending on the fixed training set $\mathcal{L}_n$ and ϵ) that approximates a_ϵ pointwise. We now use the regularity assumption (c.f. Assumption 5.2) to conclude that

$$\lim_{k \to \infty} L(a_k; \mathcal{L}_n) = L(a^\epsilon; \mathcal{L}_n).$$

Hence,

$$V^*(\mathcal{L}_n) := \lim_{k \to \infty} \inf_{a \in \mathcal{N}_k} L(a; \mathcal{L}_n) \leq \lim_{k \to \infty} L(a_k; \mathcal{L}_n) = L(a^\epsilon; \mathcal{L}_n) \leq L_n^* + \epsilon,$$

and consequently, $V^*(\mathcal{L}_n) \leq L_n^*$. The opposite inequality follows from Lemma 7.3. Indeed, as $\mathcal{N}_k \subset \mathcal{C} \subset \mathcal{B}$, (7.1) implies that $L_n^* \leq L(a; \mathcal{L}_n)$ for any k and $a \in \mathcal{N}_k$. As L_n^* is independent of k and a, we first take the infimum over $a \in \mathcal{N}_k$ and then let k tend to infinity to arrive at $L_n^* \leq V^*(\mathcal{L}_n)$. □

Proof. (of Theorem 7.1). In view of Theorem 7.4 and Lemma 7.3, $V^*(\mathcal{L}_n) = L_n^* \leq L(A^g; \mathcal{L}_n)$ for any $g \in \mathcal{B}$. As the training data is drawn independently from the distribution of Z, by law of large numbers,

$$\limsup_{n \to \infty} V^*(\mathcal{L}_n) \leq \lim_{n \to \infty} L(A^g; \mathcal{L}_n) = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^n \ell(g, Z^{(i)}) = \mathbb{E}[\ell(g, Z)] = v(A^g).$$

We complete the proof by taking infimum over $g \in \mathcal{B}$. □

Remark 7.5. If for every $\epsilon > 0$, there exists $g_\epsilon \in \mathcal{B}$ satisfying

$$\ell(g_\epsilon(z), z) \leq \inf_{g \in \mathcal{B}} \ell(g, z) + \epsilon, \qquad \forall z \in \mathcal{Z},$$

then, instantly it follows that $\limsup_{n \to \infty} V^*(\mathcal{L}_n) = v_{nt}^*$. Moreover, one may construct g_ϵ through a standard use of a measurable selection theorem under some mild additional assumptions on the functions. As this result is tangential to the main trust of the paper, we chose to omit this technical discussion.

Remark 7.6. We have shown that the trained actions may overperform the optimal value v^*. However, as they are in feedback form, theoretically the expected value of their performance is bounded by v^*. So overperformance is a subtle and a data-dependent one. Indeed, the *coefficients* of the trained actions use the future data explicitly and therefore become *non-adapted* on the *training data* and the upper bound v^* obtained by adapted actions does not hold. On the other hand, their out-of-sample performance are bounded by v^* and in our numerical studies they underperform substantially.

The proof of Theorem 7.4 also shows the importance of the dimension d as well. Indeed, in higher dimensions, the training data is 'more and more distinct' allowing for easier overlearning, an effect we observe numerically as well. The separation between the training data is also a factor in the Rademacher complexity that is discussed in the next section.

Remark 7.7 (Regularization)**.** A common approach to reduce over-learning is to add regularization such as restricting the hypothesis classes $\mathcal{N}_k$ to be subsets of the set of K-Lipschitz functions,

$$Lip_K := \{h \in \mathcal{C} \ : \ |h(z) - h(z')| \leq K|z - z'|\}.$$

Then, we argue in Remark 8.4 below that Rademacher complexity of these restricted spaces goes to zero as the training data gets larger. As we prove in the subsection 8.2 below, this convergence implies that

$$\lim_{n\to\infty} \lim_{k\to\infty} \inf_{a\in\mathcal{N}_k\cap Lip_K} L(a;\mathcal{L}_n) = \inf_{a\in\mathcal{C}\cap Lip_K} v(a).$$

In many control problems, the right-hand side of above converges to the optimal value as the Lipschitz constant K gets larger. However, in financial applications this constant is large and thus, we are close to the limit considered Theorem 7.4, making the data need very large. Another numerical difficulty is to restrict the Lipschitz constant of deep neural networks.

8 Estimates and Convergence

We first recall several classical definitions and results, cf. [6, 7, 8, 15, 16, 41].

Let $\mathcal{G}$ be a hypothesis space of a set of real-valued functions defined on the set of trajectories.

Definition 8.1. The *empirical Rademacher complexity* of $\mathcal{G}$ on the training set $\mathcal{L}_n$ is given by

$$R_e(\mathcal{G};\mathcal{L}_n) := \mathbb{E}\left[\sup_{g\in\mathcal{G}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i g(Z^{(i)}) \,\middle|\, \mathcal{L}_n\right],$$

where the expectation is over the *Rademacher variables* σ_i, which are identically and independently distributed taking values ± 1 with equal probability.

Definition 8.2. The *Rademacher complexity* of $\mathcal{G}$ is given by

$$r(\mathcal{G};n) := \mathbb{E}[R_e(\mathcal{G},\mathcal{L}_n)],$$

where the expectation is over the random training set $\mathcal{L}_n = \{Z^{(1)}, Z^{(2)}, \ldots, Z^{(n)}\}$ whose elements are independently and identically drawn and the dependence on this distribution is not shown in our notation.

