Session 3.4

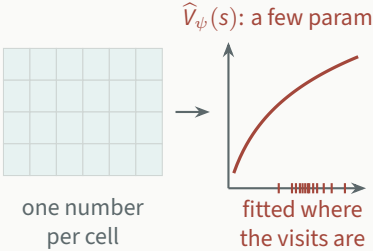
From Tables to Networks: Function Approximation, Policy Gradient, Actor–Critic



Zhigang Feng

© Zhigang Feng Redistribution prohibited without permission

Where a Table Stops Fitting: Store a Function Instead



- Decks 3.2–3.3 kept a table, one number per cell. **Deck 3.1's two learners store two functions instead: how?**
- In the lab: the rule a sigmoid of a line in $\ln k$, shifted for z, **three numbers**; the value a quadratic in $\ln k$, its level and slope shifted for z, **five numbers**.
- **Function approximation**: far fewer parameters than states, fitted where samples fall; a table is the indicator case.
- **Two routes: a basis an economist picks, then features a network learns. Then the rule carries the $\max$, in steps.**

Source: Session 3 lab, block B.

© Zhigang Feng
Redistribution prohibited without permission

Session 3.4

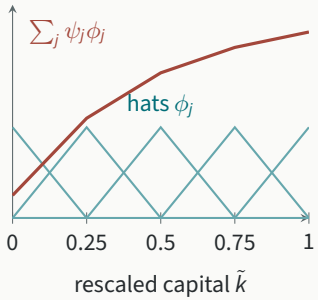
Route One: Linear in Parameters

A basis the economist picks: what it is, what dimension costs, what it buys and what it costs



© Zhigang Feng · Redistribution prohibited without permission

Linear in the Parameters, Not in the State



- **Yes: linear in the weights ψ, not in k.** $\widehat{V}_\psi(s) = \sum_j \psi_j \, \phi_j(s)$, each **basis function** ϕ_j a fixed curve; a table is the indicator basis.
- The gradient is $\phi(s)$: one update moves every state sharing a feature (p. 5).
- On the grid: $\phi(k, z) = (1, \tilde{k}, \tilde{k}^2, \mathbf{1}\{z = 1.1\}, \tilde{k}\,\mathbf{1}\{z = 1.1\})$, $\tilde{k}$ capital rescaled to $[0, 1]$.
- A smooth, concave value: polynomials in $\ln k$. A kink, as at a borrowing limit: local pieces.
- **The economist chooses the ϕ_j, so must guess the value's shape. Next: what that buys, what it costs.**

© Zhigang Feng
Redistribution prohibited without permission

Route One: What It Buys, What It Costs, and Why Route Two

	a basis you pick: this route	**features learned**: Route Two
you choose	the curves ϕ_j; they must contain the value's shape ($\ln k$ terms for log utility)	an architecture; its hidden layers are the features
one update	$\psi \leftarrow \psi + \eta\, \delta_t\, \phi(s_t)$, δ_t the TD error, the target held fixed	a gradient step; a frozen copy holds the target (p. 8)
many state variables	all products of the 1-D curves: m^d weights; sparse grids thin them	the first layer grows with d, not m^d: Deck 3.1's First Answer
guaranteed	on the rule's own data: error within a constant of the best fit's	no general result
can go wrong	the **deadly triad**: a fitted value, a self-made target and another rule's data can run away together	the same; Q-learning on a network has all three

- **Why Route Two: a network learns the features a basis must guess, model by model, and can take a histogram as one input (Deck 3.5). The price is the guarantee.**

© Zhigang Feng Redistribution prohibited without permission

Session 3.4

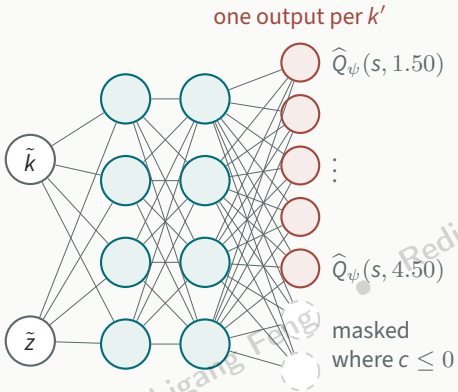
Route Two: Learn the Features

A network for Q; then a rule of its own, which carries the $\max$ in gradient steps



