Economics-Inspired Neural Networks with Stabilizing ~~Homotopies~~ Adiabatic Model Transformations

Marlon Azinović[1] Jan Žemlička[2,3,4]

[1]University of Pennsylvania

[2]CERGE-EI, Charles University, [3]University of Zurich, [4]Swiss Finance Institute

Econometric Society Conference in Dynamic Structural Econometrics
University of Lausanne

25th of August 2023

Motivation

- Macro finance models involve aggregate risk + multiple assets
- Those models are challenging to solve
- Classical solution methods often fail, because
  - Linearization kills risk premia
  - Higher-order perturbations require too much smoothness
  - Grid based methods don't scale

Starting point

- Deep learning based solution methods: Azinovic et al. (2022) (DEQN), Ebrahimi Kahou et al. (2021); Maliar et al. (2021); Kase et al. (2022); Gu et al. (2023).[1]
  - Deep neural networks as an approximator for equilibrium functions of the economy
  - Trained to minimize equilibrium conditions error on a simulated ergodic set
- DEQN can handle stochastic models with many state variables, however, two pain points remain:
  - Portfolio choice
  - Market clearing

[1] See also Han et al. (2021); Valaitis and Villa (2021); Fernández-Villaverde et al. (2023), for different approaches involving deep learning.

This paper

- Two complementary innovations
  1. Market clearing layers
  2. Adiabatic model transformations for portfolio choice and asset prices
- Market clearing layers: neural network predictions are consistent with market clearing by design
- Adiabatic model tranformations: smoothly solve models with multiple assets

DEQN

Violations of equilibrium conditions as loss function

Basic idea in Azinovic et al. (2022): write equilibrium conditions as

$$\mathbf{G}(\mathbf{x}, \mathbf{f}) = 0 \ \forall \mathbf{x}$$

$$\begin{aligned} \mathbf{G} &: \text{ equilibrium conditions: FOC's, market clearing, Bellman equations, } \ldots \\ \mathbf{x} &: \text{ state of the economy} \\ \mathbf{f} &: \text{ equilibrium functions.} \end{aligned}$$

Approximate equilibrium functions by neural network $\mathcal{N}_{\boldsymbol{\rho}}(\cdot)$ ▸ Neural Nets

$$\mathcal{N}_{\boldsymbol{\rho}}(\mathbf{x}) \approx \mathbf{f}(\mathbf{x})$$

Given network parameters $\boldsymbol{\rho}$, we define a loss function

$$\mathrm{l}_{\boldsymbol{\rho}} := \frac{1}{N_{\text{path length}}} \sum_{\mathbf{x}_i \text{ on sim. path}} \left(\mathbf{G}(\mathbf{x}_i, \mathcal{N}_{\boldsymbol{\rho}})\right)^2$$

If $\mathrm{l}_{\boldsymbol{\rho}} \approx 0$, then $\mathcal{N}_{\boldsymbol{\rho}}(\mathbf{x})$ gives us a good approximation of $\mathbf{f}(\mathbf{x})$.

Training DEQNs

1. Simulate a sequence of states $\mathcal{D}^i_{\text{train}} \leftarrow \{\mathbf{x}^i_1, \mathbf{x}^i_2, \ldots, \mathbf{x}^i_T\}$ from the policy encoded by the network parameters $\boldsymbol{\rho}^i$.
2. Evaluate the errors of the equilibrium conditions on the newly generated set $\mathcal{D}_{\text{train}}$.
3. If the error statistics are not low enough:
   3.1 Update the parameters of the neural network with a gradient descent step (or a variant):
   $$\rho^{i+1}_k = \rho^i_k - \alpha_{\text{learn}} \frac{\partial \ell_{\mathcal{D}^i_{\text{train}}}(\boldsymbol{\rho}^i)}{\partial \rho^i_k}.$$
   3.2 Set new starting states for simulation: $\mathbf{x}^{i+1}_0 = \mathbf{x}^i_T$.
   3.3 Increase i by one and go back to step 1.

Simple Model

Simple OLG model with Capital and Bond

- Representative firm produces with

$$F(z_t, K_t, L_t) = z_t K_t^{\alpha} L_t^{1-\alpha} \quad (1)$$
$$w_t = \alpha z_t K_t^{\alpha-1} L_t^{1-\alpha} \quad (2)$$
$$r_t = z_t(1-\alpha) K_t^{\alpha} L_t^{\alpha}. \quad (3)$$

- Uncertainty in TFP z_t, and depreciation of capital δ_t

$$\log(z_{t+1}) = \rho_z \log(z_t) + \sigma_z \epsilon_t \quad (4)$$
$$\epsilon_t \sim N(0,1) \quad (5)$$
$$\delta_t = \delta \frac{2}{1+z} \quad (6)$$

- Assets
  - One period bond with price p_t in aggregate supply B
  - Risky capital K_t
  - Borrowing constraints on both assets

$$b_t^h \geq 0 \quad (7)$$
$$k_t^h \geq 0 \quad (8)$$

- Households
  - $H = 32$ age-groups, indexed with $h \in \mathcal{H} := \{1, \dots, 32\}$
  - supply labor units l_t^h inelastically
  - adjustment costs on capital

$$\Delta_{k,t}^h := k_{t+1}^{h+1} - k_t^h \quad (9)$$
$$\text{adj. costs} = \psi \left(\Delta_{k,t}^h\right)^2 \quad (10)$$

  - budget constraint

$$c_t^h = l_t^h w_t + b_{t-1}^{h-1} + k_{t-1}^{h-1}(1 - \delta_t + r_t) - p_t^b b_t^h + k_t^h - \psi \left(\Delta_{k,t}^h\right)^2 \quad (11)$$

  - Maximize

$$\mathrm{E}\left[\sum_{i=h}^{H} \beta^{i-h} u(c_{t+i}^{h+i})\right] \quad (12)$$
$$u(c) := \frac{c^{1-\gamma} - 1}{1-\gamma} \quad (13)$$

Equilibrium conditions

- **Market clearing**:

$$K_t := \sum_{h \in \mathcal{H}} k_t^h \tag{14}$$

$$B = \sum_{h \in \mathcal{H}} b_t^h \Leftrightarrow \epsilon_t^B := B - \sum_{h \in \mathcal{H}} b_t^h = 0 \tag{15}$$

- **Firms optimize**:

$$w_t := \alpha z_t K_t^{\alpha-1} L_t^{1-\alpha}$$
$$r_t := z_t(1-\alpha) K_t^{\alpha} L_t^{\alpha}$$

- **Households optimize**:

$$\left.\begin{aligned} 1 &= \frac{\beta \mathrm{E}\left[u'(c_{t+1}^{h+1})(1-\delta_{t+1}+r_{t+1}+2\psi^k \Delta_{k,t+1}^{h+1}\right]+\mu_t^h}{(1+2\psi^k \Delta_{k,t}^h)u'(c_t^h)} \\ k_t^h &\geq 0 \\ \mu_t^h &\geq 0 \\ k_t^h \mu_t^h &= 0 \end{aligned}\right\} \Leftrightarrow \epsilon_t^{k,h} := \psi^{FB}\left(\frac{u'^{-1}\left(\beta \mathrm{E}\left[u'(c_{t+1}^{h+1})\frac{(1-\delta_{t+1}+r_{t+1}+2\psi^k \Delta_{k,t+1}^{h+1}}{(1+2\psi^k \Delta_{k,t}^h)}\right]\right)}{c_t^h} - 1, \frac{k_t^h}{c_t^h} \right) = 0$$

$$\left.\begin{aligned} 1 &= \frac{\beta \mathrm{E}\left[u'(c_{t+1}^{h+1})\right]+\lambda_t^h}{p_t^b u'(c_t^h)} \\ b_t^h - \underline{b} &\geq 0 \\ \lambda_t^h &\geq 0 \\ (b_t^h - \underline{b}^h)\lambda_t^h &= 0 \end{aligned}\right\} \Leftrightarrow \epsilon_t^{b,h} := \psi^{FB}\left(\frac{u'^{-1}\left(\beta \mathrm{E}\left[\frac{1}{p_t^b} u'(c_{t+1}^{h+1})\right]\right)}{c_t^h} - 1, \frac{b_t^h - \underline{b}}{c_t^h} \right) = 0$$

where

$$\psi^{FB}(a,b) := a + b - \sqrt{a^2 + b^2} \tag{16}$$

Approximation with standard DEQN

- State of the economy

$$\mathbf{x}_t = [\underbrace{z_t}_{\text{ex. shock}}, \underbrace{k_t^1, \dots, k_t^{32}}_{\text{dist. of cap.}}, \underbrace{b_t^1, \dots, b_t^{32}}_{\text{dist. of bonds}}] \tag{17}$$

- Equilibrium policies

$$\mathbf{f}(\mathbf{x}_t) = [\underbrace{k_{t+1}^1, \dots, k_{t+1}^{32}}_{\text{capital policy}}, \underbrace{b_{t+1}^1, \dots, b_{t+1}^{32}}_{\text{bond policy}}, \underbrace{p_t^b}_{\text{bond price}}] \tag{18}$$

- Neural network approximates

$$\mathcal{N}_\rho(\mathbf{x}_t) = [\underbrace{\hat{k}_{t+1}^1, \dots, \hat{k}_{t+1}^{32}}_{\text{capital policy}}, \underbrace{\hat{b}_{t+1}^1, \dots, \hat{b}_{t+1}^{32}}_{\text{bond policy}}, \underbrace{\hat{p}_t^b}_{\text{bond price}}] \approx \mathbf{f}(\mathbf{x}_t) \tag{19}$$

- loss function

$$\ell_\rho(\mathbf{x}_t) := \underbrace{w_{hh,k}}_{\text{weight}} \underbrace{\left(\sum_{h=1}^{H-1} \left(\epsilon_t^{k,h}\right)^2\right)}_{\text{opt. cond. cap.}} + \underbrace{w_{hh,b}}_{\text{weight}} \underbrace{\left(\sum_{h=1}^{H-1} \left(\epsilon_t^{b,h}\right)^2\right)}_{\text{opt. cond. bond}} + \underbrace{w_{mc,B}}_{\text{weight}} \underbrace{\left(\epsilon_t^B\right)^2}_{\text{market clearing}} \tag{20}$$

Motivation for market clearing layers

- Loss function should encode all equations which pin-down equilibrium objects of the economy
  - First-order conditions
  - Bellman equations
  - Market clearing
  - Government budget constraint
- Problem: different equations have different units or different economic relevance
- More terms in loss function $\Rightarrow$ harder to interpret the resulting loss
- Simulated states are not consistent with the equilibrium conditions
- Why force the neural network to learn something we already know?

Innovation 1: Market clearing layers

- Neural network first predicts

$$\mathcal{N}_{\boldsymbol{\rho}}^{\text{pre}}(\mathbf{x}_t) = [\hat{k}_{t+1}^{1}, \ldots, \hat{k}_{t+1}^{32}, \tilde{b}_{t+1}^{1}, \ldots, \tilde{b}_{t+1}^{32}, \hat{p}_t^{b}] \tag{21}$$

- Apply transformation $m(\ldots, \cdot)$

$$[\hat{b}_{t+1}^{1}, \ldots, \hat{b}_{t+1}^{32}] = m\left(\mathcal{N}_{\boldsymbol{\rho}}^{\text{pre}}(\mathbf{x}_t), B\right) \tag{22}$$

- Such that

$$B = \sum_{h=1}^{32} \hat{b}_{t+1}^{h} \tag{23}$$

- Put together

$$\mathcal{N}_{\boldsymbol{\rho}}(\mathbf{x}_t) := [\hat{k}_{t+1}^{1}, \ldots, \hat{k}_{t+1}^{32}, \hat{b}_{t+1}^{1}, \ldots, \hat{b}_{t+1}^{32}, \hat{p}_t^{b}] \tag{24}$$

- Loss function now

$$\ell_{\boldsymbol{\rho}}(\mathbf{x}_t) := \underbrace{w_{hh,k}}_{\text{weight}} \underbrace{\left(\sum_{h=1}^{H-1} \left(\epsilon_t^{k,h}\right)^2\right)}_{\text{opt. cond. cap.}} + \underbrace{w_{hh,b}}_{\text{weight}} \underbrace{\left(\sum_{h=1}^{H-1} \left(\epsilon_t^{b,h}\right)^2\right)}_{\text{opt. cond. bond}} + \cancelto{= 0}{\underbrace{w_{mc,B}}_{\text{weight}} \underbrace{\left(\epsilon_t^{B}\right)^2}_{\text{market clearing}}} \tag{25}$$

Innovation 1: Market clearing layers

- Neural network first predicts

$$\mathcal{N}_{\rho}^{\text{pre}}(\mathbf{x}_t) = [\hat{k}_{t+1}^{1}, \ldots, \hat{k}_{t+1}^{32}, \tilde{b}_{t+1}^{1}, \ldots, \tilde{b}_{t+1}^{32}, \hat{p}_t^{b}] \tag{21}$$

- Apply transformation $m(\ldots, \cdot)$

$$[\hat{b}_{t+1}^{1}, \ldots, \hat{b}_{t+1}^{32}] = m\left(\mathcal{N}_{\rho}^{\text{pre}}(\mathbf{x}_t), B\right) \tag{22}$$

- Such that

$$B = \sum_{h=1}^{32} \hat{b}_{t+1}^{h} \tag{23}$$

- Put together

$$\mathcal{N}_{\rho}(\mathbf{x}_t) := [\hat{k}_{t+1}^{1}, \ldots, \hat{k}_{t+1}^{32}, \hat{b}_{t+1}^{1}, \ldots, \hat{b}_{t+1}^{32}, \hat{p}_t^{b}] \tag{24}$$

- Loss function now

$$\ell_{\rho}(\mathbf{x}_t) := \underbrace{w_{hh,k}}_{\text{weight}} \underbrace{\left(\sum_{h=1}^{H-1}\left(\epsilon_t^{k,h}\right)^2\right)}_{\text{opt. cond. cap.}} + \underbrace{w_{hh,b}}_{\text{weight}} \underbrace{\left(\sum_{h=1}^{H-1}\left(\epsilon_t^{b,h}\right)^2\right)}_{\text{opt. cond. bond}} + \cancelto{0}{\underbrace{w_{mc,B}}_{\text{weight}} \underbrace{\left(\epsilon_t^{B}\right)^2}_{\text{market clearing}}} \tag{25}$$

1. remaining loss easier to interpret
2. states simulated from the policy are always consistent with market clearing

Innovation 1: Details on the market clearing transformation function

- Simple market clearing layer: subtract excess demand ED_t from initial predictions

$$ED_t := \sum_{h \in \mathcal{H}} \tilde{b}^h_{t+1} - B \tag{26}$$

$$\hat{b}^h_{t+1} := \tilde{b}^h_{t+1} - \frac{1}{H} ED_t \tag{27}$$

- Why this adjustment?