Let $v : \mathcal{C} \to \mathbb{R}$ be as in (3.3), L be as in (5.1). For $\mathcal{N}_k$, $\mathcal{L}_n$, set

$$G(\mathcal{N}_k,\mathcal{L}_n) := \sup_{a\in\mathcal{N}_k} |v(a) - L(a;\mathcal{L}_n)|.$$

The following result that uniformly connects empirical averages to expected values is classical. Suppose that $\mathcal{L}_n$ is drawn independently and identically and $|g| \leq c^*$ for every $g \in \mathcal{G}$. Then, for a given $\delta \in (0,1)$, with probability at least $1-\delta$ the following estimates hold,

$$G(\mathcal{N}_k, \mathcal{L}_n) \leq c(\mathcal{N}_k, n, \delta) \leq C_e(\mathcal{N}_k, \mathcal{L}_n, \delta), \tag{8.1}$$

where with $\ell(\mathcal{N}_k) = \{\ell(a,\cdot) \mid a \in \mathcal{N}_k\}$,

$$c(\mathcal{N}_k, n, \delta) := 2r(\ell(\mathcal{N}_k); n) + 2c^* \sqrt{\frac{\ln(2/\delta)}{2n}},$$
$$C_e(\mathcal{N}_k, \mathcal{L}_n, \delta) := 2R_e(\ell(\mathcal{N}_k); \mathcal{L}_n) + 6c^* \sqrt{\frac{\ln(2/\delta)}{n}}.$$

One-sided version of these estimates for functions $0 \leq g \leq 1$ is proved, for instance in Theorem 3.3 by [44] and elementary arguments yield the above two-sided estimates.

Remark 8.3 (Complexities of Neural Networks)**.** Suppose that the hypothesis spaces $\mathcal{N}_k$ is a sequence of neural networks with increasing depth and width.

As the neural networks $\mathcal{N}_k$ get wider and deeper, the constant $c(\mathcal{N}_k, n, \delta)$ and the random variables $G(\mathcal{N}_k, \mathcal{L}_n), C_e(\mathcal{N}_k, \mathcal{L}_n, \delta)$ increase. The monotonicity in the δ variable is also clear. One may obtain further estimates by using the Rademacher calculus as described in Section 26.1 of [51]. Indeed, if the mapping $a \in \mathcal{C} \mapsto \ell(a,z)$ is uniformly Lipschitz, then the Kakade & Tewari composition Lemma (see [39], also Lemma 26.9 in [51]) implies that one can estimate the complexities $R_e(\ell(\mathcal{N}_k); \mathcal{L}_n)$ and $r(\ell(\mathcal{N}_k); n)$ by the Rademacher complexities $R_e(\mathcal{N}_k; \mathcal{L}_n)$, $r(\mathcal{N}_k; n)$ of the neural networks.

Moreover, one may use the Massart Lemma together with what is known as the Pisier's trick and the composition lemma, to prove that the Rademacher complexity $r(\mathcal{N}_k; n)$ of the neural networks converges to zero as the size n of the training data goes to infinity; see for example problem 3.11 in [44] or Corollary 3.8 in the lecture notes of [54]. In fact detailed estimates are also available in [28, 45]. Since the regularity of ℓ can be directly proven under Lipschitz assumptions on the coefficients of the decision problem, this procedure shows that under natural assumptions on the coefficients, the complexity $r(\ell(\mathcal{N}_k); n)$ also converges to zero.

Remark 8.4 (Lipschitz spaces)**.** The complexities $R_\epsilon(\ell(\mathcal{N}_k \cap Lip_K); \mathcal{L}_n)$ can be effectively estimated by the deep convergence rates obtained in [26] for the uniform convergence of the empirical measure in the Wasserstein metric W_1. As for any two probability measures μ, ν, and $a \in Lip_K$

$$|(\mu, a) - (\nu, a)| \leq K W_1(\mu, \nu),$$

convergence of the complexity directly follows from [26].

8.1 Complexity Estimates

Recall that v^* is defined in (4.3), $\theta^*_{k,n}$ is a minimizer of (5.2) and $A^*_{k,n} := h(\cdot; \theta^*_{k,n})$ is an optimal feedback action that can be constructed by $\mathcal{N}_k$ on the set $\mathcal{L}_n$. Let $\widehat{\mathcal{L}}_n$ be another data set drawn identically and independently from the same distribution as $\mathcal{L}_n$. In this section, we obtain empirical bounds on the differences of the in-sample performance $L(A^*_{k,n}; \mathcal{L}_n)$, out-of-sample performance $L(A^*_{k,n}; \widehat{\mathcal{L}}_n)$, and the average performance $v(A^*_{k,n})$ of $A^*_{k,n}$, as well as their deviations from v^*.

Theorem 8.5. *Under the density and the regularity assumptions (c.f. Assumptions 5.1, 5.2) , for every $\epsilon > 0$, there exists k_ϵ such that*

$$\left|v^* - L(A^*_{k,n}; \mathcal{L}_n)\right| \leq G(\mathcal{N}_k, \mathcal{L}_n) + \epsilon, \quad \forall\ k \geq k_\epsilon. \tag{8.2}$$

In particular, for all $\delta > 0$ the following hold with at least $1-\delta$ probability for every $k \geq k_\epsilon$,

$$\begin{aligned} \left|v^* - L(A^*_{k,n}; \mathcal{L}_n)\right| &\leq c(\mathcal{N}_k, n, \delta) + \epsilon \leq C_e(\mathcal{N}_k, \mathcal{L}_n, \delta) + \epsilon, \\ \left|v^* - v(A^*_{k,n})\right| &\leq 2c(\mathcal{N}_k, n, \delta) + \epsilon \leq 2C_e(\mathcal{N}_k, \mathcal{L}_n, \delta) + \epsilon. \end{aligned} \tag{8.3}$$

Proof. For $\epsilon > 0$ choose $a^*_\epsilon \in \mathcal{C}$ satisfying $v(a^*_\epsilon) \leq v^* + \frac{1}{2}\epsilon$. By Assumptions 5.1 and 5.2, there exists a sequence $a_k \in \mathcal{N}_k$ and k_ϵ such that $v(a_k) \leq v(a^*_\epsilon) + \frac{1}{2}\epsilon$ for all $k \geq k_\epsilon$. Hence,

$$v(a_k) \leq v(a^*_\epsilon) + \frac{1}{2}\epsilon \leq v^* + \epsilon, \quad \forall k \geq k_\epsilon.$$