© Zhigang Feng
Redistribution prohibited without permission

Let the Features Be Learned: a Network for Q

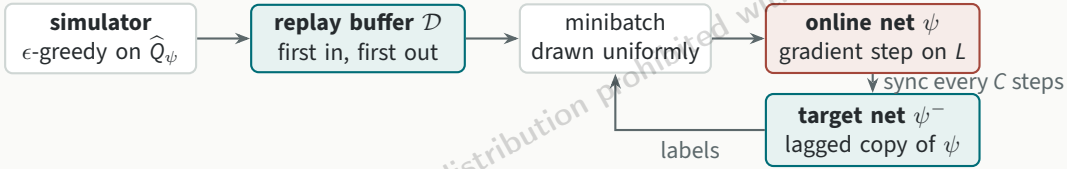


- **Why one output per k'**: Q-learning's target needs $\max_{a'} \widehat{Q}(s', a')$; with an output per k', one pass returns Deck 3.3's whole row.
- With k' as an input, each candidate costs a pass and the $\max$ is a search. So DQN suits a few discrete actions; a share has no row (p. 9).
- The hidden layers *are* the basis, now trained on the data instead of guessed. Weights ψ; θ is kept for the rule.
- The triad of p. 5 is complete: a fitted value, a self-made target, exploring data. **DQN adds two devices that hold the regression still.**

© Zhigang Feng — Redistribution prohibited without permission

DQN: a Target Network and a Replay Buffer

- **The question.** Each step on ψ moves $\widehat{Q}$ everywhere, the label included, and successive transitions are alike. What holds the regression still?



$$L(\psi) = E_{(s,a,r,s')\sim\mathcal{D}}\left[\left(r + \gamma \max_{a'\in\mathcal{A}(s')} \widehat{Q}_{\psi^-}(s',a') - \widehat{Q}_\psi(s,a)\right)^2\right].$$

- **Target network**: a frozen copy ψ^- holds the labels still for C steps.
- **Experience replay**: minibatches from stored transitions mix times and places.
- **Neither removes the triad: the loss and the rule still swing. Keep the best iterate, found by evaluating, not the last.**

Source: Session 3 lab, block F.

© Zhigang Feng Redistribution prohibited without permission

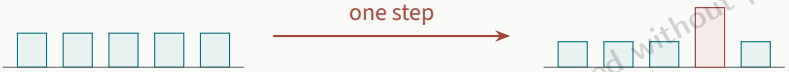
No Row to Maximise: Improve the Rule Itself, from Samples

- **The question.** DQN maximises over a row; a saving share has none, and $\arg\max_a \widehat{Q}_\psi$ can jump on a small change in ψ. How else can a rule improve?
- **The answer: make the rule a distribution** $g_\theta(a \mid s)$, smooth in θ. It tries actions; those that did better than expected become likelier.
- Why random, when p. 15 moves a share along a derivative? Without the model, a better action shows only if it is tried: Deck 3.2's coverage.
- And the derivative then runs through the rule's own odds, which we know, not through the environment, which we do not.
- On the grid: a probability on each feasible k'. For a share: p. 2's sigmoid share, with a tremble added.
- **Q-learning fits a value, then maximises. A policy gradient climbs V^{g_θ} directly: the rule carries the $\max$, one step at a time.**

© Zhengyang Jiang. Redistribution prohibited without permission

REINFORCE: Try, Score, Reinforce

$g_\theta(k' \mid s)$ **over the feasible k': uniform** **after one path: the k' taken (red) did better than expected**

