Developing an Agent-Based Model of the Banking System

Georgi Demirev

Bulgarian National Bank Scholarship Recipient

georgi.demirev@barcelonagse.eu

January 14, 2020

Overview

Motivation - DSGEs vs ABMs

DSGE Models

- Well understood and widely adopted
- Theoretically appealing and interpretable
- Rely on exogenous shocks
- Assume perfect optimization
- Equilibrium assumption
- Sometimes require linearization

Agent-Based Models

- Rely for simulation instead of requiring solving
- Allow for higher complexity and non-linear dynamics
- Not that well understood or theoretically rigorous
- Assume that agents follow simple behavior rules

Reinforcement Learning

- A subfield of CS and AI
- Adaptive approximation of solutions to dynamic programming problems
- Handles large state and action spaces
- Bounded rationality

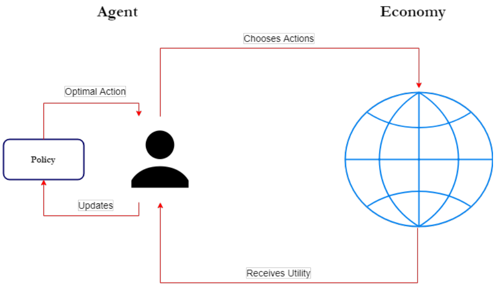
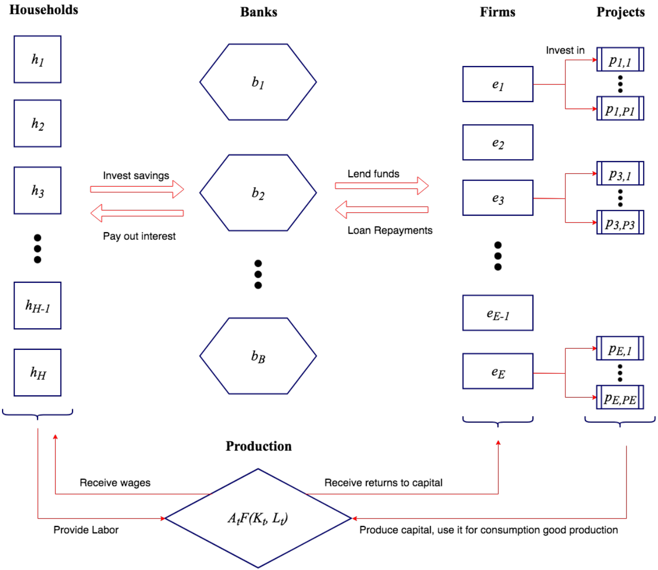


Figure: Agent interacting with the economy

Circular Flow



Agents

Households:

$$\max_{w_{i,t}, s_{i,t}} \sum_t \beta_H^t \mathbf{E}\left[\log c_h(w_{i,t}, s_{i,t}, \mathbf{A_{-i}}, \mathbf{S_t})\right] \tag{1}$$

Firms:

$$\max_{e_{j,t}, b_{j,t}} \sum_t \beta_F^t \mathbf{E}\left[\log c_f(e_{j,t}, d_{j,t}, \mathbf{A_{-i}}, \mathbf{S_t})\right] \tag{2}$$

Banks:

$$\max_{l_{k,t}, d_{k,t}, a_{k,t}} \sum_t \beta_B^t \mathbf{E}\left[\log c_b(l_{k,t}, d_{k,t}, a_{k,t}, \mathbf{A_{-i}}, \mathbf{S_t})\right] \tag{3}$$

Interactions

1. Production of the consumption good. Wages and rental rates are paid
2. Potential bank bankruptcies are resolved
3. Banks approve or reject loan applications
4. Banks adjust their interest rates and approval cut-off rate
5. Banks give out dividends to shareholders (receive utility)
6. Firms receive capital from outstanding projects and draw a new project opportunity
7. Firms make loan repayments
8. Firms decide whether to invest in the new project
9. Firms consume leftover cash
10. Households decide whether to keep their deposits
11. Households receive interest on deposits