Since $L(A^*_{k,n}; \mathcal{L}_n) \leq L(a; \mathcal{L}_n)$ for any $a \in \mathcal{N}_k$, the definition of G implies that

$$\begin{aligned} L(A^*_{k,n}; \mathcal{L}_n) &\leq L(a_k; \mathcal{L}_n) \leq v(a_k) + G(\mathcal{N}_k, \mathcal{L}_n) \\ &\leq v^* + G(\mathcal{N}_k, \mathcal{L}_n) + \epsilon, \quad \forall k \geq k_\epsilon. \end{aligned}$$

As $v^* := \inf_{a\in\mathcal{C}} v(a)$ and $A^*_{k,n} \in \mathcal{N}_k \subset \mathcal{C}$, $v^* \leq v(A^*_{k,n}) \leq L(A^*_{k,n}; \mathcal{L}_n) + G(\mathcal{N}_k, \mathcal{L}_n)$. Now (8.2) follows from the above inequalities, and (8.3) follows from (8.2) and (8.1). Finally,

$$\left|v^* - v(A^*_{k,n})\right| \leq \left|v^* - L(A^*_{k,n}; \mathcal{L}_n)\right| + \left|v(A^*_{k,n}) - L(A^*_{k,n}; \mathcal{L}_n)\right| \leq 2G(\mathcal{N}_k, \mathcal{L}_n) + \epsilon.$$

□

By (8.1), the following holds with at least $1-\delta$ probability for every $a \in \mathcal{N}_k$,

$$\begin{aligned} \left|L(a; \widehat{\mathcal{L}}_n) - L(a; \mathcal{L}_n)\right| &\leq |v(a) - L(a; \mathcal{L}_n)| + \left|v(a) - L(a; \widehat{\mathcal{L}}_n)\right| \\ &\leq G(\mathcal{N}_k, \mathcal{L}_n) + G(\mathcal{N}_k, \widehat{\mathcal{L}}_n) \leq 2c(\mathcal{N}_k, n, \delta/2). \end{aligned}$$

This shows that the in-sample and out-of-sample performance difference of any network provides an empirical lower bound for the Rademacher complexity with high probability. In fact, in many applications it is a standard practice to monitor this difference. Thus, also in view of the estimate (8.3), the following quantity, *maximal performance difference*, could be taken as a proxy for overlearning,

$$O(\mathcal{N}_k, \mathcal{L}_n, \widehat{\mathcal{L}}_n) := \sup_{a\in\mathcal{N}_k} |L(a; \mathcal{L}_n) - L(a; \widehat{\mathcal{L}}_n)|.$$

We restate that the following holds with at least $1-\delta$ probability,

$$|L(a; \mathcal{L}_n) - L(a; \widehat{\mathcal{L}}_n)| \leq O(\mathcal{N}_k, \mathcal{L}_n, \widehat{\mathcal{L}}_n) \leq 2c(\mathcal{N}_k, n, \delta/2), \quad \forall a \in \mathcal{N}_k. \tag{8.4}$$

8.2 Convergence

In this subsection, we prove convergence under an assumption on the complexity of the hypothesis spaces $\mathcal{N}_k$. Recall $A^*_{k,n} = h(\cdot, \theta^*_{k,n})$ of (5.3).

Corollary 8.6. *Suppose that for each k, the Rademacher complexity $r(\ell(\mathcal{N}_k); n)$ converges to zero as the training size n tends to zero. Then, the following holds with probability one,*

$$\lim_{n\to\infty} L(A^*_{k,n}; \mathcal{L}_n) = \lim_{n\to\infty} L(A^*_{k,n}; \widehat{\mathcal{L}}_n) = \lim_{n\to\infty} v(A^*_{k,n}) = v^*_k := \inf_{a\in\mathcal{N}_k} v(a).$$

As discussed in Remark 8.3 above, the above assumption on the complexity is satisfied by the neural networks under natural assumptions on the coefficients. Also, in Lemma 5.3, under the density and the regularity assumptions, we have shown that v^*_k converges to v^*. Hence, the above result states that as the size of training data increases, the performance of the feedback actions constructed by $\mathcal{N}_k$ converge to the optimal value provided that the size of the hypothesis classes also tends to infinity in a controlled manner.

Proof. By the definition of G, $v^*_k \le v(A^*_{k,n}) \le L(A^*_{k,n}; \mathcal{L}_n) + G(\mathcal{N}_k, \mathcal{L}_n)$. Also, for any $a \in \mathcal{N}_k$,

$$L(A^*_{k,n}; \mathcal{L}_n) \le L(a; \mathcal{L}_n) \le v(a) + G(\mathcal{N}_k, \mathcal{L}_n).$$

We take the infimum over $a \in \mathcal{N}_k$ to conclude that $L(A^*_{k,n}; \mathcal{L}_n) \le v^*_k + G(\mathcal{N}_k, \mathcal{L}_n)$. Hence,

$$\left|v^*_k - L(A^*_{k,n}; \mathcal{L}_n)\right| \le G(\mathcal{N}_k, \mathcal{L}_n).$$

Fix $\epsilon > 0$ and set $\delta_n = 2\exp(-2n\epsilon^2/(6c^*)^2)$ so that

$$c(\mathcal{N}_k, n, \delta_n) = 2r(\ell(\mathcal{N}_k); n) + 6c^*\sqrt{\frac{\ln(2/\delta_n)}{2n}} = 2r(\ell(\mathcal{N}_k), n) + \epsilon.$$

Then, by (8.1), for every k with at least $1 - \delta_n$ probability

$$\left|v^*_k - L(A^*_{k,n}; \mathcal{L}_n)\right| \le G(\mathcal{N}_k, \mathcal{L}_n) \le c(\mathcal{N}_k, \mathcal{L}_n, \delta_n) = 2r(\ell(\mathcal{N}_k), n) + \epsilon.$$

Equivalently, $\mathbb{P}(\Omega_{k,n,\epsilon}) \le \delta_n$, where

$$\Omega_{k,n,\epsilon} := \{|v^*_k - L(A^*_{k,n}; \mathcal{L}_n)| > 2r(\ell(\mathcal{N}_k), n) + \epsilon\}.$$

Since $\sum_n \delta_n < \infty$, by the Borel–Cantelli Lemma, for every k,

$$\limsup_{n\to\infty} \left|v^*_k - L(A^*_{k,n}; \mathcal{L}_n)\right| \le \lim_{n\to\infty} 2r(\ell(\mathcal{N}_k), n) + \epsilon = \epsilon,$$

with probability one.