$\rightarrow$ we try to minimize the modification to the initial predictions $\{\tilde{b}^h_{t+1}\}_{h \in \mathcal{H}}$.

- Final predictions $\{\hat{b}^h_{t+1}\}_{h \in \mathcal{H}}$ solve

$$\operatorname*{arg\,min}_{\{x^h_{t+1}\}_{h \in \mathcal{H}}} \sum_{h \in \mathcal{H}} \left(x^h_{t+1} - \tilde{b}^h_{t+1} \right)^2 \tag{28}$$

subject to

$$\sum_{h \in \mathcal{H}} x^h_{t+1} = B \tag{29}$$

Innovation 1: Details on the market clearing transformation function

- Simple market clearing layer: subtract excess demand ED_t from initial predictions

$$ED_t := \sum_{h \in \mathcal{H}} \tilde{b}^h_{t+1} - B \tag{26}$$

$$\hat{b}^h_{t+1} := \tilde{b}^h_{t+1} - \frac{1}{H} ED_t \tag{27}$$

- Why this adjustment?

$\rightarrow$ we try to minimize the modification to the initial predictions $\{\tilde{b}^h_{t+1}\}_{h \in \mathcal{H}}$.

- Final predictions $\{\hat{b}^h_{t+1}\}_{h \in \mathcal{H}}$ solve

$$\underset{\{x^h_{t+1}\}_{h \in \mathcal{H}}}{\arg\min} \sum_{h \in \mathcal{H}} \left(x^h_{t+1} - \tilde{b}^h_{t+1} \right)^2 \tag{28}$$

subject to

$$\sum_{h \in \mathcal{H}} x^h_{t+1} = B \tag{29}$$

- Downside: borrowing constraint not necessarily satisfied.
- In the paper: enforcing market clearing & borrowing constraints using implicit layer
- Downside: slows down the algorithm, need to restore signal for falsely constrained agents

Innovation 2: Stabilizing adiabatic model transformations

- Single asset models are easy

Innovation 2: Stabilizing adiabatic model transformations

- Single asset models are easy
- Many asset models are hard

Innovation 2: Stabilizing adiabatic model transformations

- Single asset models are easy
- Many asset models are hard
- Why?
  - Portfolio choice only pinned down at low errors in equilibrium conditions
  - But how do we get there?

Innovation 2: Stabilizing adiabatic model transformations

- Single asset models are easy
- Many asset models are hard
- Why?
  - Portfolio choice only pinned down at low errors in equilibrium conditions
  - But how do we get there?
- Adiabatic model transformations
  1. $N-1$ asset models are nested in N asset models

Innovation 2: Stabilizing adiabatic model transformations

- Single asset models are easy
- Many asset models are hard
- Why?
  - Portfolio choice only pinned down at low errors in equilibrium conditions
  - But how do we get there?
- Adiabatic model transformations
  1. $N-1$ asset models are nested in N asset models
  2. Start with single asset model

$$\mathcal{N}^1_{\boldsymbol{\rho}}(\mathbf{x}_t) = [\hat{k}^1_{t+1}, \ldots, \hat{k}^{32}_{t+1}, 0 \times \hat{b}^1_{t+1}, \ldots, 0 \times \hat{b}^{32}_{t+1}, \hat{p}^b_t], B^1 = 0 \tag{30}$$

Innovation 2: Stabilizing adiabatic model transformations

- Single asset models are easy
- Many asset models are hard
- Why?
  - Portfolio choice only pinned down at low errors in equilibrium conditions
  - But how do we get there?
- Adiabatic model transformations
  1. $N-1$ asset models are nested in N asset models
  2. Start with single asset model

$$\mathcal{N}^1_{\boldsymbol{\rho}}(\mathbf{x}_t) = [\hat{k}^1_{t+1}, \ldots, \hat{k}^{32}_{t+1}, 0 \times \hat{b}^1_{t+1}, \ldots, 0 \times \hat{b}^{32}_{t+1}, \hat{p}^b_t], B^1 = 0 \tag{30}$$

  3. Solve the model

Innovation 2: Stabilizing adiabatic model transformations

- Single asset models are easy
- Many asset models are hard
- Why?
  - Portfolio choice only pinned down at low errors in equilibrium conditions
  - But how do we get there?
- Adiabatic model transformations
  1. $N-1$ asset models are nested in N asset models
  2. Start with single asset model

$$\mathcal{N}^1_{\rho}(\mathbf{x}_t) = [\hat{k}^1_{t+1}, \dots, \hat{k}^{32}_{t+1}, 0 \times \hat{b}^1_{t+1}, \dots, 0 \times \hat{b}^{32}_{t+1}, \hat{p}^b_t], B^1 = 0 \tag{30}$$

  3. Solve the model
  4. Train bond price (supervised, from zero liquidity limit)

Innovation 2: Stabilizing adiabatic model transformations

- Single asset models are easy
- Many asset models are hard
- Why?
  - Portfolio choice only pinned down at low errors in equilibrium conditions
  - But how do we get there?
- Adiabatic model transformations
  1. $N-1$ asset models are nested in N asset models
  2. Start with single asset model

$$\mathcal{N}^1_{\boldsymbol{\rho}}(\mathbf{x}_t) = [\hat{k}^1_{t+1}, \dots, \hat{k}^{32}_{t+1}, 0\times\hat{b}^1_{t+1}, \dots, 0\times\hat{b}^{32}_{t+1}, \hat{p}^b_t], B^1 = 0 \tag{30}$$

  3. Solve the model
  4. Train bond price (supervised, from zero liquidity limit)
  5. Slowly introduce the second asset (adiabatically)

$$\mathcal{N}^2_{\boldsymbol{\rho}}(\mathbf{x}_t) = [\hat{k}^1_{t+1}, \dots, \hat{k}^{32}_{t+1}, 1\times\hat{b}^1_{t+1}, \dots, 1\times\hat{b}^{32}_{t+1}, \hat{p}^b_t], B^2 = 0.1$$

Innovation 2: Stabilizing adiabatic model transformations

- Single asset models are easy
- Many asset models are hard
- Why?
  - Portfolio choice only pinned down at low errors in equilibrium conditions
  - But how do we get there?
- Adiabatic model transformations
  1. $N-1$ asset models are nested in N asset models
  2. Start with single asset model

$$\mathcal{N}^1_{\boldsymbol{\rho}}(\mathbf{x}_t) = [\hat{k}^1_{t+1}, \dots, \hat{k}^{32}_{t+1}, 0 \times \hat{b}^1_{t+1}, \dots, 0 \times \hat{b}^{32}_{t+1}, \hat{p}^b_t], B^1 = 0 \tag{30}$$

  3. Solve the model
  4. Train bond price (supervised, from zero liquidity limit)
  5. Slowly introduce the second asset (adiabatically)

$$\mathcal{N}^3_{\boldsymbol{\rho}}(\mathbf{x}_t) = [\hat{k}^1_{t+1}, \dots, \hat{k}^{32}_{t+1}, 1 \times \hat{b}^1_{t+1}, \dots, 1 \times \hat{b}^{32}_{t+1}, \hat{p}^b_t], B^3 = 0.2$$

Innovation 2: Stabilizing adiabatic model transformations

- Single asset models are easy
- Many asset models are hard
- Why?
  - Portfolio choice only pinned down at low errors in equilibrium conditions
  - But how do we get there?
- Adiabatic model transformations
  1. $N - 1$ asset models are nested in N asset models
  2. Start with single asset model

$$\mathcal{N}_{\boldsymbol{\rho}}^1(\mathbf{x}_t) = [\hat{k}_{t+1}^1, \ldots, \hat{k}_{t+1}^{32}, 0 \times \hat{b}_{t+1}^1, \ldots, 0 \times \hat{b}_{t+1}^{32}, \hat{p}_t^b], B^1 = 0 \tag{30}$$

  3. Solve the model
  4. Train bond price (supervised, from zero liquidity limit)
  5. Slowly introduce the second asset (adiabatically)

$$\mathcal{N}_{\boldsymbol{\rho}}^4(\mathbf{x}_t) = [\hat{k}_{t+1}^1, \ldots, \hat{k}_{t+1}^{32}, 1 \times \hat{b}_{t+1}^1, \ldots, 1 \times \hat{b}_{t+1}^{32}, \hat{p}_t^b], B^4 = 0.3$$

Innovation 2: Stabilizing adiabatic model transformations

- Single asset models are easy
- Many asset models are hard
- Why?
  - Portfolio choice only pinned down at low errors in equilibrium conditions
  - But how do we get there?
- Adiabatic model transformations
  1. $N-1$ asset models are nested in N asset models
  2. Start with single asset model

$$\mathcal{N}_{\boldsymbol{\rho}}^{1}(\mathbf{x}_t) = [\hat{k}_{t+1}^{1}, \ldots, \hat{k}_{t+1}^{32}, 0\times\hat{b}_{t+1}^{1}, \ldots, 0\times\hat{b}_{t+1}^{32}, \hat{p}_t^{b}], B^1 = 0 \tag{30}$$

  3. Solve the model
  4. Train bond price (supervised, from zero liquidity limit)
  5. Slowly introduce the second asset (adiabatically)

$$\mathcal{N}_{\boldsymbol{\rho}}^{5}(\mathbf{x}_t) = [\hat{k}_{t+1}^{1}, \ldots, \hat{k}_{t+1}^{32}, 1\times\hat{b}_{t+1}^{1}, \ldots, 1\times\hat{b}_{t+1}^{32}, \hat{p}_t^{b}], B^5 = 0.4$$

Innovation 2: Stabilizing adiabatic model transformations

- Single asset models are easy
- Many asset models are hard
- Why?
  - Portfolio choice only pinned down at low errors in equilibrium conditions
  - But how do we get there?
- Adiabatic model transformations
  1. $N-1$ asset models are nested in N asset models
  2. Start with single asset model

$$\mathcal{N}_{\boldsymbol{\rho}}^{1}(\mathbf{x}_t) = [\hat{k}_{t+1}^{1}, \ldots, \hat{k}_{t+1}^{32}, 0 \times \hat{b}_{t+1}^{1}, \ldots, 0 \times \hat{b}_{t+1}^{32}, \hat{p}_t^{b}], B^1 = 0 \tag{30}$$

  3. Solve the model
  4. Train bond price (supervised, from zero liquidity limit)
  5. Slowly introduce the second asset (adiabatically)

$$\mathcal{N}_{\boldsymbol{\rho}}^{\cdots}(\mathbf{x}_t) = [\hat{k}_{t+1}^{1}, \ldots, \hat{k}_{t+1}^{32}, 1 \times \hat{b}_{t+1}^{1}, \ldots, 1 \times \hat{b}_{t+1}^{32}, \hat{p}_t^{b}], B^{\cdots} = \ldots$$

Innovation 2: Stabilizing adiabatic model transformations

- Single asset models are easy
- Many asset models are hard
- Why?
  - Portfolio choice only pinned down at low errors in equilibrium conditions
  - But how do we get there?
- Adiabatic model transformations
  1. $N-1$ asset models are nested in N asset models
  2. Start with single asset model

$$\mathcal{N}_{\boldsymbol{\rho}}^{1}(\mathbf{x}_t) = [\hat{k}_{t+1}^{1}, \ldots, \hat{k}_{t+1}^{32}, 0 \times \hat{b}_{t+1}^{1}, \ldots, 0 \times \hat{b}_{t+1}^{32}, \hat{p}_t^{b}], B^1 = 0 \tag{30}$$

  3. Solve the model
  4. Train bond price (supervised, from zero liquidity limit)
  5. Slowly introduce the second asset (adiabatically)

$$\mathcal{N}_{\boldsymbol{\rho}}^{101}(\mathbf{x}_t) = [\hat{k}_{t+1}^{1}, \ldots, \hat{k}_{t+1}^{32}, 1 \times \hat{b}_{t+1}^{1}, \ldots, 1 \times \hat{b}_{t+1}^{32}, \hat{p}_t^{b}], B^{100} = 10$$

