Overview of (some) machine learning methods in stochastic control

Huyên PHAM

Ateliers FIME
EDF lab Paris-Saclay, 22-23 september 2021

Basic framework of (stochastic) control

- *State variables*: process $X = (X_t)_t$, $t \in \mathbb{T}$ (discrete or continuous time) valued in $\mathcal{X}$ (e.g. $\mathbb{R}^d$)

- *Control/decision variables*: process $\alpha = (\alpha_t)_t$ valued in action space $A \subset \mathbb{R}^m$
  - Markov setting: $\alpha_t = \pi(t, X_t)$ for some **policy** π (feedback control)

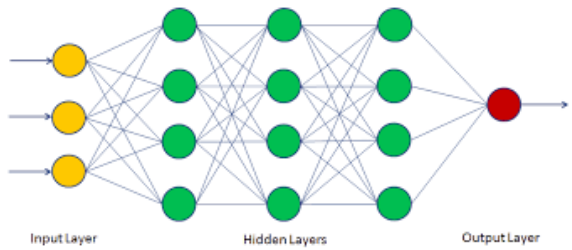
- *Performance criterion*: maximize over α (or π) a functional (in finite horizon)

$$J(\alpha) \quad = \quad \mathbb{E}\big[\sum_{t\in\mathbb{T}} f(X_t^\alpha, \alpha_t) + g(X_T^\alpha)\big]$$

$\rightarrow$ **Value function**: $V(t,x)$, $t \in \mathbb{T}$, $x \in \mathcal{X}$.

Basic principles of deep learning for stochastic control

- Aim: approximate the policy and/or value function, i.e. a function $\vartheta : \mathcal{X} \to \mathcal{Y}$
- Class of approximate functions: (deep) neural networks (NN in short)
  - A highly parametric class of functions with expressive power: break the curse of dimensionality



- Train the DNN to learn the map: $x \in \mathcal{X} \mapsto \vartheta(x) \in \mathcal{Y}$
  - Minimize some loss function $\leftrightarrow$ objective, criterion of the control problem
  - (Stochastic) gradient descent based on simulations (input data on $\mathcal{X}$) $\leftrightarrow$ underlying model/structure of the problem
- Extensive literature: algorithms differ according to the choice of
  - Loss functions
  - Training simulations
  - Global vs local approach

Global approach

- Approximate policy by a DNN $(t,x) \mapsto \phi^\theta(t,x)$ of parameter θ
- Maximize over θ the criterion

$$\tilde{J}(\theta) \quad = \quad \mathbb{E}\Big[\sum_{t\in\mathbb{T}} f(X_t^\theta, \phi^\theta(t, X_t^\theta)) + g(X_T^\theta)\Big]$$

▶ Stochastic gradient ascent algorithm for computing $\hat{\theta}$ based on simulations of X.

Gobet, Munos (05), Han, E, Jentzen (18-20)

Remarks on global approach

• Not a new idea but revitalized thanks to computational power and open-access of optimization libraries (TensorFlow, Keras, etc)

• "Deep Hedging" paper (Bühler, Teichmann) pushes forward this idea:
  - Flexibility in the choice of criteria, of the state/control process
  - Take into account market constraints: liquidity, transaction costs
  - Handle high dimension: number of assets, factors,

• Mathematical techniques and methods of stochastic control are no more required: change of paradigm or back to the 60'?

• Few theoretical analysis for justifying successful empirical results of global approach

Local approach

- Combine methods from stochastic control:
  - dynamic programming (DP)
  - Bellman PDE and Backward SDE (in continuous time)
  - Numerical probability
- Together with neural networks techniques

▶ Design algorithms for solving stochastic control problems ($\leftrightarrow$ Deep reinforcement learning)

Algo Hybrid (actor/critic) for Markov Decision Process

Learn sequentially both value function (critic) and optimal policy (actor)

- **Backward induction from DP**:
  - Initialize $\hat{V}_T = g$
  - For $t = T-1, \ldots, 0$

    (i) Compute the approximated optimal policy at time t by DNN: $\hat{\pi}_t = \phi^{\hat{\theta}_t}(.)$,

$$\hat{\theta}_t \quad \in \quad \arg\max_{\theta} \mathbb{E}\Big[f(X_t, \phi^{\theta}(X_t)) + \hat{V}_{t+1}(X_{t+1}^{\theta})\Big],$$

    where $X_t \sim \mu_t$ (training distribution)

    (ii) Update value function at time t by DNN: $\hat{V}_t = \Phi^{\hat{\beta}_t}(.)$,

$$\hat{\beta}_t \quad \in \quad \arg\min_{\beta} \mathbb{E}\Big|f(X_t, \hat{\pi}_t(X_t)) + \hat{V}_{t+1}(X_{t+1}^{\hat{\theta}_t}) - \Phi^{\beta}(X_t)\Big|^2$$

Huré, P., Bachouch, Langrené (18)

Deep Backward DP algo for continuous-time SC

• A continuous-time version of Algo Hybrid relying on Backward SDE representation of Bellman (dynamic programming) equation

• Learn simultaneously by backward induction (DP) the value function and its gradient ($\leftrightarrow$ optimal control) by minimizing loss functions $\leftrightarrow$ BSDE scheme (Huré, P. , Warin 19)

• The case of controlled volatility (fully non-linear HJB) is more challenging: need to learn accurately Hessian.
(Germain, P., Warin 20)

Comparison local vs global approach

- Sequence of loss functions vs one global objective function
- Global approach applies for general criteria (time inconsistent, CVaR, etc) while local approach relies on DP (time consistent problem)
- Local approach computes directly both value function and optimal control
- Error validation is easy to check on local approach
- Local approach is more appropriate for error analysis: (Germain, P., Warin 21)
  - Rate of convergence w.r.t. number of layers and neurons of NN
  - GroupSort networks (Lipschitz property)

- Machine learning approaches for control require a lot of simulations:
  - Historical data are not enough: need to generate simulated data close to real data. Important literature on this topic, see talk by C. Remlinger.

- Mean-field control (infinite dimension), see talk by M. Germain

- Crucial to design NN architecture that fits well the considered control problems:
  - Recent architectures of symmetric NN (DeepSet and their extensions) to deal with symmetric problems
  - Convex NN, see Lemaire, Pagès, Warin (21)
  - DeepOnet or Fourier Neural Operator dealing with functional input: suitable for robust control problem