In view of (8.4), by at least $1 - \delta_n$ probability

$$\left|L(A^*_{k,n}; \widehat{\mathcal{L}}_n) - L(A^*_{k,n}; \mathcal{L}_n)\right| \le 2c(\mathcal{N}_k, n, \delta_n/2) = 4r(\ell(\mathcal{N}_k), n) + \ln(4)\epsilon.$$

The above Borel–Cantelli argument also implies that with probability one,

$$\limsup_{n\to\infty} \left|L(A^*_{k,n}; \widehat{\mathcal{L}}_n) - L(A^*_{k,n}; \mathcal{L}_n)\right| \le \ln(4)\epsilon.$$

□

9 Numerical Experiments

In this section we present the numerical implementations of Example 6.1. We take $\lambda = 1, r = 0$ and as discussed in that example the return of the second period $Z_2 = \zeta\eta$, where ζ is Gaussian with mean 18% and volatility 0.44%[2] and is independent of Z_1. Then, the optimal solution given in Example 6.1 is $a^* = 0.9297$ and $\text{ce}(v^*) = -a^*$. Moreover, for all parameters, $v^*_{nt} = 1$ yielding $\text{ce}(v^*_{nt}) = -\infty$. To focus the training on a compact input domain, $Z_1 \in \mathbb{R}^d$ is distributed uniformly over the d-dimensional hypercube $[-0.5, 0.5]^d$. We use the certainty equivalent ce defined in Example 6.1 to compare the performance of different actions.

As our main goal is to illustrate the potential overlearning, we try to strike a balance between avoiding unnecessary tuning parameters while still implementing commonly accepted best practices. The simple but representative structure of the chosen example allows us to easily evaluate the trained feedback actions by comparing them to explicit formulae, and also provides an understanding of the performance of this algorithm on a general class of decision problems. We emphasize that our claim is not that overlearning cannot be alleviated in these problems, but that it does occur even with a seemingly reasonable learning setup and that one has to be aware of the possibility. Indeed some degree of tuning could possibly lead to improvement in this particular example, but such methods are not systematic, and it is not clear that they generalize when the ground truth is not available. Corollary 8.6 and Lemma 5.3 show that increasing the training set (and possibly the architecture complexity in a controlled manner) does provide a systematic method for improvement. This is also observed in the computations that follow.

To describe our findings succinctly, let a^* be the (constant) optimal feedback action and $A_{k,n}$ be the feedback action computed by the neural network $\mathcal{N}_k$ on the training set $\mathcal{L}_n$. Although the optimization algorithm is trying to compute the minimizer $A^*_{k,n}$ of (5.3), in actual computations, the stochastic gradient algorithm is stopped before reaching $A^*_{k,n}$. Thus, $A_{k,n}$ depends not only on the training data $\mathcal{L}_n$ and the network $\mathcal{N}_k$ but also on the optimization procedure, in particular, the stopping rule.

By taking advantage of the explicitly available solution, we define the in-sample relative performance p_{in} and the out-of-sample relative performance p_{out} of the trained actions $A_{k,n}$ by,

$$p_{in} := \frac{\text{nn}_{\text{in-sample}} - \text{true}_{\text{in-sample}}}{\text{true}_{\text{in-sample}}},$$

$$p_{out} := \frac{\text{nn}_{\text{out-of-sample}} - \text{true}_{\text{out-of-sample}}}{\text{true}_{\text{out-of-sample}}},$$

where

$$\text{nn}_{\text{in-sample}} := \text{ce}(L(A_{k,n}; \mathcal{L}_n)), \qquad \text{nn}_{\text{out-of-sample}} := \text{ce}(L(A_{k,n}; \widehat{\mathcal{L}}_n)),$$
$$\text{true}_{\text{in-sample}} := \text{ce}(L(a^*; \mathcal{L}_n)), \qquad \text{true}_{\text{out-of-sample}} := \text{ce}(L(a^*; \widehat{\mathcal{L}}_n)),$$

and $\mathcal{L}_n$ is the training set used to compute $A_{k,n}$ and $\widehat{\mathcal{L}}_n$ is the training set chosen identically and independently of $\mathcal{L}_n$. Then, the appropriately normalized performance difference $p_{in} - p_{out}$ provides an understanding of the overlearning proxy $O(\mathcal{N}_k, \mathcal{L}_n, \widehat{\mathcal{L}}_n)$ as in (8.4). Indeed, larger values of the difference imply larger values of O.

We focus on these measures, p_{in}, p_{out}, for two reasons. Firstly, although our samples are large enough to give a good representation of the distribution, the above formulae eliminate

[2]We have chosen the mean and the volatility values randomly among those with a^* close to one and which are neither too small or large. For these parameter values, overlearning is not particularly easy.

some dependency on the sample by subtracting the true optimizers performance on each sample. Secondly, there are circumstances where seemingly the training immediately tries to interpolate data instead of first approaching the true solution before starting to interpolate, as one might expect. This leads the out-of-sample performance to increase very early on, and with our stopping rule based on the out-of-sample performance, it thus leads to almost immediate stopping. In this sense, the combination of stopping rule and performance measure is relatively conservative for measuring overlearning.

9.1 Implementation Details

Our implementation is written in the programming language OCaml [43] using the library Owl [53]. To fully reproduce the computations, the code and the logs, including the random number generator seeds, are available at `https://gitlab.com/mreppen/dderm`. Note that most neural network 'best practices' are developed for other types of problems than the control problems we study here. We still follow these practices along with common defaults so as not to color the results by specific choices. Nevertheless, we observe the same qualitative results also with alternative implementations. In all examples, the activation functions are set to ReLU and the parameters are optimized by stochastic gradient descent using the Adam scheme with parameters $(\alpha, \beta_1, \beta_2) = (0.001, 0.9, 0.999)$, as proposed by [40]. The neural networks are constructed with three hidden layers. This architecture is kept fixed regardless of data dimensionality to better isolate the dimensionality's impact on overlearning. The weights are initialized with a uniform centered distribution of width inversely proportional to the square root of the number of neuron inputs.[3]

As the neural networks are capable of overlearning the data, we must employ stopping rules for early stopping. Such stopping rules are commonly used in practice as implicit regularizers. In our studies, we mainly use a conservative stopping rule that monitors the out-of-sample performance on a separate validation set after each epoch and terminates when that out-of-sample performance exceeds its past minimum.[4] To demonstrate the potential overlearning, we also performed some experiments running the stochastic gradient without stopping for a fixed number of epochs.