Learning by Interaction

- Agents observe a subset $s_t \in S_t$ of the state at each time step t
- The goal of each agent in the economy is to learn a mapping between states (observation sets) s_t and actions a_t (the action set is discrete). Call this mapping (policy) π^*
- All agents begin in a 'blank slate' state, following a random policy π_0^*.
- As the simulation unfolds, agents observe recieved utility and incrementally update their policy

Q Function

Action value function

$Q^{\pi}(s, a)$ denotes the expected discounted stream of future rewards (i.e. utility), following state s in which action a is taken, and policy π is followed afterwards

$$Q^{\pi}(s, a) = u_0(s, a) + \beta E_{s^t \sim \pi}\left[\sum_{t=1}^{\infty} u_t(s^t, \pi(s^t))\right] = u_0(s, a) + \beta E_{s^t \sim \pi}\left[Q^{\pi}(s', \pi(s'))\right] \quad (4)$$

Where $\pi(s)$ is the action prescribed by the policy π at state s, u_t is the utility at period t, β is the discount factor, $s^t \sim \pi$ denotes the sequence of future states under the policy, and s' denotes the very next state.

Q Learning

Q-learning relates to a general algorithm of finding an approximate solution to the dynamic programming program, given by the Bellman equation:

$$Q^*(s,a) = max_a u(s,a) + \beta E_{s,a} V^*(s') \tag{5}$$

Where

$$V^*(s') = max_{a'} Q(s', a') \tag{6}$$

With a' being any action available at s'.

Function Approximation

Q Approximation function

Define the differentiable function Q_ϕ (parametrized by ϕ) as the agent's current approximation to Q^*.

Bellman error

Define the Bellman error (λ) at state s_i, after action a_i as:

$$\lambda_i = Q_\phi(s_i, a_i) - u(s_i, a_i) - \beta V_\phi(s_i', a_i') = Q_\phi(s_i, a_i) - u(s_i, a_i) - \beta \max_{a_i'} Q_\phi(s_i', a_i') \quad (7)$$

Learning Algorithm

Offline

1. Collect data $(s_i, a_i, s'_i, u(s_i, a_i))$
2. Set $y_i = u(s_i, a_i) + \beta \max_{a'} Q_\phi(s'_i, a'_i)$
3. $\phi \leftarrow arg \min_\phi \sum_i ||Q_\phi(s_i, a_i) - y_i||^2$

Online

1. Take an action a_i at state S_i and record $S_i, a_i, S'_i, u(S_i, a_i)$
2. Calculate the Bellman error $\lambda_i = Q_\phi(S_i, a_i) - u(S_i, a_i) - \beta \max_{a'} Q_\phi(S'_i, a'_i)$
3. Set $\phi \leftarrow \phi - \gamma \nabla_\phi Q_\phi(s_i, a_i) \lambda_i$

Using Neural Networks as Approximators

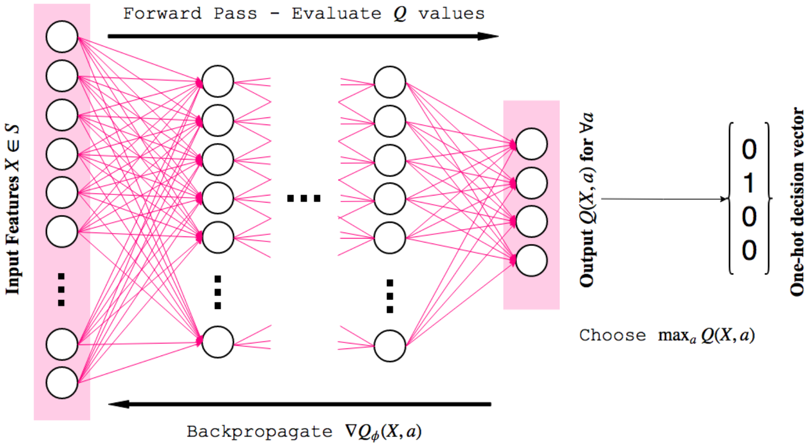


Figure: A representation of a Q-Network

Double Q Learning

1. Take action a_i, record $(S_i, a_i, S_i', u(S_i, a_i))$. Add it to $\mathcal{B}$
2. Sample mini-batch $\{S_j, a_j, S_j', u(S_j, a_j)\}_{j=1}^N$ from $\mathcal{B}$
3. Compute the Bellman errors
   $\lambda_j = Q_\phi(S_j, a_j) - u(S_j, a_j) - \beta Q_{\phi'}(S_j', arg\max_{a_j'} Q_\phi(S_j', a_j'))$
4. Update $\phi \leftarrow \phi - \gamma \sum_j \nabla_\phi Q_\phi(S_j, a_j)\lambda_j$
5. Every T steps set $\phi' \leftarrow \phi$