Innovation 2: Stabilizing adiabatic model transformations

- Single asset models are easy
- Many asset models are hard
- Why?
  - Portfolio choice only pinned down at low errors in equilibrium conditions
  - But how do we get there?
- Adiabatic model transformations
  1. $N-1$ asset models are nested in N asset models
  2. Start with single asset model

$$\mathcal{N}_{\boldsymbol{\rho}}^{1}(\mathbf{x}_t) = [\hat{k}_{t+1}^{1}, \ldots, \hat{k}_{t+1}^{32}, 0\times\hat{b}_{t+1}^{1}, \ldots, 0\times\hat{b}_{t+1}^{32}, \hat{p}_t^{b}], B^{1} = 0 \tag{30}$$

  3. Solve the model
  4. Train bond price (supervised, from zero liquidity limit)
  5. Slowly introduce the second asset (adiabatically)

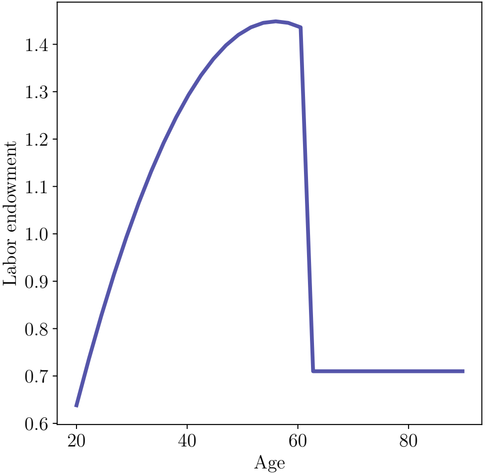
$$\mathcal{N}_{\boldsymbol{\rho}}^{101}(\mathbf{x}_t) = [\hat{k}_{t+1}^{1}, \ldots, \hat{k}_{t+1}^{32}, 1\times\hat{b}_{t+1}^{1}, \ldots, 1\times\hat{b}_{t+1}^{32}, \hat{p}_t^{b}], B^{100} = 10$$

  6. Equilibrium errors always remain low

Application to our simple model

Parameters

Parameters	H	β	γ	ψ	ρ	σ	α
Values	32	0.912	4	0.1	0.693	0.052	0.333
Meaning	num. age groups	patience	RRA	adj. costs	pers. tfp	std. innov. tfp	cap. share



Step 1: solve single asset model

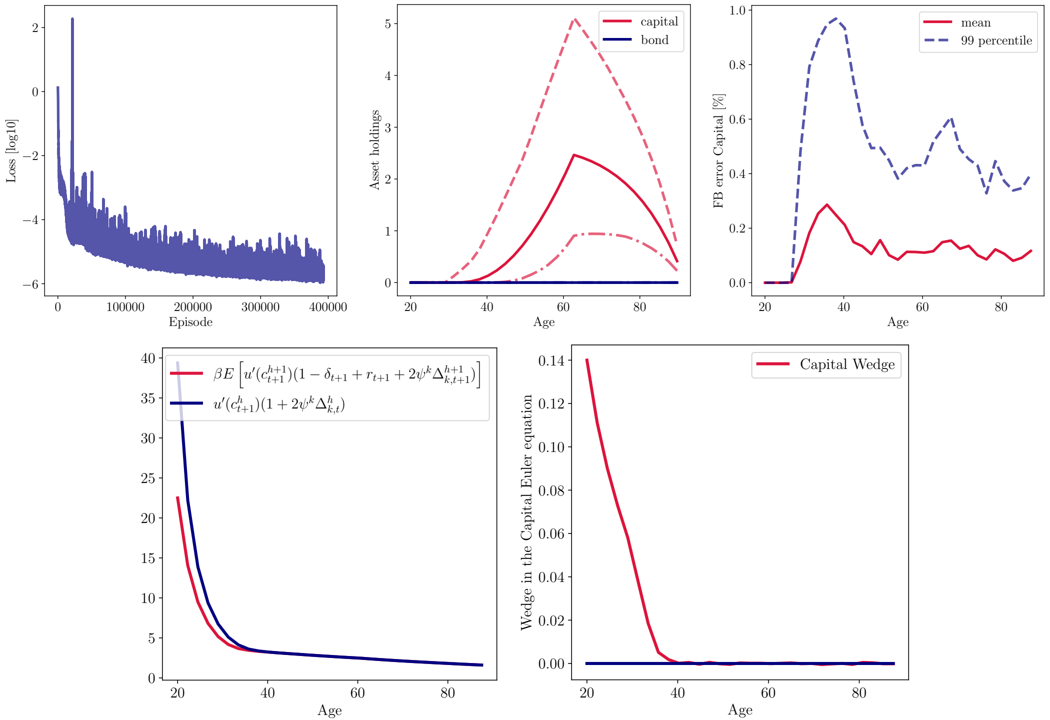
- Borrowing constraint $\underline{b} = 0$, net-supply $B = 0$
- Neural network predicts

$$\mathcal{N}_{\boldsymbol{\rho}}^{\text{pre}}(\mathbf{x}_t) = [\hat{k}_{t+1}^{1}, \ldots, \hat{k}_{t+1}^{32}, 0\times \tilde{b}_{t+1}^{1}, \ldots, 0\times \tilde{b}_{t+1}^{32}, \hat{p}_t^{b}] \tag{31}$$

$$\Rightarrow \mathcal{N}_{\boldsymbol{\rho}}(\mathbf{x}_t) = [\hat{k}_{t+1}^{1}, \ldots, \hat{k}_{t+1}^{32}, 0, \ldots, 0, \hat{p}_t^{b}] \tag{32}$$

- Loss function

$$\ell_{\boldsymbol{\rho}}(\mathbf{x}_t) := 1\times \underbrace{\left(\sum_{h=1}^{H-1}\left(\epsilon_t^{k,h}\right)^2\right)}_{\text{opt. cond. cap.}} + \underbrace{0\times \underbrace{\left(\sum_{h=1}^{H-1}\left(\epsilon_t^{b,h}\right)^2\right)}_{\text{opt. cond. bond}}}_{=0} \tag{33}$$


Loss [log10]
Episode
Asset holdings
Age
capital
bond
FB error Capital [%]
mean
99 percentile
$\beta E\left[u'(c_{t+1}^{h+1})(1-\delta_{t+1}+r_{t+1}+2\psi^k\Delta_{k,t+1}^{h+1})\right]$
$u'(c_{t+1}^{h})(1+2\psi^k\Delta_{k,t}^{h})$
Wedge in the Capital Euler equation
Capital Wedge

Step 2: pretrain bond price in the capital only model

- Keep borrowing constraint $\underline{b} = 0$, net-supply $B = 0$, and neural network masks

$$\mathcal{N}_\rho(\mathbf{x}_t) = [\hat{k}^1_{t+1}, \ldots, \hat{k}^{32}_{t+1}, 0, \ldots, 0, \hat{p}^b_t] \tag{34}$$

- In equilibrium we know that

$$p^b_t \geq \frac{\beta \mathrm{E}\left[u'(c^{h+1}_{t+1})\right]}{u'(c^h_t)} \tag{35}$$

with equality for unconstrained agents.

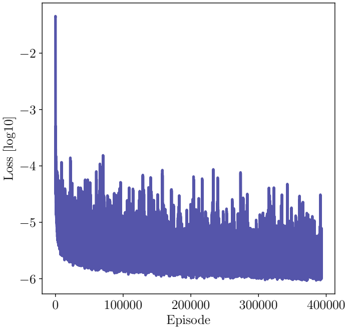
- With market clearing policies, we have a closed form expression for the bond price and can define pre-train price and error

$$p^{b,\text{pretrain}}_t := \max_{h \in \mathcal{H}} \left\{ \frac{\beta \mathrm{E}\left[u'(c^{h+1}_{t+1})\right]}{u'(c^h_t)} \right\} \tag{36}$$

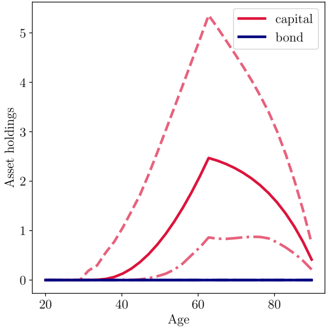
$$\epsilon^{\text{pretrain}}_t := p^{b,\text{pretrain}}_t - \hat{p}^b_t \tag{37}$$

- Loss function

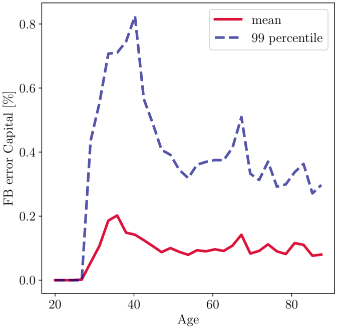
$$\ell_\rho(\mathbf{x}_t) := 1 \times \underbrace{\left(\sum_{h=1}^{H-1} \left(\epsilon^{k,h}_t \right)^2 \right)}_{\text{opt. cond. cap.}} + 0 \times \underbrace{\underbrace{\left(\sum_{h=1}^{H-1} \left(\epsilon^{b,h}_t \right)^2 \right)}_{\text{opt. cond. bond}}}_{=0} + 1 \times \underbrace{\left(\epsilon^{\text{pretrain}}_t \right)^2}_{\substack{\text{price pretrain error} \\ \text{train supervised}}} \tag{38}$$



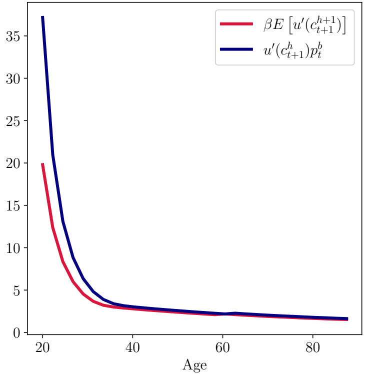
Loss [log10]
−2
−3
−4
−5
−6
0
100000
200000
300000
400000
Episode



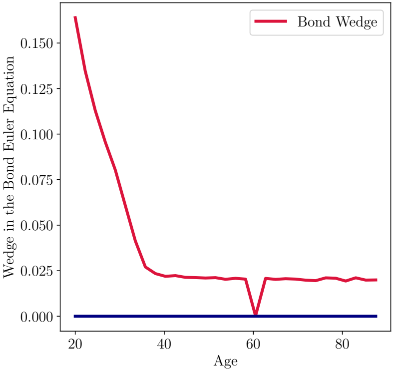
capital
bond
Asset holdings
0
1
2
3
4
5
20
40
60
80
Age



mean
99 percentile
FB error Capital [%]
0.0
0.2
0.4
0.6
0.8
20
40
60
80
Age



$\beta E\left[u'(c^{h+1}_{t+1})\right]$
$u'(c^{h}_{t+1})p^{b}_{t}$
0
5
10
15
20
25
30
35
20
40
60
80
Age



Bond Wedge
Wedge in the Bond Euler Equation
0.000
0.025
0.050
0.075
0.100
0.125
0.150
20
40
60
80
Age

Step 3: slowly increase bond supply

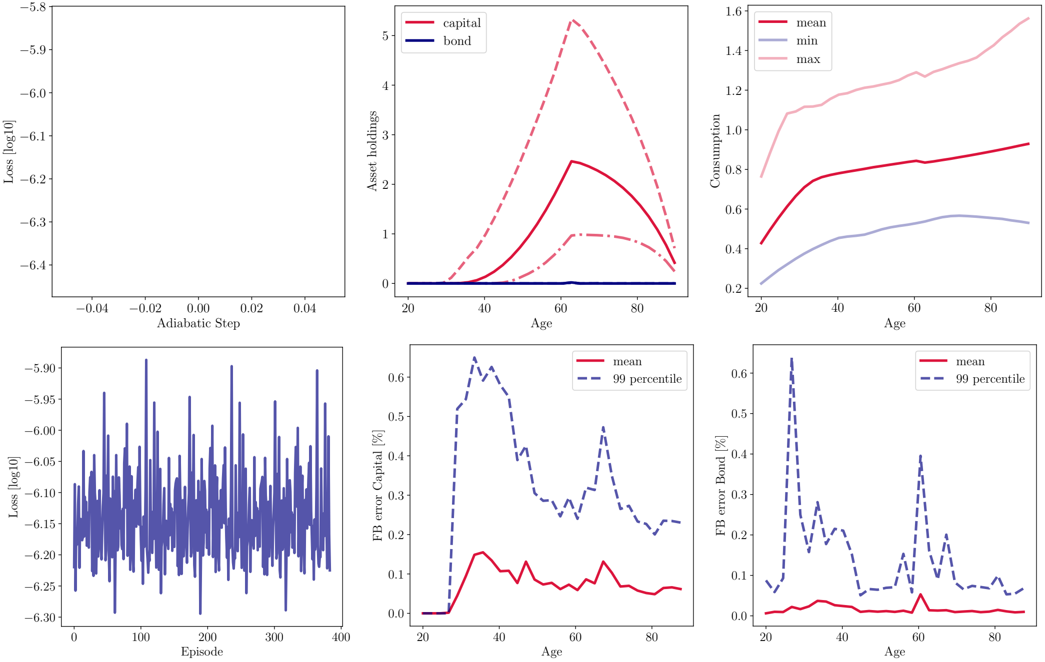
- Borrowing constraint $\underline{b} = 0$, increase net-supply from $B = 0.1$ to $B = 10$
- Neural network predicts

$$\mathcal{N}_{\rho}^{\text{pre}}(\mathbf{x}_t) = [\hat{k}_{t+1}^{1}, \ldots, \hat{k}_{t+1}^{32}, \underbrace{0.01 \times \tilde{b}_{t+1}^{1}, \ldots, 0.01 \times \tilde{b}_{t+1}^{32}}_{\text{bond policies active}}, \hat{p}_t^b] \tag{39}$$