We train using minibatches sampled randomly from the training set. Overlearning can also be observed with batch gradient descent—equivalent to the extreme case of setting the minibatch size to the full training set—but we have opted to default to minibatches as it is far more common and computationally efficient[5]. On the issue of minibatch size, we use the Keras default of 32.

9.2 Results

Table 1 reports the neural network's average relative in-sample performance, and its comparison to the out-of-sample test set performance with the above described conservative stopping rule.

[3]The uniform He-initializer [33]—which differs only by a factor $\sqrt{6}$ in the width of the uniform distribution and is commonly recommended for training ReLU networks—has not shown qualitatively different results with regards to overlearning.

[4]As the parameter landscape is expected to have plateaus and the out-of-sample performance is not expected to be perfectly monotone, this calls for some tolerance, thereby introducing a tuning parameter. To be conservative, we keep this tolerance small to encourage early stopping and reduce overlearning.

[5]The computational burden of each gradient computation scales as $O(N)$ in the batch size N, but the accuracy is of order $O(1/\sqrt{N})$, leading to computational advantages of small batch sizes (but not too small, due to SIMD instructions in modern CPUs and GPUs). It is sometimes argued that the more 'chaotic' nature of small batches leads to beneficial regularization. However, due to the complex interaction between the batch size and the stopping rule, the effect of this is not clear-cut.

For each dimension, the corresponding μ value is the average of 30 runs and σ is the standard deviation. We keep the data size of $N = 100,000$ and the network architecture of three hidden layers of width 10 fixed. Even though with this rule the stochastic gradient descent is stopped quite early, there is substantial overperformance increasing with dimension.

dims	p_{in} (%)		$p_{in} - p_{out}$ (%)	
	μ	σ	μ	σ
100	10.12820	1.09290	23.67080	2.01177
85	8.38061	1.35575	20.16440	2.30489
70	7.32720	0.86458	15.62060	1.94043
55	5.05783	0.81518	10.93950	1.54431
40	3.74648	0.62588	7.91105	1.32581
25	2.11501	0.43845	4.58954	0.88461
10	0.53982	0.34432	1.46138	0.39078

Table 1: Average relative in-sample performance, and its comparison to the out-of-sample performance with the above described conservative stopping rule. Everything is in % with training size of $N = 100,000$ and three hidden layers of width 10. The μ value is the average of 30 runs and σ is the standard deviation.

To isolate the impact of the dimension, in the second experiment, we keep all parameters except the width of layers as before. The last two hidden layers again have width 10. But the width of the first hidden layer is adjusted so that the number of parameters is equal to that of a neural network with three hidden layers of width 10 and input layer of dimension as in column 'parameters-equivalent'. There are three groups with parameters-equivalent dimensions of 40, 70 and 100. For example in the group with parameters-equivalent dimension 70, in the row with actual dimension 70, all layers have width 10. But in that group, the networks for the actual dimensions of 40 and 10 have wider first layer so that they all have the same number of parameters. Table 2 also shows a clear increase of overlearning with dimension. Although, the architecture is not exactly same, we believe that this experiment shows that the apparent dimensional dependence is not simply due to the increase in the number of parameters.

dims	params-equiv	p_{in} (%)		$p_{in} - p_{out}$ (%)	
		μ	σ	μ	σ
100	100	10.12820	1.09290	23.67080	2.01177
70	100	8.86214	1.45962	21.65000	3.12209
40	100	7.28550	1.19811	15.27540	2.10167
10	100	1.99793	0.54664	4.18041	1.22285
70	70	7.32720	0.86458	15.62060	1.94043
40	70	5.67500	0.84644	12.45610	1.90450
10	70	1.50328	0.93772	3.46245	1.19606
40	40	3.74648	0.62588	7.91105	1.32581
10	40	1.13566	0.65512	2.84677	0.78069

Table 2: All other parameters except the width of layers are as in Table 1. The last two hidden layers again have width 10 and the width of the first hidden layer is adjusted so that the number of parameters is equal to that of a neural network with three hidden layers of width 10 and the number of dimension is as in the parameters-equivalent column.

We have also implemented an aggressive optimization by running the algorithm for 100 and 200 epochs in 100 dimensions without a stopping rule with other parameters as in Table 1. In these experiments trained actions $A_{k,n}$ are closer to optimal actions $A^*_{k,n}$ and Theorem 7.4 predicts a larger overperformance. Indeed, Table 3 shows that, overlearning is quite substantial even with a training size of $100,000$ and there is a noticeable deterioration in the out-of-sample performance.

epochs	p_{in} (%)		$p_{in} - p_{out}$ (%)	
	μ	σ	μ	σ
200	30.3161	2.46850	315.875	540.4750
100	25.8374	1.72027	111.553	41.5841

Table 3: Longer iterations performance in 100 dimensions. Based on 15 runs. Especially the 200 epoch runs show signs of a heavy tail, as expected with high degrees of overlearning. All other parameters as in Table 1.

Finally, Table 4 illustrates the convergence proved in Section 8.2. In 100 dimensions we increase the size of the training data from $100,000$ to twenty-fold keeping all the other parameters as in the first experiment. The results show a remarkable improvement in the accuracy demonstrating the power of dynamic deep empirical risk minimization.

sample size	dims	p_{in} (%)		$p_{in} - p_{out}$ (%)	
		μ	σ	μ	σ
2,000,000	100	0.49597	0.11849	1.12846	0.32184
1,000,000	100	1.14094	0.14640	2.39532	0.27415
500,000	100	2.36352	0.20154	5.24018	0.81235
250,000	100	4.41388	0.37928	10.02040	1.45355

Table 4: Performance for larger sample sizes. Based on 15 runs. All other parameters as in Table 1.