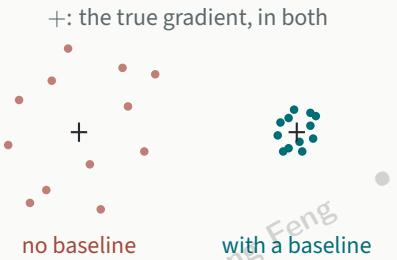


- **Try**: simulate T periods under the current random rule g_θ; its randomness explores.
- **Score**: the **score** $\nabla_\theta \ln g_\theta(a_t \mid s_t)$ points where a_t becomes likelier: $+(1-g)$ on the action taken, $-g$ on each other.
- **Reinforce**: weight each score by what followed, $G_t = \sum_{j\geq 0} \gamma^j r_{t+j+1}$ (credit assignment, Deck 3.3), less a **baseline** $b(s_t)$ of the state alone: $\theta \leftarrow \theta + \eta \sum_t \gamma^t \nabla_\theta \ln g_\theta(a_t \mid s_t)\big(G_t - b(s_t)\big)$.
- On average the step is the exact uphill direction of the rule's lifetime value; nature's odds carry no θ, so only the rule's own odds are differentiated, never the model.
- **One gradient step up the log-likelihood of your own choices, weighted by what followed. No model, no value function.**

© Zhiyang Feng — Redistribution prohibited without permission

REINFORCE with a Baseline

$$\theta \leftarrow \theta + \eta \sum_t \gamma^t \nabla_\theta \ln g_\theta(a_t \mid s_t) \left(G_t - \widehat{V}_\psi(s_t)\right), \qquad \widehat{V}_\psi \text{ moved toward the returns } G_t$$



- **The question.** Without b, every action tried in a good state looks good, lucky or not. Judge against what?
- Against the state's usual payoff $\widehat{V}_\psi(s)$, fitted to the same returns: what is left is better or worse than expected, the **advantage**.
- A function of the state alone leaves the mean step unchanged: a control variate.
- **Same average direction, far less noise. This is the critic's first job (p. 24).**

© Zhigang Feng
Redistribution prohibited without permission

Session 3.4

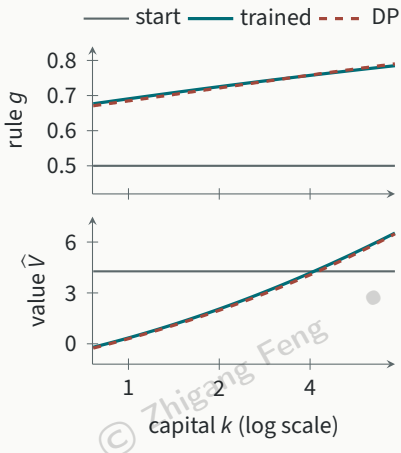
Actor–Critic

Deck 3.1’s two learners, step by step: with the model in hand, then without it, then on networks



© Zhigang Feng · Redistribution prohibited without permission

Two Functions: a Rule and a Value



© Zhigang Feng
Redistribution prohibited without permission

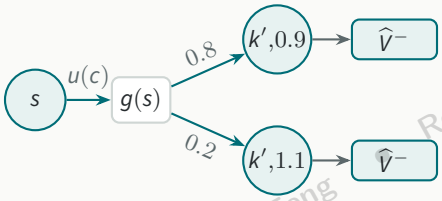
- **The question.** Model in hand, can two functions and two steps find DP's rule with no grid, no sweep and no search over actions?
- The **rule** $g(s)$: the saving share at $s = (k, z)$, so $c = (1 - g)\, y > 0$. The **value** $\widehat{V}(s)$: what s is worth under the current rule.
- Each is a function to evaluate and nudge anywhere: Route One's forms, p. 2.
- The rule is deterministic: one share per state.
- **Trained: the rule on DP's curve, the value close to** V^* (why: valuing a fixed rule is Deck 3.1's contraction T^g).

Source: Session 3 lab, block B; $z = 0.9$; trained: median of 20 seeds; how they are stored: p. 2.