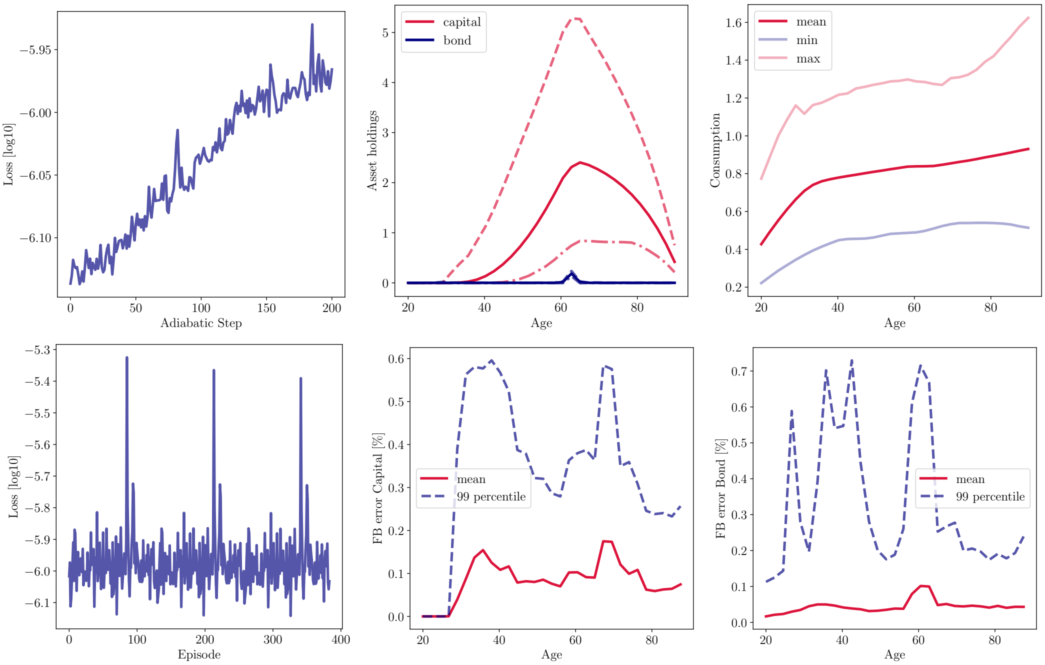
$$\Rightarrow \mathcal{N}_{\rho}(\mathbf{x}_t) = [\hat{k}_{t+1}^{1}, \ldots, \hat{k}_{t+1}^{32}, \underbrace{\hat{b}_{t+1}^{1}, \ldots, \hat{b}_{t+1}^{32}}_{\text{always add up the B}}, \hat{p}_t^b] \tag{40}$$

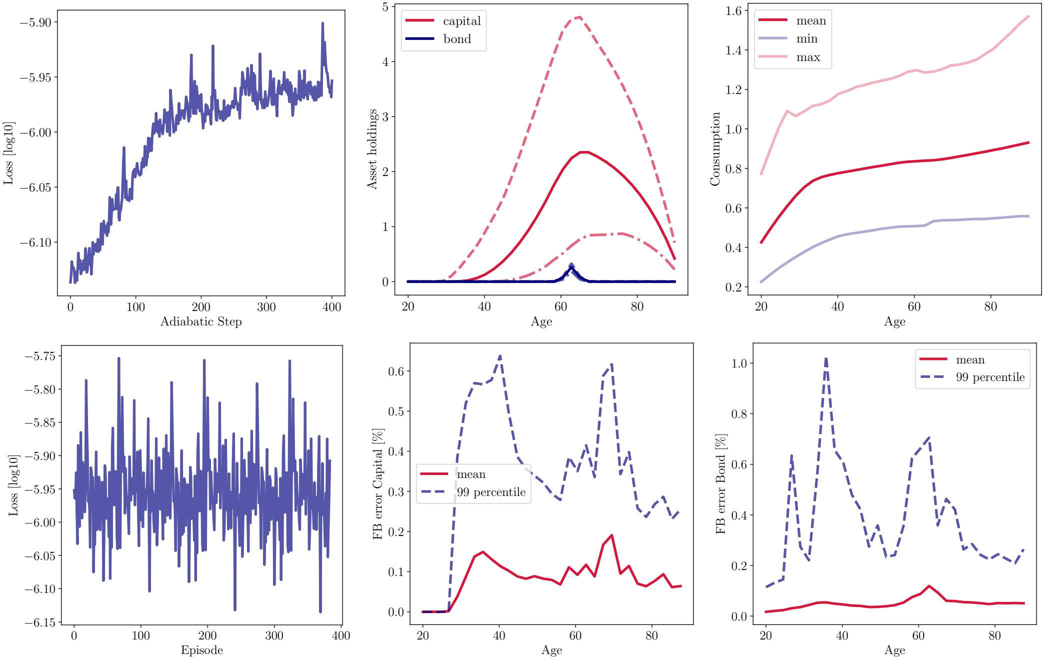
- Loss function

$$\ell_{\rho}(\mathbf{x}_t) := 1 \times \underbrace{\left(\sum_{h=1}^{H-1} \left(\epsilon_t^{k,h} \right)^2 \right)}_{\text{opt. cond. cap.}} + \underbrace{1 \times}_{\text{bond equ. cond. active}} \underbrace{\left(\sum_{h=1}^{H-1} \left(\epsilon_t^{b,h} \right)^2 \right)}_{\text{opt. cond. bond}} \tag{41}$$

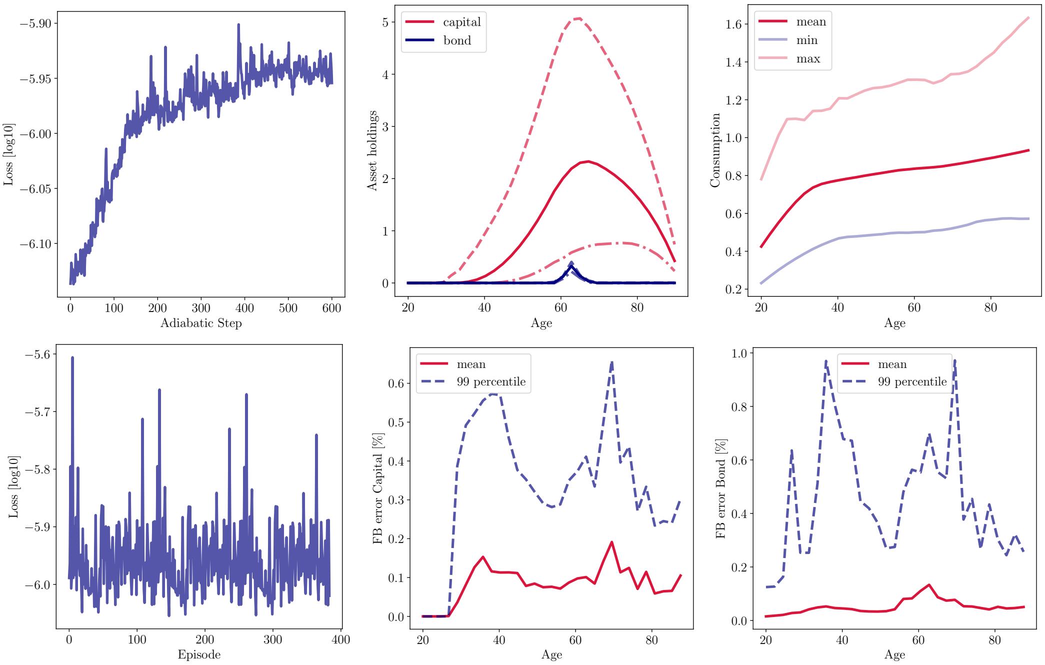


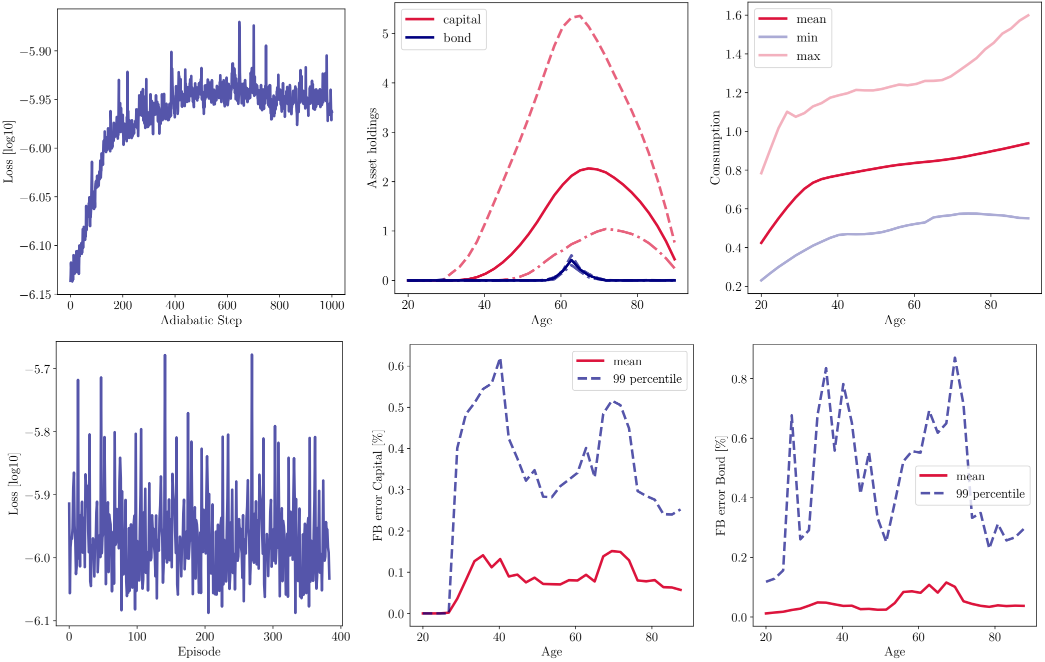
−5.8
−5.9
−6.0
−6.1
−6.2
−6.3
−6.4
Loss [log10]
−0.04
−0.02
0.00
0.02
0.04
Adiabatic Step
capital
bond
Asset holdings
Age
mean
min
max
Consumption
Age
Loss [log10]
Episode
mean
99 percentile
FB error Capital [%]
Age
mean
99 percentile
FB error Bond [%]
Age

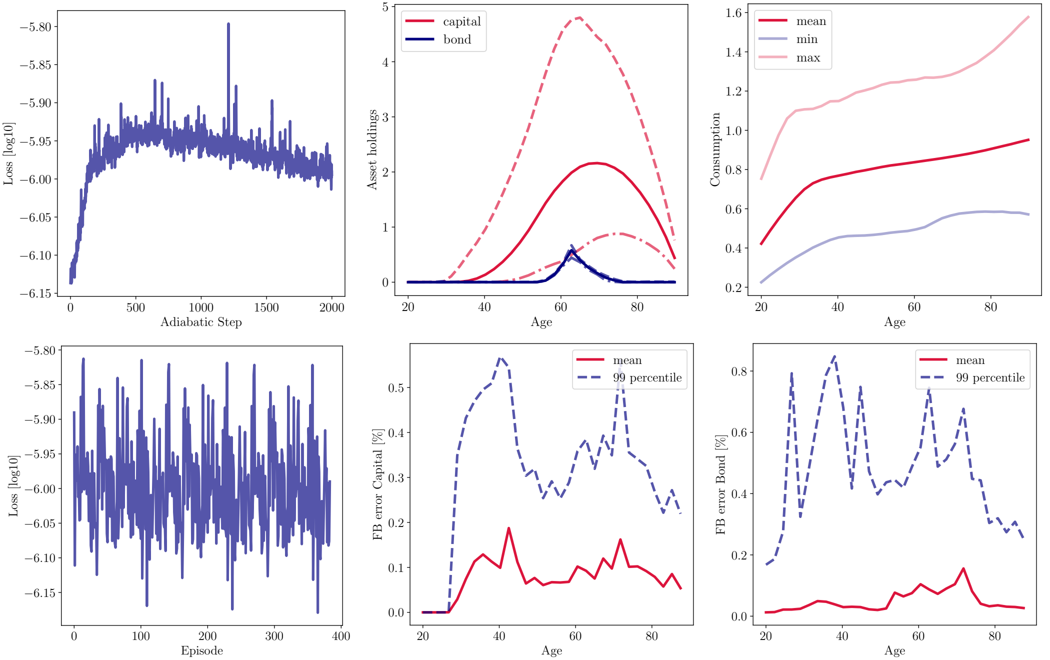

Loss [log10]
Adiabatic Step
capital
bond
Asset holdings
Age
mean
min
max
Consumption
Age
Loss [log10]
Episode
mean
99 percentile
FB error Capital [%]
Age
mean
99 percentile
FB error Bond [%]
Age