10 Conclusions

Dynamic deep empirical risk minimization is a highly effective computational tool for many investment or hedging problems, or more generally, decision making under uncertainty. It can handle general random structures in high dimensions and complex dynamics with ease. The simplicity of the algorithm and the recent advances in the training of deep neural networks are key to these properties. By both theoretical results and numerical experiments, we have demonstrated that the hypothesis spaces can overlearn the data, and consequently, construct forward-looking feedback actions. As shown in Theorem 7.1, the optimization step is able to by-pass the adaptedness requirement, and the trained actions may become non-adapted to the flow of information on the training data. Thus, in-sample-value approximates the strictly smaller minimum given by the anticipative controls.

The estimates proved in Theorem 8.5 show that overlearning is negligible when the data set is sufficiently large compared to the complexity of the networks. As one needs sufficient complexity of the neural networks to achieve appropriate accuracy, the size of the training set is critical. When a particular model is assumed, dynamic simulation during the training is an effective way

of increasing the size of the data. Without such a model and small size data, one must design simulation mechanisms based on the given data and use it dynamically during the optimization. Therefore, for an efficient application of these methods in data-driven envoroinments, calibration of complex models to the market data is a crucial step.

Although this approach is particularly valuable in high dimensions, overlearning becomes easier for those problems, thus requiring richer training sets, as clearly demonstrated by the numerical studies reported in Section 9. For optimal control, an in-depth study of this dependence, both numerically and theoretically, remains an interesting question. The proof of the asymptotic overlearning result Theorem 7.4 provides an initial insight indicating that the average distance between the data points and the regularity of the networks are important for a better understanding of this dependence. Indeed, the Rademacher complexity which is present in the upper bound (8.3) is also influenced by both of them. Closely related covering numbers providing an upper for the Rademacher complexity [cf. 51, Lemma 27.4] could also be useful in understanding this dependence.

Acknowledgments. The authors thank Professors E, Fan and Han of Princeton University, and two anonymous referees for their valuable comments.

References

[1] S. A. Assefa, D. Dervovic, M. Mahfouz, R. E. Tillman, P. Reddy, and M. Veloso. Generating synthetic data in finance: opportunities, challenges and pitfalls. In *Proceedings of the First ACM International Conference on AI in Finance*, pages 1–8, 2020.

[2] A. Bachouch, C. Huré, N. Langrené, and H. Pham. Deep neural networks algorithms for stochastic control problems on finite horizon, part 2: Numerical applications. *arXiv preprint:1812.05916*, 2018.

[3] J. Backhoff, D. Bartl, M. Beiglböck, and J. Wiesel. Estimating processes in adapted Wasserstein distance. *The Annals of Applied Probability*, 32(1):529–550, 2022.

[4] D. Bartl and S. Mendelson. On Monte-Carlo methods in convex stochastic optimization. *arXiv preprint: 2101.07794*, 2021.

[5] D. Bartl, S. Drapeau, and L. Tangpi. Computational aspects of robust optimized certainty equivalents and option pricing. *Mathematical Finance*, 30(1):287–309, 2020.

[6] P. L. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

[7] P. L. Bartlett, S. Boucheron, and G. Lugosi. Model selection and error estimation. *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory*, pages 286–297, 2000.

[8] P. L. Bartlett, S. Boucheron, and G. Lugosi. Model selection and error estimation. *Machine Learning*, 48(1):85–113, 2002.

[9] S. Becker, P. Cheridito, and A. Jentzen. Deep optimal stopping. *Journal of Machine Learning Research*, 20(4):1–25, 2019.

[10] S. Becker, P. Cheridito, A. Jentzen, and T. Welti. Solving high-dimensional optimal stopping problems using deep learning. *arXiv preprint:1908.01602*, 2019.

[11] M. Belkin, D. Hsu, S. Ma, and S. Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

[12] D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-dynamic programming*. Athena Scientific, 1996.

[13] D. Bertsimas, S. Shtern, and B. Sturt. A data-driven approach for multi-stage linear optimization. *Optimization Online*, 2018.

[14] D. Bertsimas, S. Shtern, and B. Sturt. Two-stage sample robust optimization. *arXiv preprint:1907.07142*, 2019.

[15] S. Boucheron, O. Bousquet, and G. Lugosi. Theory of classification: A survey of some recent advances. *ESAIM: probability and statistics*, 9:323–375, 2005.

[16] O. Bousquet, S. Boucheron, and G. Lugosi. Introduction to statistical learning theory. In *Summer School on Machine Learning*, pages 169–207. Springer, 2003.

[17] H. Buehler, L. Gonon, J. Teichmann, and B. Wood. Deep hedging. *Quantitative Finance*, 19 (8):1271–1291, 2019.

[18] H. Buehler, L. Gonon, J. Teichmann, B. Wood, and B. Mohan. Deep hedging: hedging derivatives under generic market frictions using reinforcement learning. Technical report, Swiss Finance Institute, 2019.

[19] Q. Chan-Wai-Nam, J. Mikael, and X. Warin. Machine learning for semi linear pdes. *Journal of Scientific Computing*, 79(3):1667–1712, 2019.

[20] C. Cuchiero, W. Khosrawi, and J. Teichmann. A generative adversarial network approach to calibration of local stochastic volatility models. *Risks*, 8(4):101, 2020.

[21] G. Cybenko. Approximations by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:183–192, 1989.

[22] P. Dupuis and H. J. Kushner. *Numerical methods for stochastic control problems in continuous time*, volume 24. Springer Science & Business Media, 2001.

[23] P. M. Esfahani and D. Kuhn. Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1-2):115–166, 2018.

[24] S. Fecamp, J. Mikael, and X. Warin. Deep learning for discrete-time hedging in incomplete markets. *Journal of computational Finance*, 2020.

[25] W. H. Fleming and H. M. Soner. *Controlled Markov processes and viscosity solutions*, volume 25. Springer Science & Business Media, 2006.