The Critic's Step: Move the Value Toward a One-Period Backup

$$B(s) = u(c) + \gamma \sum_{z'} P_z(z' \mid z)\, \widehat{V}^{-}(k', z'), \quad c = (1 - g(s))\, y, \quad k' = g(s)\, y$$

$$\widehat{V} \leftarrow \arg\min_{\widehat{V} \in \mathcal{F}} \sum_{i=1}^{N} \left(\widehat{V}(s_i) - B(s_i)\right)^2$$



at $z = 0.9$: both branches weighted

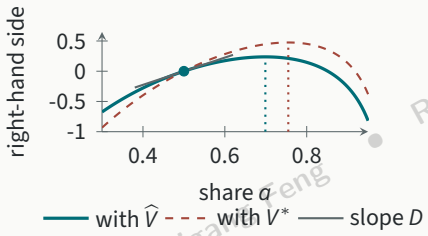
- $B(s)$: Deck 3.1's one-period backup (*Challenge 2: Why Look Further Ahead*) at one state; $\widehat{V}^{-}$: the value as the iteration began, held fixed.
- $\mathcal{F}$: what the critic can store. A table copies B into cells; any other form takes the closest fit (p. 5).
- First a **prefit**: 200 iterations with the rule frozen, so the critic starts by valuing $g \equiv \frac{1}{2}$ right.
- T^{g} **(Deck 3.1, *Two Value Functions*) applied once, exactly, then written into the value.**

© Zhigang Feng
Redistribution prohibited without permission

The Actor's Step: Which Way to Move the Share Is a Derivative

$$\max_g J(g) = \tfrac{1}{N}\sum_i \Big[u\big((1-g(s_i))\,y_i\big) + \gamma \sum_{z'} P_z(z' \mid z_i)\,\widehat{V}\big(g(s_i)\,y_i,\, z'\big)\Big], \quad \widehat{V} \text{ held fixed}$$

$$D(s) = \partial_a[\,\cdot\,]_{a=g(s)} = y\Big[-u'(c) + \gamma \sum_{z'} P_z(z' \mid z)\,\partial_k \widehat{V}(k', z')\Big]$$



- One small step uphill in J; $D(s_i)$ is its slope at s_i (the sigmoid of p. 2 keeps g in $(0, 1)$).
- A **pathwise gradient**: differentiate through the budget constraint, into u and the critic.
- Peak at 0.699: Howard's improvement of the flat rule. DP's: 0.755.
- **Still a $\max$, taken in steps**: one step uphill, not Howard's jump to the peak at 0.699.

Source: Session 3 lab, block B; $k = 3.75$, $z = 0.9$; curves shifted to zero at $a = \frac{1}{2}$.

© Zhigang Feng
Redistribution prohibited without permission

The Whole Algorithm on One Page

Actor–critic with the model in hand, on drawn states

1 initialise $g \equiv \frac{1}{2}$ and $\widehat{V} \equiv u(c^*)/(1-\gamma) = 4.27$ (c^*: steady-state consumption)
2 **prefit**: repeat line 5 with the rule frozen, 200 times
3 **repeat**
4 draw $N = 256$ states, uniform in $\ln k$ over the band and over $z \in \{0.9, 1.1\}$
5 **critic**: $B(s_i) \leftarrow u(c_i) + \gamma \sum_{z'} P_z(z' \mid z_i)\, \widehat{V}^{-}(k_i', z')$; fit $\widehat{V}$ to B at the s_i
6 **actor**: $D(s_i) \leftarrow y_i \big[-u'(c_i) + \gamma \sum_{z'} P_z(z' \mid z_i)\, \partial_k \widehat{V}(k_i', z') \big]$; one step uphill in J
7 **until** the rule stops moving (1000 iterations here); grade g against the key

- Lines 5 and 6 read u, f and P_z: the model, whole. Nothing is drawn but the states.
- Every line is written on the two functions; p. 2 said how the lab stores them.
- Deck 3.1's grade: $0.0025\,\%$ loss over the band after 1000 iterations.
- **Deck 3.1's two questions, each answered once per iteration: evaluate on line 5, improve on line 6.**

Source: Session 3 lab, block B; the grade is the median of 20 seeds.