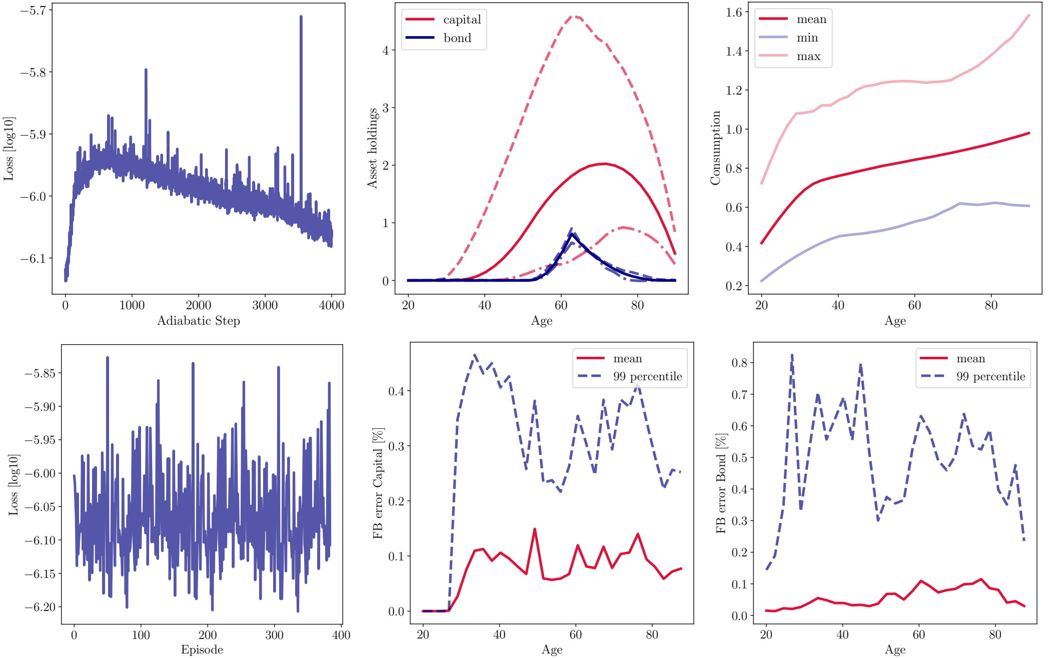
Loss [log10]
Adiabatic Step
Asset holdings
Age
capital
bond
Consumption
mean
min
max
Loss [log10]
Episode
FB error Capital [%]
mean
99 percentile
FB error Bond [%]
mean
99 percentile

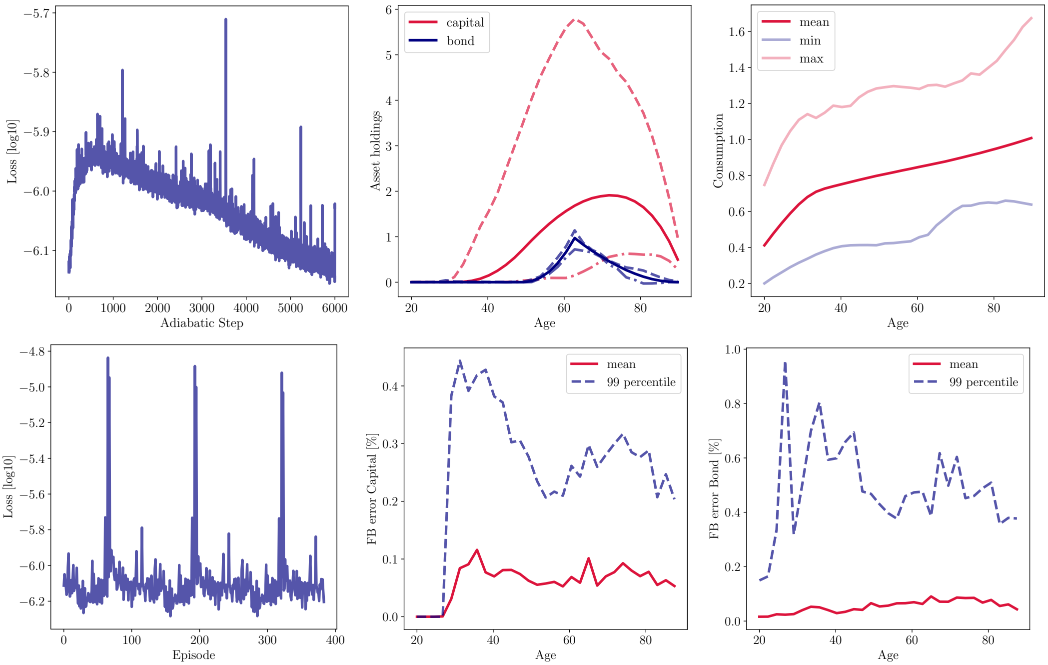

Loss [log10]
Adiabatic Step
capital
bond
Asset holdings
Age
mean
min
max
Consumption
Loss [log10]
Episode
mean
99 percentile
FB error Capital [%]
FB error Bond [%]

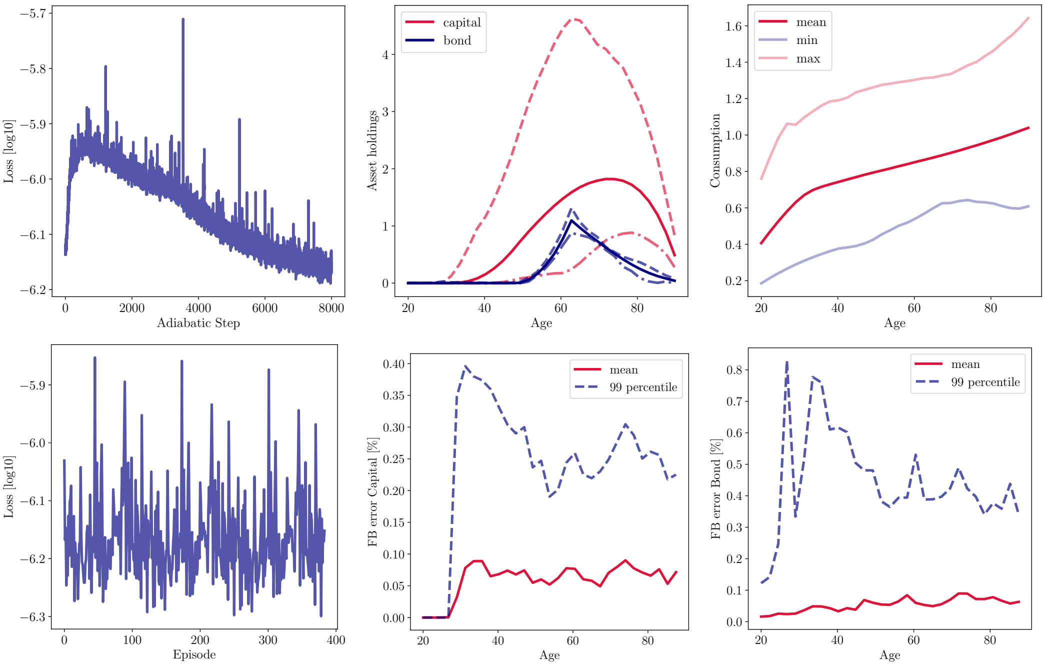

Loss [log10]
Adiabatic Step
Asset holdings
capital
bond
Age
Consumption
mean
min
max
Age
Loss [log10]
Episode
FB error Capital [%]
mean
99 percentile
Age
FB error Bond [%]
mean
99 percentile
Age



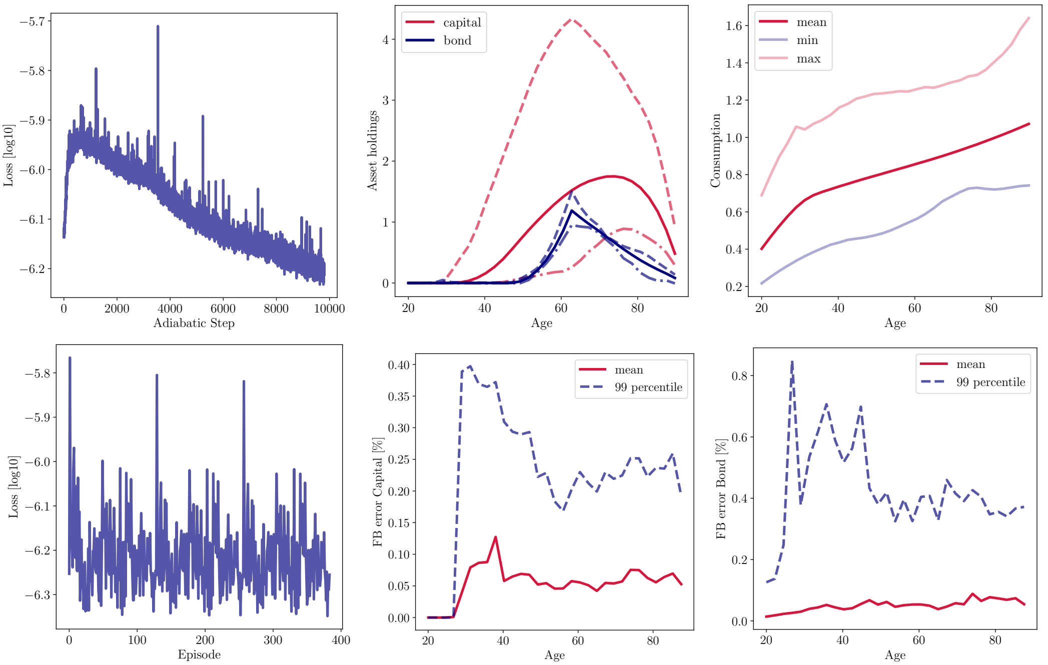
Loss [log10]
Adiabatic Step
capital
bond
Asset holdings
Age
mean
min
max
Consumption
Age
Loss [log10]
Episode
mean
99 percentile
FB error Capital [%]
Age
mean
99 percentile
FB error Bond [%]
Age


Loss [log10]
Adiabatic Step
capital
bond
Asset holdings
Age
mean
min
max
Consumption
Age
Loss [log10]
Episode
mean
99 percentile
FB error Capital [%]
Age
mean
99 percentile
FB error Bond [%]
Age


Loss [log10]
Adiabatic Step
capital
bond
Asset holdings
Age
mean
min
max
Consumption
Age
Loss [log10]
Episode
mean
99 percentile
FB error Capital [%]
Age
mean
99 percentile
FB error Bond [%]
Age



Loss [log10]
Adiabatic Step
capital
bond
Asset holdings
Age
mean
min
max
Consumption
Age
Loss [log10]
Episode
mean
99 percentile
FB error Capital [%]
Age
mean
99 percentile
FB error Bond [%]
Age



Loss [log10]
Adiabatic Step
capital
bond
Asset holdings
Age
mean
min
max
Consumption
Age
Loss [log10]
Episode
mean
99 percentile
FB error Capital [%]
Age
mean
99 percentile
FB error Bond [%]
Age

Step 4: some more training with the final supply

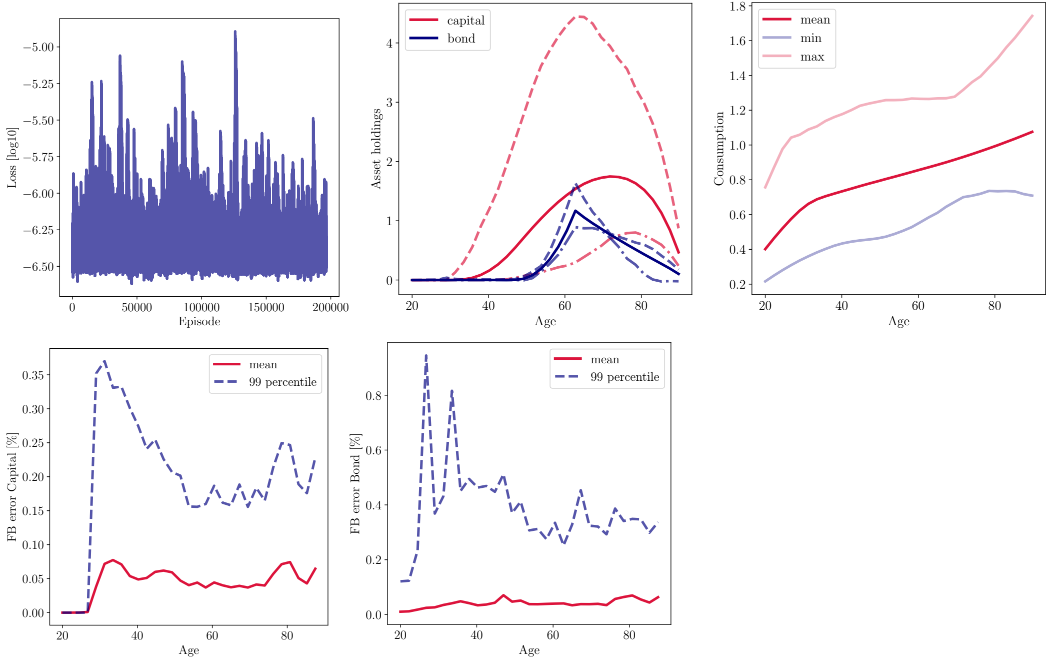
- Borrowing constraint $\underline{b} = 0$, bond at full net-supply from $B = 10$
- Neural network predicts

$$\mathcal{N}_{\boldsymbol{\rho}}^{\text{pre}}(\mathbf{x}_t) = [\hat{k}^1_{t+1}, \dots, \hat{k}^{32}_{t+1}, \underbrace{0.01 \times \tilde{b}^1_{t+1}, \dots, 0.01 \times \tilde{b}^{32}_{t+1}}_{\text{bond policies active}}, \hat{p}^b_t] \tag{42}$$

$$\Rightarrow \mathcal{N}_{\boldsymbol{\rho}}(\mathbf{x}_t) = [\hat{k}^1_{t+1}, \dots, \hat{k}^{32}_{t+1}, \underbrace{\hat{b}^1_{t+1}, \dots, \hat{b}^{32}_{t+1}}_{\text{always add up the B}}, \hat{p}^b_t] \tag{43}$$