[26] N. Fournier and A. Guillin. On the rate of convergence in Wasserstein distance of the empirical measure. *Probability Theory and Related Fields*, 162(3):707–738, 2015.

[27] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias-variance dilemma. *Neural computation*, 4(1):1–58, 1992.

[28] N. Golowich, A. Rakhlin, and O. Shamir. Size-independent sample complexity of neural networks. *Information and Inference: A Journal of the IMA*, 9(2):473–504, 2020.

[29] L. Gonon, J. Muhle-Karbe, and X. Shi. Asset pricing with general transaction costs: Theory and numerics. *arXiv preprint:1905.05027*, 2019.

[30] J. Han and W. E. Deep learning approximation for stochastic control problems. In *Deep Reinforcement Learning Workshop, NIPS*, 2016.

[31] J. Han and J. Long. Convergence of the deep BSDE method for coupled FBSDEs. *arXiv preprint:1811.01165*, 2018.

[32] J. Han, A. Jentzen, and W. E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, 2018.

[33] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[34] P. Henry-Labordère. Deep primal-dual algorithm for BSDEs: Applications of machine learning to CVA and IM. *SSRN 3071506*, 2017.

[35] K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

[36] C. Huré, H. Pham, A. Bachouch, and N. Langrené. Deep neural networks algorithms for stochastic control problems on finite horizon, part I: convergence analysis. *arXiv preprint:1812.04300*, 2018.

[37] C. Huré, H. Pham, and X. Warin. Some machine learning schemes for high-dimensional nonlinear pdes. *arXiv preprint:1902.01599*, 33, 2019.

[38] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

[39] S. M. Kakade and A. Tewari. On the generalization ability of online strongly convex programming algorithms. In *Advances in Neural Information Processing Systems*, pages 801–808, 2009.

[40] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint:1412.6980*, 2014.

[41] V. Koltchinskii and D. Panchenko. Empirical margin distributions and bounding the generalization error of combined classifiers. *Annals of statistics*, 30(1):1–50, 2002.

[42] E. L. Lehmann and G. Casella. *Theory of point estimation*. Springer Science & Business Media, 2006.

[43] X. Leroy, D. Doligez, A. Frisch, J. Garrigue, D. Rémy, and J. Vouillon. *The OCaml system release 4.10*, 2 2020. https://caml.inria.fr/pub/docs/manual-ocaml/.

[44] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[45] B. Neyshabur, R. Tomioka, and N. Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pages 1376–1401, 2015.

[46] G. C. Pflug and A. Pichler. From empirical observations to tree models for stochastic optimization: convergence properties. *SIAM Journal on Optimization*, 26(3):1715–1740, 2016.

[47] A. M. Reppen, H. M. Soner, and V. Tissot-Daguette. Neural optimal stopping boundary. *arXiv:2205.04595*, 2022.

[48] A. M. Reppen, H. M. Soner, and V. Tissot-Daguette. Deep stochastic optimization in finance. *arXiv:2205.04604*, 2022.

[49] J. Ruf and W. Wang. Hedging with neural networks. *arXiv preprint: 2004.08891*, 2020.

[50] J. Ruf and W. Wang. Neural networks for option pricing and hedging: a literature review. *Journal of Computational Finance*, 24(1), 2020.

[51] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[52] J. Sirignano and K. Spiliopoulos. Dgm: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375:1339–1364, 2018.

[53] L. Wang. Owl: A general-purpose numerical library in OCaml. *arXiv preprint:1707.09616*, 2018.

[54] M. M. Wolf. *Mathematical foundations of supervised learning*. Lecture notes, Technical University of Munich., 2018.

[55] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. *arXiv:1611.03530*, 2017. International Conference on Learning Representations.

A Asymptotic Overlearning

In this section, we prove an extension of Theorem 7.4 proved for distinct training data. Although this is a natural assumption which holds for instance, when $Z^{(i)}$ are drawn independently from an atomless distribution, we provide this extension to further facilitate our understanding of overlearning.

Fix a training set $\mathcal{L}_n$. Let $\mathcal{K} = \{\mathcal{K}^{(1)}, \dots, \mathcal{K}^{(m)}\}$ be a partition of $\mathcal{L}_n$ satisfying:

- $\mathcal{K}^{(j)}$ are disjoint subsets of $\mathcal{L}_n$;
- $\cup_j \mathcal{K}^{(j)} = \mathcal{L}_n$;
- if $z \in \mathcal{K}^{(j)}$ for some j, and if there is a trajectory $\hat{z} \in \mathcal{L}_n$ and $t \in \mathcal{T}$ such that $z_t = \hat{z}_t$, then $\hat{z} \in \mathcal{K}^{(j)}$.

There are partitions satisfying the above conditions and one can even define and would like to use the maximal partition satisfying above conditions. As this is tangential to the main thrust of the paper, we do not pursue it here. When the data is distinct, the maximal partition is $\mathcal{K}^{(i)} = \{Z^{(i)}\}$ and we are back in the setting of Theorem 7.4.

For a constant control $\alpha = (\alpha_0, \dots, \alpha_{T-1}) \in \mathcal{A}^{\mathcal{T}}$, and for $j = 1, \dots, m$, define

$$\overline{\ell}(\alpha, j) := \frac{1}{\left|\mathcal{K}^{(j)}\right|} \sum_{z \in \mathcal{K}^{(j)}} \ell(\alpha, z).$$

Let $\alpha^\epsilon(j) \in \mathcal{A}^{\mathcal{T}}$ be an ϵ-minimizer of $\overline{\ell}(\cdot, j)$. Analogously to L_n^* defined in (7.3), define

$$\overline{L}_n^* := \frac{1}{m} \sum_{j=1}^{m} \inf_{\alpha \in \mathcal{A}^{\mathcal{T}}} \overline{\ell}(\alpha, j) = \lim_{\epsilon \downarrow 0} \frac{1}{m} \sum_{j=1}^{m} \overline{\ell}(\alpha^\epsilon(j), j).$$

For $z \in \mathcal{L}_n$, let $j(z)$ be the unique index so that $z \in \mathcal{K}^{(j(z))}$. We now follow the arguments of Theorem 7.4 *mutadis mutandis* to show that the hypothesis spaces can approximate the function

$$\overline{a}_\epsilon^*(z) := \alpha^\epsilon(j(z)), \quad z \in \mathcal{L}_n.$$

This implies the following extension of the overlearning result Theorem 7.4.