© Zhigang Feng Redistribution prohibited without permission

Without the Model: One-Step Actor–Critic

One-step actor–critic, online

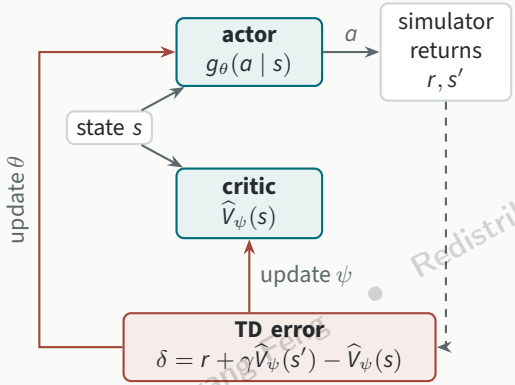
1 initialise θ, ψ; draw s from the start distribution
2 **for** $t = 0, 1, 2, \ldots$ up to the step limit:
3 draw $a \sim g_\theta(\cdot \mid s)$ on the feasible set; the simulator returns r and s'
4 $\delta \leftarrow r + \gamma \widehat{V}_\psi(s') - \widehat{V}_\psi(s)$
5 $\psi \leftarrow \psi + \eta_\psi \, \delta \, \nabla_\psi \widehat{V}_\psi(s)$ **critic**
6 $\theta \leftarrow \theta + \eta_\theta \, \delta \, \nabla_\theta \ln g_\theta(a \mid s)$ **actor**
7 at a reset, redraw s; otherwise $s \leftarrow s'$

- The **critic** evaluates by TD on the actor's own data; the **actor** climbs the score, weighted by δ: better than expected, a grows likelier.
- The rule's forms (p. 9): on the grid a **masked softmax**, $g_\theta(a \mid s) \propto \exp h_\theta(s, a)$ over the feasible k'; for a share, $x \sim N(\mu_\theta(s), \sigma^2)$, $a = \mathrm{sigmoid}(x)$.
- **One TD error does both jobs; the critic must move faster, it is the actor's yardstick.**

Source: Session 3 lab, block G.

© Vincent Pons — Redistribution prohibited without permission

The Data Flow: One TD Error, Two Updates



© Zhigang Feng
Redistribution prohibited without permission

- One number a period, the surprise δ: what happened less what the critic expected. It fixes the value and grades the action.
- The valuer must keep pace with the decider: two step sizes, the critic's the larger.
- The price: the surprise is noisy, and it leans on the critic; a wrong critic misleads the actor.
- **A differentiable model lets the actor differentiate through it instead (p. 15): deep value iteration, next.**

Session 3.4

With the Model: Deep Value Iteration

The actor–critic when the model is in hand: both functions networks, both steps n periods deep



© Zhigang Feng · Redistribution prohibited without permission

Deep Value Iteration: the Simplest Actor–Critic's Two Steps, with Networks



- **Deep value iteration is an actor–critic with the model in hand**: the actor steps along a pathwise derivative through n periods; the critic's target is that rollout, detached.
- Session 2's deterministic growth model, its own calibration. Actor g_θ: Session 2's share network. Critic V_ψ: its level started at $u(c^*)/(1-\gamma)$.
- Against p. 17's: no drawn actions, no exploring, a pathwise gradient. Against p. 16's: networks, n periods in both steps, one gradient step a round.
- **The name: at $n = 1$, with the actor at its $\max$, the critic's label is TV_ψ: fitted value iteration. The actor carries the $\max$.**

Source: Session 3 lab, block C.

For Miguel's Use Only
Redistribution prohibited without permission

The Two Steps, Written Out

Deep value iteration, as the lab runs it

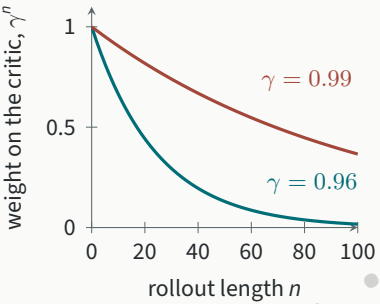
1 initialise the actor near the steady-state share, the critic's level at the steady-state value
2 **repeat** for a fixed number of rounds:
3 draw a fresh batch $k_1, \ldots, k_N$, uniform in $\ln k$ over the band
4 roll the actor n periods: $k_{t+1} = g_\theta(k_t)\, y(k_t)$, $c_t = y(k_t) - k_{t+1}$; form $J_n(k_i)$
5 $\theta \leftarrow \theta + \eta_\theta \nabla_\theta \frac{1}{N} \sum_i J_n(k_i)$ (through the budget constraint and into $V_\psi(k_n)$) **actor**
6 recompute $J_n(k_i)$ under the new θ, **detached**; $\psi \leftarrow \psi - \eta_\psi \nabla_\psi \frac{1}{N} \sum_i \left(V_\psi(k_i) - J_n(k_i)\right)^2$ **critic**