- Loss function contains all remaining equilibrium conditions

$$\ell_{\boldsymbol{\rho}}(\mathbf{x}_t) := 1 \times \underbrace{\left(\sum_{h=1}^{H-1} \left(\epsilon_t^{k,h} \right)^2 \right)}_{\text{opt. cond. cap.}} + 1 \times \underbrace{\left(\sum_{h=1}^{H-1} \left(\epsilon_t^{b,h} \right)^2 \right)}_{\text{opt. cond. bond}} \tag{44}$$



Loss [log10]
−5.00
−5.25
−5.50
−5.75
−6.00
−6.25
−6.50
0
50000
100000
150000
200000
Episode
capital
bond
Asset holdings
0
1
2
3
4
20
40
60
80
Age
mean
min
max
Consumption
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
20
40
60
80
Age
mean
99 percentile
FB error Capital [%]
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
20
40
60
80
Age
mean
99 percentile
FB error Bond [%]
0.0
0.2
0.4
0.6
0.8
20
40
60
80
Age

More Challenging Application

Model

Additional to capital and bond as in simple model:

- Include fluctuations in the size of the incoming cohort
  - size of incoming cohort m_t^1 follows an AR(2)
  - mass-distribution $[m_t^1, \ldots, m_t^H]$ becomes start of the state-space
  - expectation now over ϵ_{t+1}^m and ϵ_{t+1}^Z

Model

Additional to capital and bond as in simple model:

- Include fluctuations in the size of the incoming cohort
  - size of incoming cohort m_t^1 follows an AR(2)
  - mass-distribution $[m_t^1, \ldots, m_t^H]$ becomes start of the state-space
  - expectation now over ϵ_{t+1}^m and ϵ_{t+1}^Z
- Add land as a third asset
  - Production function: $Y_t = Z_t K_t^{\alpha_K} L_t^{\alpha_L} N_t^{1-\alpha_K-\alpha_L}$
  - additional asset price: p_t^L
  - $H-1$ new state-variables: l_t^h
  - $H-1$ new policies: l_{t+1}^{h+1}
  - $H-1$ new optimality conditions: FOC's for land choice

Model

Additional to capital and bond as in simple model:

- Include fluctuations in the size of the incoming cohort
  - size of incoming cohort m_t^1 follows an AR(2)
  - mass-distribution $[m_t^1, \ldots, m_t^H]$ becomes start of the state-space
  - expectation now over ϵ_{t+1}^m and ϵ_{t+1}^Z
- Add land as a third asset
  - Production function: $Y_t = Z_t K_t^{\alpha_K} L_t^{\alpha_L} N_t^{1-\alpha_K-\alpha_L}$
  - additional asset price: p_t^L
  - $H-1$ new state-variables: l_t^h
  - $H-1$ new policies: l_{t+1}^{h+1}
  - $H-1$ new optimality conditions: FOC's for land choice
- Introduce Epstein Zin utility
  - need to approximate $H-1$ value functions: V_t^h
  - $H-1$ new equilibrium conditions: Bellman equations
- Adjustment costs on aggregate capital (via a financial intermediary)

Model

- State of the economy:

$$\mathbf{x}_t = [\underbrace{Z_t}_{\text{TFP}}, \underbrace{m_t^1, \ldots, m_t^H}_{\text{mass distribution}}, \underbrace{k_t^1, \ldots, k_t^H}_{\text{capital distribution}}, \underbrace{b_t^1, \ldots, b_t^H}_{\text{bond distribution}}, \underbrace{l_t^1, \ldots, l_t^H}_{\text{land distribution}}]$$

- Equilibrium functions to approximate

$$\mathcal{N}_\rho(\mathbf{x}_t) \approx \mathbf{f}(\mathbf{x}_t) = [\underbrace{p_t^b, p_t^L}_{\text{prices}}, \underbrace{V_t^1, \ldots, V_t^{H-1}}_{\text{value functions}}, \underbrace{k_{t+1}^2, \ldots, k_{t+1}^H}_{\text{capital savings}}, \underbrace{b_{t+1}^2, \ldots, b_{t+1}^H}_{\text{bond savings}}, \underbrace{l_{t+1}^2, \ldots, l_{t+1}^H}_{\text{land savings}}]$$

- Loss function

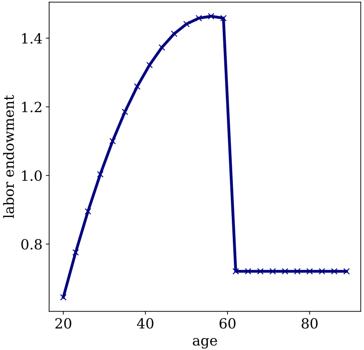
$$\ell_\rho(\mathbf{x}_t) := \underbrace{\left(\sum_{h=1}^{H-1} \left(\epsilon_t^{k,h}\right)^2\right)}_{\text{opt. cond. cap.}} + \underbrace{\left(\sum_{h=1}^{H-1} \left(\epsilon_t^{b,h}\right)^2\right)}_{\text{opt. cond. bond}} + \underbrace{\left(\sum_{h=1}^{H-1} \left(\epsilon_t^{l,h}\right)^2\right)}_{\text{opt. cond. land}} + \underbrace{\left(\sum_{h=1}^{H-1} \left(\epsilon_t^{\text{Bellman},h}\right)^2\right)}_{\text{rel. Bellman error}}$$

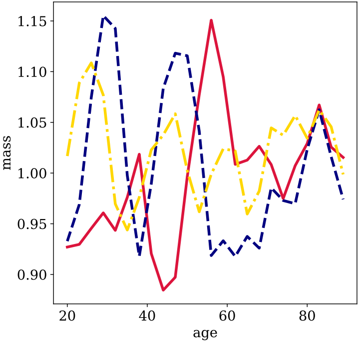
$$+ \cancelto{=0}{\underbrace{\left(\epsilon_t^{MCB}\right)^2}_{\text{market clearing bond}}} + \cancelto{=0}{\underbrace{\left(\epsilon_t^{MCL}\right)^2}_{\text{market clearing land}}}$$

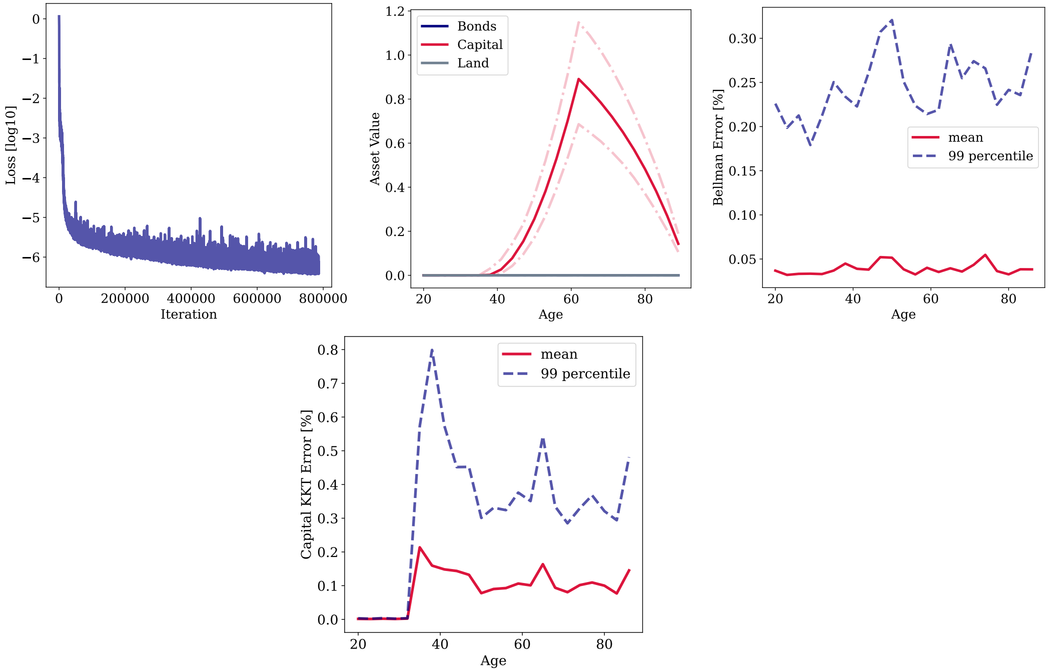
Parameters

Parameters	H	β	γ	ψ^{hh}	$\zeta^{k,hh}$	$\zeta^{l,hh}$	$\zeta^{K,hh}$	$\underline{b}$
Values	24	0.885	5	0.5	0.02	0.2	0.3	-0.05
Meaning	age groups	patience	RRA	inv. eis	adj. c. cap.	adj. c. land	adj. c. agg. cap.	bor. const.

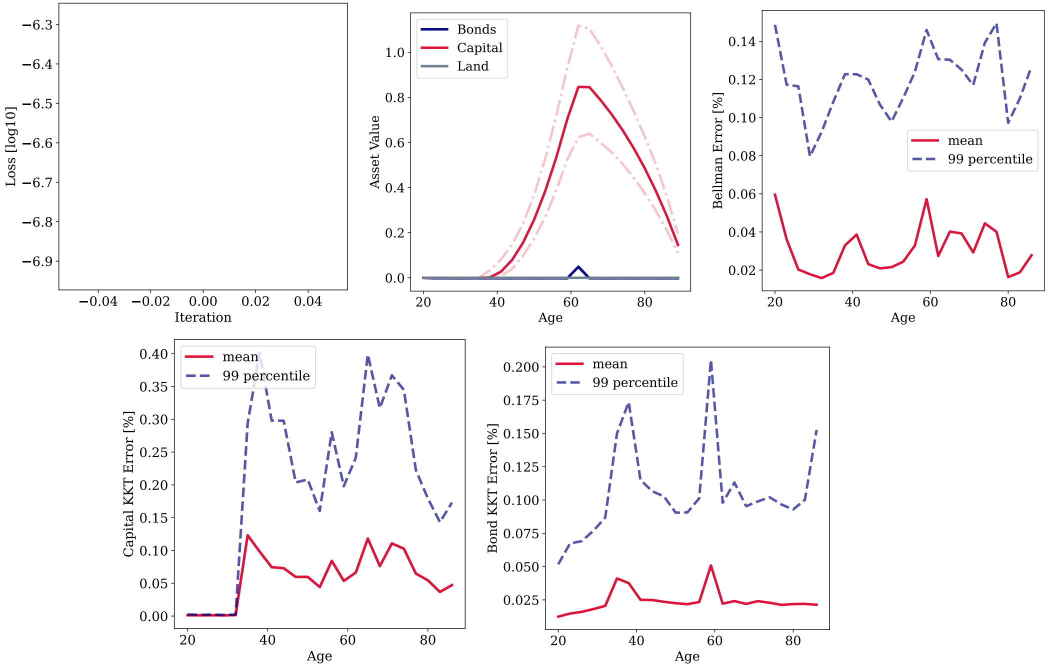
Parameters	ρ^z	σ^z	ρ_1^m	ρ_2^m	σ^m	L	α_L	α_K	δ_L	δ_K
Values	0.614	0.045	0.948	-0.588	0.0256	1	0.1	0.23	0.271	0.271
Meaning	pers. tfp	std. tfp	size inc.	size inc	std. size inc.	l. supp.	share land	share cap	maint. land	depr. cap