Lemma A.1. *Let $A_{k,n}^*$ be as in* (5.3)*. Under the density assumption* (*c.f. Assumption 5.1*)*,*

$$\lim_{k \to \infty} L(A_{k,n}^*; \mathcal{L}_n) \leq \overline{L}_n^*.$$

When the partition $\mathcal{K}$ of $\mathcal{L}_n$ is non-trivial and if the number of partitions m is large, then we may have $\overline{L}_n^* < v^*$ and consequently potential overlearning. The robust approach used in [13, 14] and also in [23, 5], essentially groups the elements of the training set into a small number of sets and identifies them by a representative element of these sets. If we then partition this processed data, this would result in a small number of partitions and the overlearning will not be possible even with modest size training sets.

B Markov Decision Processes

An essential structural requirement of our formulation is that the random process Z driving the state dynamics is independent of control. Although many control problems may not be initially expressed that way, if their dynamics is known, they could still be reformulated to fit into our framework. Here we discuss one such central example of controlled Markov chains to illustrate this point.

Let $\{Y_t\}_{t \in \mathcal{T}}$ be a $\mathcal{S} := \{s_1, \dots, s_M\}$ valued controlled Markov chain. Suppose that the transition probabilities are given by

$$p(t, y, \tilde{y}, a) := \mathbb{P}(Y_{t+1} = y \,|\, Y_t = \tilde{y},\ A_t = a), \qquad y, \tilde{y} \in \mathcal{S},\ a \in \mathcal{A},\ t \in \mathcal{T}.$$

For a given Y_0 and $\widehat{\Phi}$, the classical control problem is to

$$\text{minimize} \quad \mathbb{E}[\widehat{\Phi}(Y_1, \dots, Y_T, A_0, \dots, A_{T-1})]$$

over all feedback controls A.

To reformulate this problem, we first introduce a random process $Z = (Z_1, \dots, Z_T)$ satisfying,

$$\mathbb{P}(Z_t = y) = \frac{1}{M}, \qquad \forall\ y \in \mathcal{S},\ t = 1, \dots, T.$$

Let $X_0 = (Y_0, 1)$, $\mathcal{X} := \mathcal{S} \times [0,1]$ and for $t = 1, \dots, T$, set $X_t =: (X_t^{(1)}, X_t^{(2)}) = (Z_t, R_t)$, where the second component R is the Radon–Nikodym process defined recursively by,

$$R_0 = 1, \quad \text{and} \quad R_{t+1} = M\, p(t, Z_{t+1}, X_t^{(1)}, A_t)\ R_t, \qquad t \in \mathcal{T}.$$

We rewrite the equations for the components of X as

$$X_{t+1} = \left(Z_{t+1},\ M\, p(Z_{t+1}, X_t^{(1)}, A_t)\, X_t^{(2)} \right),$$

verifying that the dynamics of X is in the form (3.2). Also, it can be directly shown that

$$\mathbb{E}[\widehat{\Phi}(Y_1, \dots, Y_T, A_0, \dots, A_{T-1})] = \mathbb{E}[\widehat{\Phi}(X_1^{(1)}, \dots, X_T^{(1)}, A_0, \dots, A_{T-1})\, X_T^{(2)}].$$

Hence, the original problem is equivalent to the control of the process X with

$$\Phi(X, A, Z) := \widehat{\Phi}(X_1^{(1)}, \dots, X_T^{(1)}, A_0, \dots, A_{T-1})\, X_T^{(2)}.$$

Remark B.1. Randomized controls that are widely used in problems with learning can also be included in our framework. However, we should note that we are primarily interested in constructing the optimal feedbacks and in our framework they always exist and randomization is not needed.

Consider the above optimization problem with the controlled Markov chain $Y \in \mathcal{S}$ and a finite control set $\mathcal{A} = \{a^1, \dots, a^n\}$. A randomized control U_t at time t is a probability on the control set $\mathcal{A}$. Thus, it takes values in the simplex,

$$\Delta_{n-1} := \{u = (u^1, \dots, u^n) \in \mathbb{R}_+^n \mid u^1 + \ldots + u^n = 1\ \}.$$

The components of $U_t = (U_t^1, \dots, U_t^n) \in \Delta_{n-1}$ correspond to the probability that a particular control is used in that step, i.e., $U_t^i = \mathbb{P}(A_t = a^i)$. Let $p(y, \tilde{y}, a^i)$ be the transition probabilities of Y_t when control a^i is chosen. For a given $U = (U^1, \dots, U^n) \in \Sigma_{n-1}$, consider the transition probabilities for the pair $X_t := (Y_t, a_t) \in \mathcal{S} \times \mathcal{A} =: \mathcal{X}$,

$$\begin{aligned} \widehat{p}\,((y, a^i), (\tilde{y}, a^j), U) &:= \mathbb{P}(Y_{t+1} = y,\, a_{t+1} = a^i \,|\, Y_t = \tilde{y},\, U_t = U) \\ &= p(y, \tilde{y}, a^i)\, U^i, \qquad \text{for } y, \tilde{y} \in \mathcal{S},\ i, j = 1, \dots, n. \end{aligned}$$

So if the process X_t is at the state $(\tilde{y}, a^j)$ and the randomized control U is chosen, then a new control a^i is chosen using the probability U. Then, the transition of the Y component is decided by the transition distribution $p(\cdot, \tilde{y}, a^i)$. In particular, the control a^j chosen a step earlier does not impact the dynamics of this step. This construction turns the randomized controls into a standard controlled Markov chain with an enlarged state space $\mathcal{X} = \mathcal{S} \times \mathcal{A}$ and a control set Δ_{n-1}. One can then use procedure outlined above with the Radon–Nikodym process to reformulate it with an uncontrolled random process.