- $y(k)$ is output plus undepreciated capital, the resources. Line 5 is a pathwise gradient: the derivative runs through the model for n periods.
- **Detach**: the label in line 6 passes no gradient, or the critic could cut its loss by moving the label. No frozen copy: the live critic, detached (Deck 3.5's household adds one).
- **No benchmark enters either loss. Actor first, then critic, one gradient step each.**

Source: Session 3 lab, block C, which takes Adam steps.

© Zhigang Feng · Redistribution prohibited without permission

One Step or Many, and Where to Sample

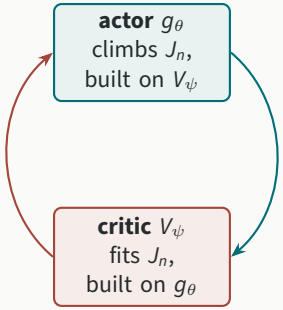


- $n = 1$ is fitted value iteration (p. 20): the actor sees the future only through the critic's slope.
- Larger n: label and objective look n periods ahead, and the critic's weight falls to γ^n; Deck 3.1's T-period backup lengthened only the label.
- No shocks here, so long rollouts add no variance; with shocks, the walk's noise (Decks 3.1, 3.3).
- Sampling: Session 2's band, uniform in $\ln k$. **Ergodic sampling** trains where the economy goes and is blind to rare states.
- **A solution is claimed only where it was trained and tested.**

Source: Session 3 lab, block C: γ of Session 2's rungs 0 and 1.

© Zhigang Feng
Redistribution prohibited without permission

Two Moving Targets, and a Residual That Can Mislead



- Each network's target moves with the other: let the critic move faster, and start both near the answer.
- The pair can agree and be wrong: a critic that values a bad actor correctly makes $V_\psi(k) - u(c) - \gamma V_\psi(k')$ small: consistent, not optimal.
- An Euler residual checks optimality, yet Session 2's lab found one small with a wrong rule, from a careless start.
- Build in shape when theory supplies it: a concave critic leaves the actor one peak in k' to climb.
- **No residual certifies an answer; a benchmark can.**

Source: Session 2 lab, block D.

© Zhigang Feng
Redistribution prohibited without permission

What Each Learns, and How It Improves the Rule

	DQN (p. 8)	**REINFORCE** (p. 10)	**actor–critic** (p. 17)
learns	values $\widehat{Q}_\psi(s, a)$; the rule is greedy in them	a rule $g_\theta(a \mid s)$; a value only as a baseline	a rule g_θ and a value $\widehat{V}_\psi$
improves the rule by	the $\max$ inside its target	score $\times$ (return $-$ baseline)	score $\times$ the surprise δ_t
waits for	one period	the end of the path	one period
data	any rule's, replayed	its own paths	its own transitions
the price	the triad; the $\max$'s optimism	variance	a wrong critic's bias

- The actor–critic itself came three times, the same two learners each time: the model summed over z' (p. 16), no model (p. 17), the model rolled forward n periods (p. 20). What differs is what each step reads.
- **Three ways to carry the future: DQN the value alone, REINFORCE the rule alone, the actor–critic both (Deck 3.1, *Two Questions, Two Learners*).**

© Zhigang Feng • Redistribution prohibited without permission

Why Economists Should Care: Actor–Critic as Two-Step Estimation

	Two-step estimation	Actor–critic
nuisance parameter	first stage	the critic's value: surprises uncorrelated with its gradient
target parameter	second stage	the actor's rule: zero update at the optimum

- p. 16's actor–critic: **a value fitted to its moment condition, a rule moved along its first-order condition**; interleaved, so the critic must keep ahead: prefitted, then refitted.
- **The saving rule is the estimand; the value function is the nuisance that makes it estimable.** The analogy stops there: none of the asymptotic theory carries over.
- **Grade against an exact solution while you can; report seeds as a range. Next, Deck 3.5: every algorithm on one growth model.**

© Zhuang Feng
Redistribution prohibited without permission