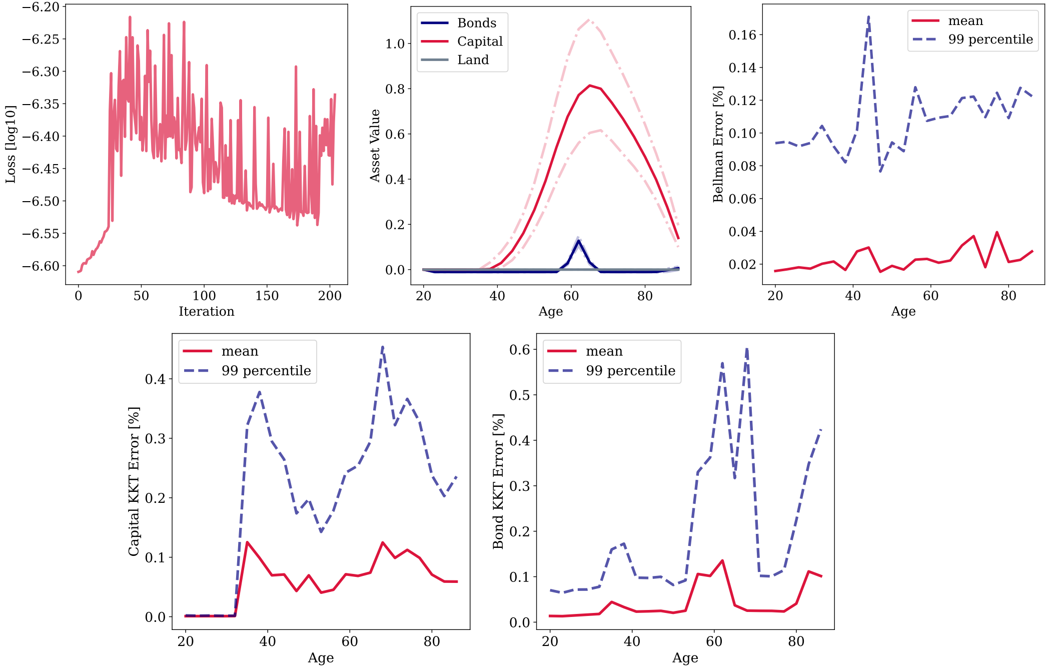


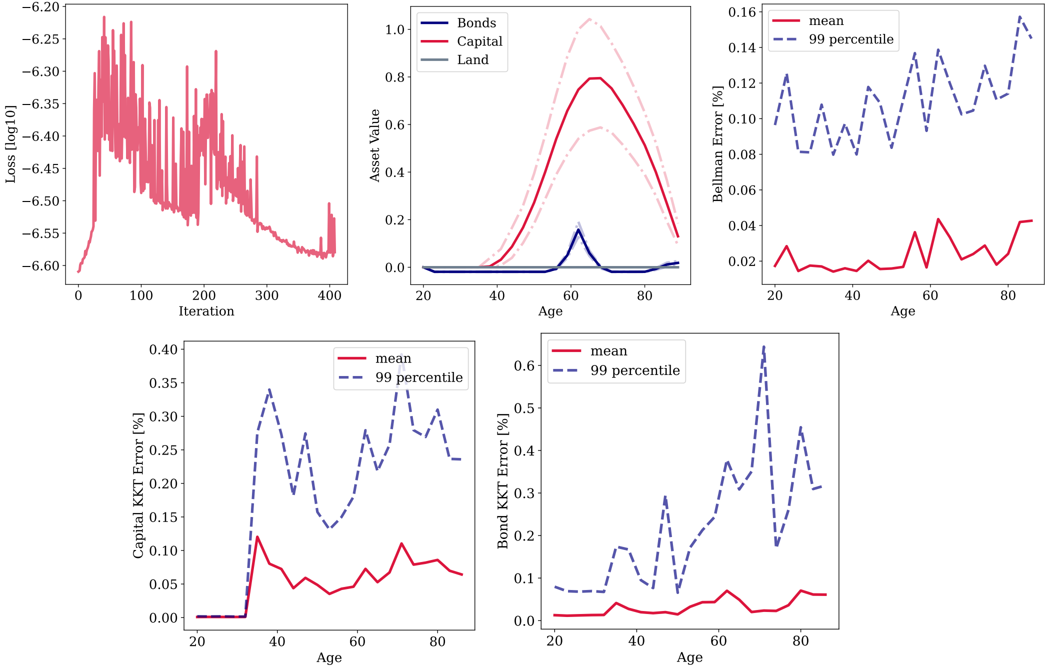




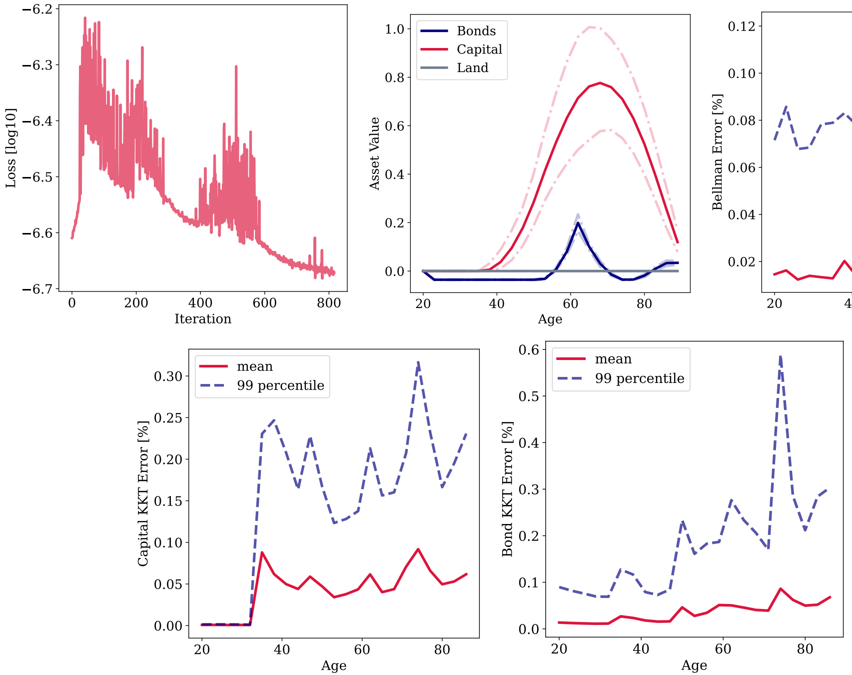
Loss [log10]
Iteration
0
−1
−2
−3
−4
−5
−6
0
200000
400000
600000
800000
Asset Value
Age
Bonds
Capital
Land
Bellman Error [%]
mean
99 percentile
Capital KKT Error [%]


Loss [log10]
Iteration
−6.3
−6.4
−6.5
−6.6
−6.7
−6.8
−6.9
−0.04
−0.02
0.00
0.02
0.04
Asset Value
Age
Bonds
Capital
Land
Bellman Error [%]
Age
mean
99 percentile
Capital KKT Error [%]
Age
mean
99 percentile
Bond KKT Error [%]
Age
mean
99 percentile

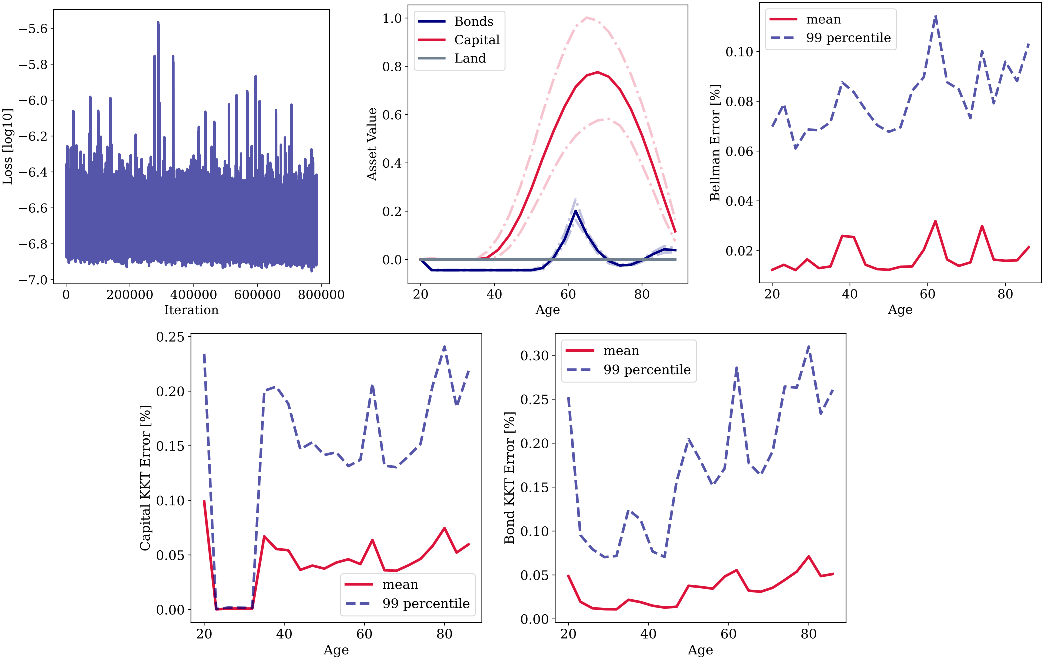

Loss [log10]
Iteration
Asset Value
Age
Bonds
Capital
Land
Bellman Error [%]
mean
99 percentile
Capital KKT Error [%]
Bond KKT Error [%]

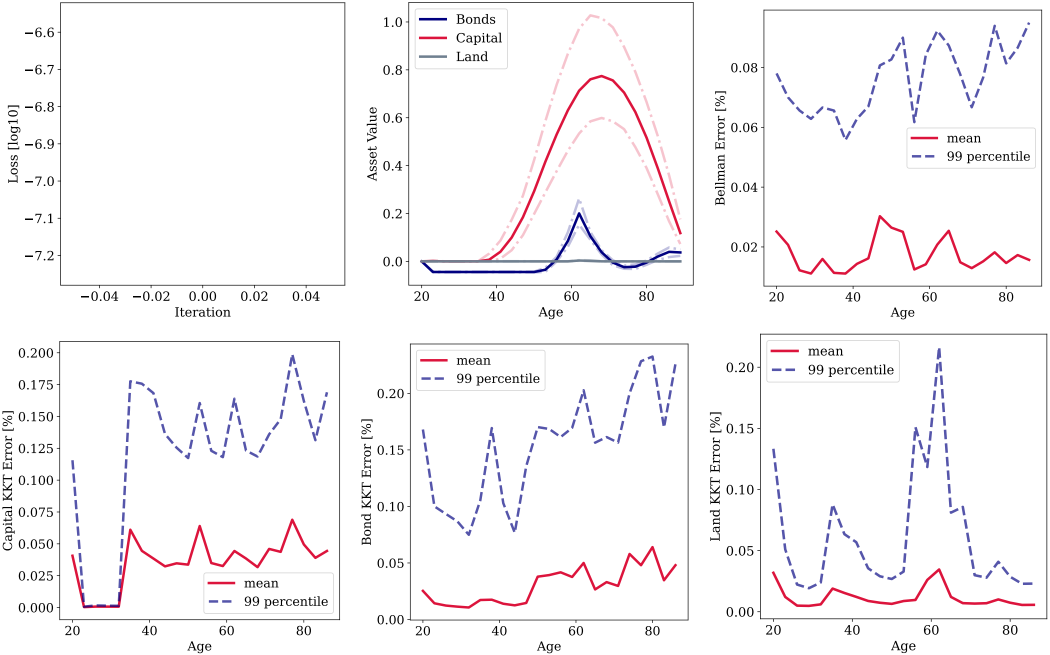


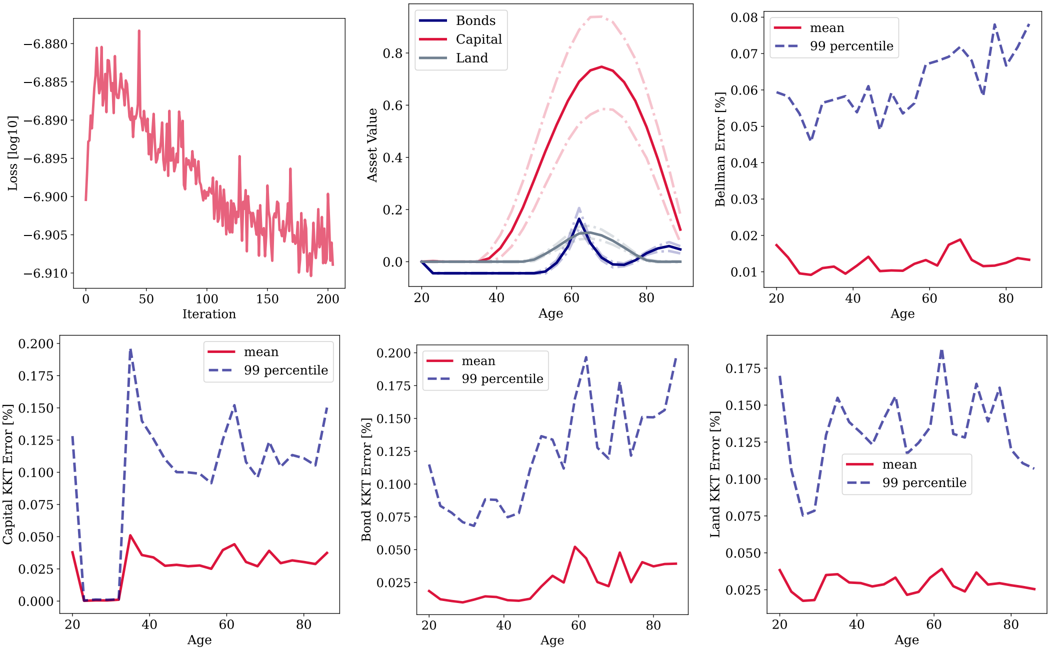
Loss [log10]
Iteration
Asset Value
Age
Bonds
Capital
Land
Bellman Error [%]
Age
mean
99 percentile
Capital KKT Error [%]
Age
mean
99 percentile
Bond KKT Error [%]
Age
mean
99 percentile



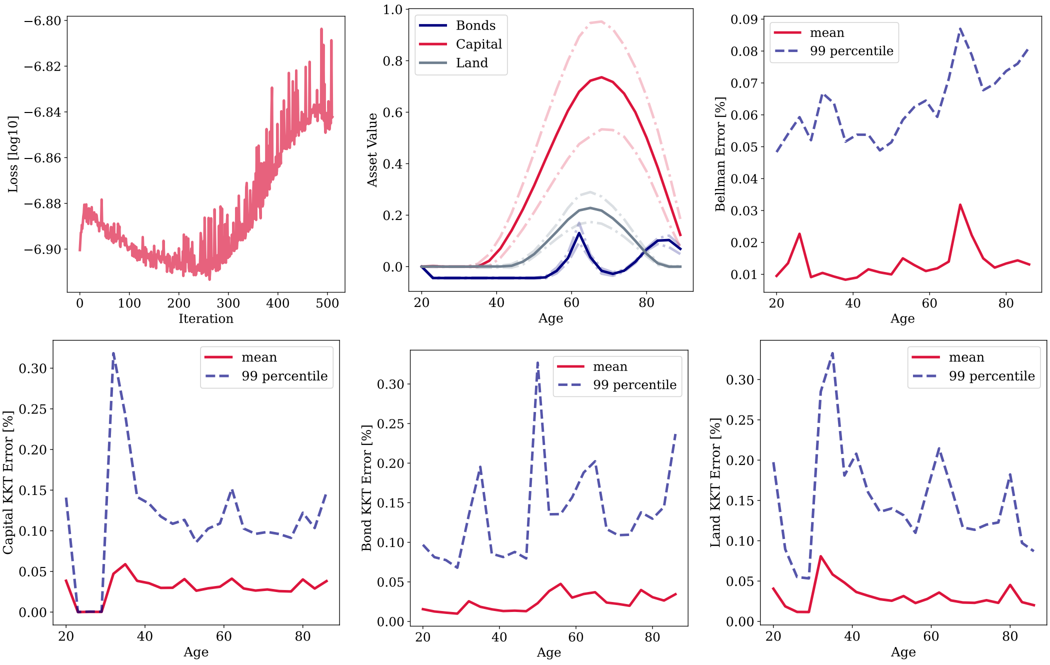
Loss [log10]
Iteration
Asset Value
Bonds
Capital
Land
Age
Bellman Error [%]
mean
99 percentile
Capital KKT Error [%]
Bond KKT Error [%]


Loss [log10]
Iteration
Asset Value
Age
Bonds
Capital
Land
Bellman Error [%]
mean
99 percentile
Capital KKT Error [%]
Bond KKT Error [%]

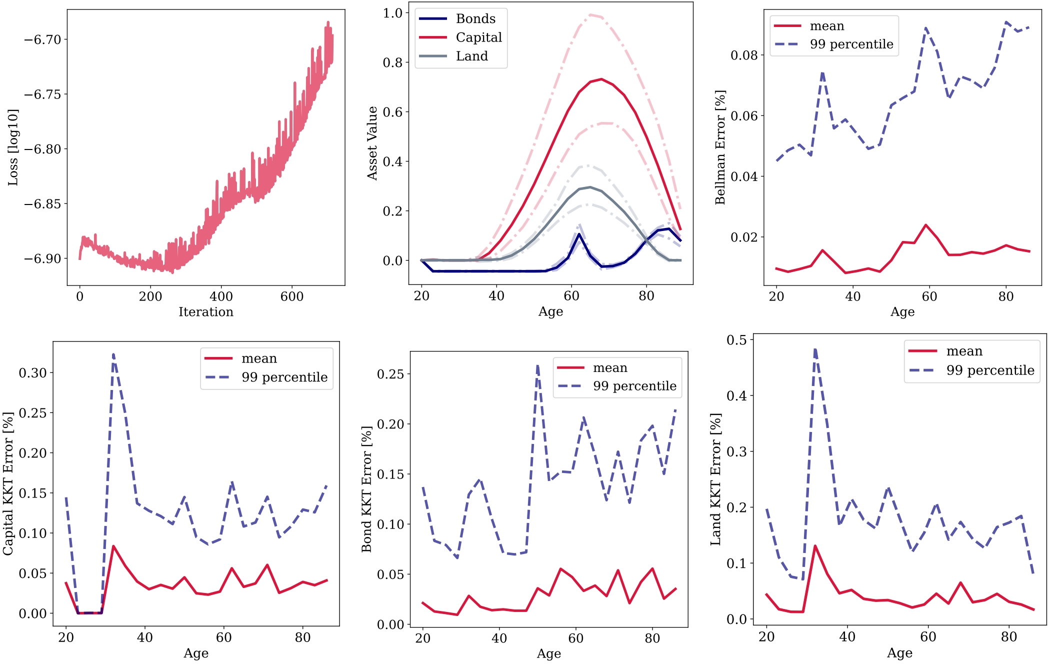

Loss [log10]
Iteration
Asset Value
Age
Bonds
Capital
Land
Bellman Error [%]
mean
99 percentile
Capital KKT Error [%]
Bond KKT Error [%]
Land KKT Error [%]

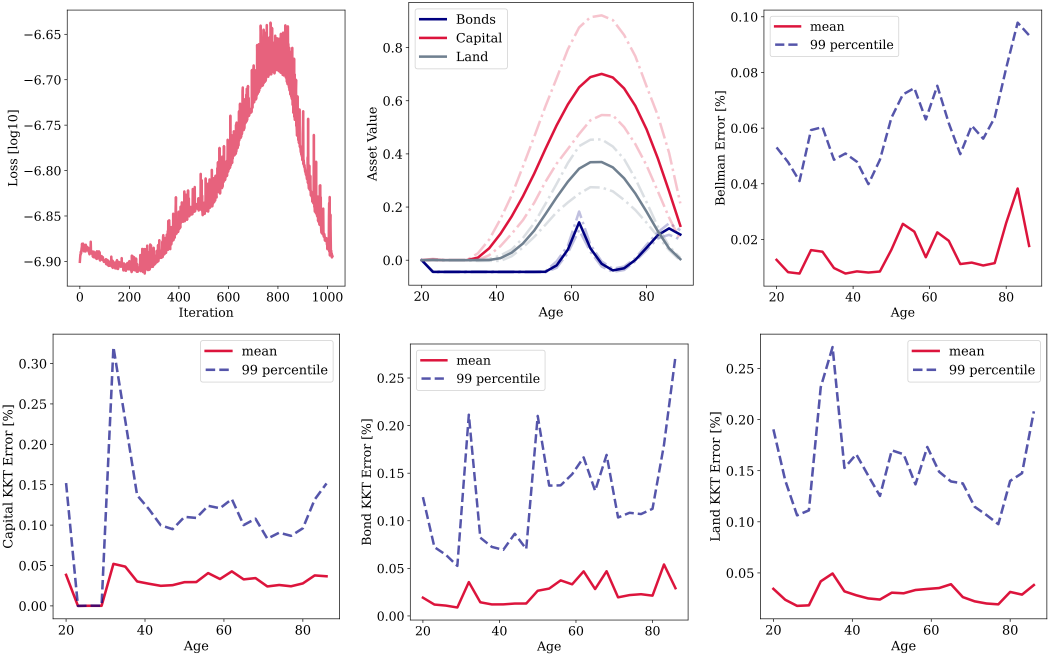


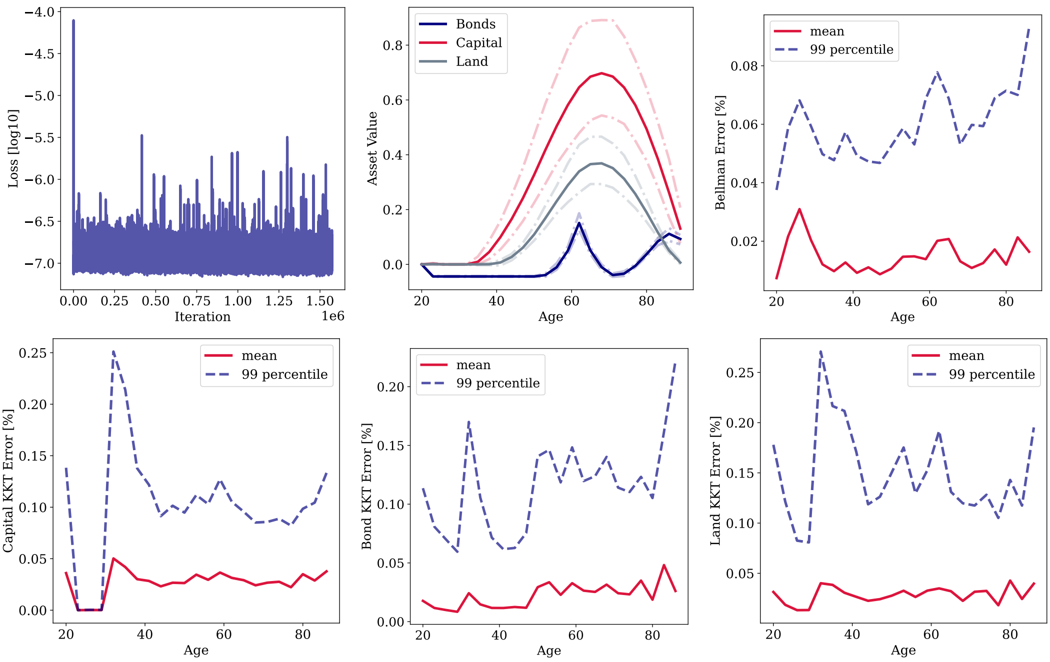
Loss [log10]
Iteration
Asset Value
Age
Bonds
Capital
Land
Bellman Error [%]
mean
99 percentile
Capital KKT Error [%]
Bond KKT Error [%]
Land KKT Error [%]



Loss [log10]
Iteration
Asset Value
Bonds
Capital
Land
Age
Bellman Error [%]
mean
99 percentile
Capital KKT Error [%]
Bond KKT Error [%]
Land KKT Error [%]


Loss [log10]
−6.70
−6.75
−6.80
−6.85
−6.90
0
200
400
600
Iteration
Asset Value
Bonds
Capital
Land
1.0
0.8
0.6
0.4
0.2
0.0
20
40
60
80
Age
Bellman Error [%]
mean
99 percentile
0.08
0.06
0.04
0.02
20
40
60
80
Age
Capital KKT Error [%]
mean
99 percentile
0.30
0.25
0.20
0.15
0.10
0.05
0.00
20
40
60
80
Age
Bond KKT Error [%]
mean
99 percentile
0.25
0.20
0.15
0.10
0.05
0.00
20
40
60
80
Age
Land KKT Error [%]
mean
99 percentile
0.5
0.4
0.3
0.2
0.1
0.0
20
40
60
80
Age


Loss [log10]
Iteration
Bonds
Capital
Land
Asset Value
Age
mean
99 percentile
Bellman Error [%]
Capital KKT Error [%]
Bond KKT Error [%]
Land KKT Error [%]


Loss [log10]
Iteration
1e6
Asset Value
Age
Bonds
Capital
Land
Bellman Error [%]
mean
99 percentile
Capital KKT Error [%]
Bond KKT Error [%]
Land KKT Error [%]

Conclusion

- We develop a deep learning solution method for solving large stochastic models with portfolio choice
- Two key innovations
  - Market Clearing Layers, an economics-inspired neutral network architecture
  - Adiabatic model transformation procedure to guide network training with many assets

Thank you!

Feedback is welcome!

azinovic@sas.upenn.edu
jan.zemlicka@cerge-ei.cz

References I

Azinovic, M., Gaegauf, L., and Scheidegger, S. (2022). Deep equilibrium nets. *International Economic Review*, 63(4):1471–1525.

Ebrahimi Kahou, M., Fernández-Villaverde, J., Perla, J., and Sood, A. (2021). Exploiting symmetry in high-dimensional dynamic programming. Working Paper 28981, National Bureau of Economic Research.

Fernández-Villaverde, J., Hurtado, S., and Nuno, G. (2023). Financial frictions and the wealth distribution. *Econometrica*, 91(3):869–901.

Gu, Z., Lauriere, M., Merkel, S., and Payne, J. (2023). Deep learning solutions to master equations for continuous time heterogeneous agent macroeconomic models.

Han, J., Yang, Y., et al. (2021). Deepham: A global solution method for heterogeneous agent models with aggregate shocks. *arXiv preprint arXiv:2112.14377*.

Kase, H., Melosi, L., and Rottner, M. (2022). Estimating nonlinear heterogeneous agents models with neural networks. *CEPR Discussion Paper No. DP17391*.

Maliar, L., Maliar, S., and Winant, P. (2021). Deep learning for solving dynamic economic models. *Journal of Monetary Economics*, 122:76–101.

Valaitis, V. and Villa, A. (2021). A machine learning projection method for macro-finance models. *Available at SSRN 4119888*.

Deep Neural Networks

What is a deep neural net? ▸ DEQN

A neural net is a sequence of affine models interleaved by element-wise nonlinear transformations

$$
\begin{aligned}
\textbf{input} := \mathbf{x} &\to \phi^1(W^1_{\boldsymbol{\rho}}\mathbf{x} + \mathbf{b}^1_{\boldsymbol{\rho}}) =: \textbf{hidden 1} \\
\to \textbf{hidden 1} &\to \phi^2(W^2_{\boldsymbol{\rho}}(\textbf{hidden 1}) + \mathbf{b}^2_{\boldsymbol{\rho}}) =: \textbf{hidden 2} \\
\to \textbf{hidden 2} &\to \phi^3(W^3_{\boldsymbol{\rho}}(\textbf{hidden 2}) + \mathbf{b}^3_{\boldsymbol{\rho}}) =: \textbf{output}
\end{aligned}
$$

The neural net is given by the choice of activation functions $\{\phi^i\}$ and the parameters $\boldsymbol{\rho}$.