Simulation

- 10 banks, 100 firms, and 500 households
- 35000 'training' iterations
- 100 periods of burnin
- 1 : 1000 chance of reset
- 600 'test' periods

Cyclical Output

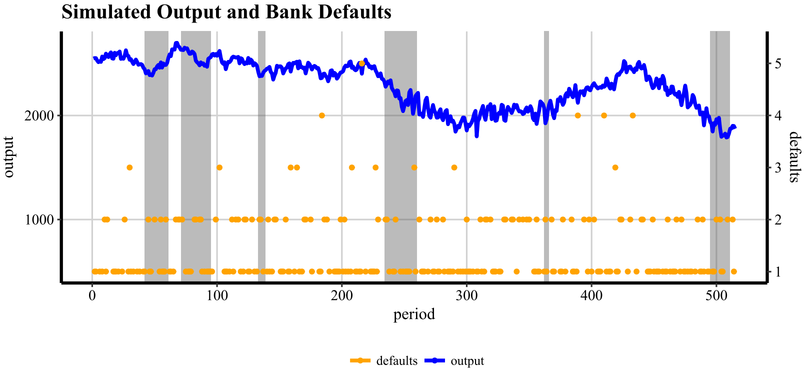
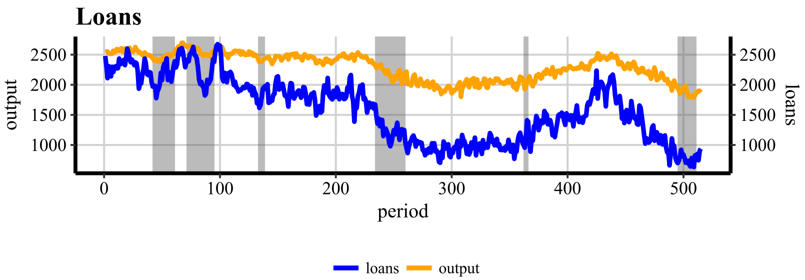
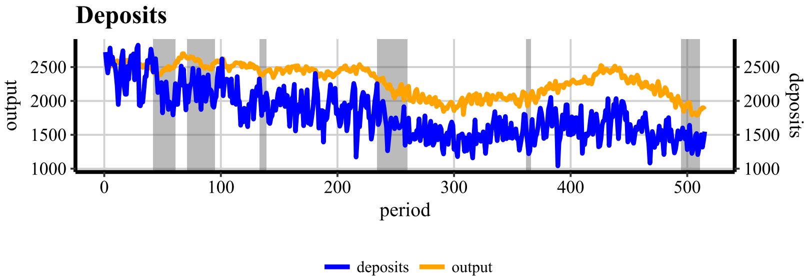


Figure: Simulated Output

Co-movements





Variance

Series	Coefficient of Variation	Auto-correlation (1 lag)	Contemporaneous correlation with output
output	0.099	0.974	1
capital	0.311	0.977	0.976
deposit interest	0.168	0.965	0.043
loan interest	0.146	0.977	-0.551
approval rate	0.007	0.975	0.523
deposits	0.201	0.824	0.747
loans	0.330	0.969	0.938
consumption	0.103	0.657	0.819
investment	0.233	0.504	0.627

Serial Correlation

Series	y_{t-3}	y_{t-2}	y_{t-1}	y_t	y_{t+1}	y_{t+2}	y_{t+3}
output	0.940	0.952	0.974	1	0.974	0.952	0.940
capital	0.933	0.942	0.955	0.976	0.991	0.968	0.948
deposit interest	0.0498	0.0483	0.0457	0.0430	0.0431	0.0431	0.0417
loan interest	-0.519	-0.530	-0.541	-0.551	-0.549	-0.547	-0.544
approval rate	0.487	0.497	0.511	0.523	0.523	0.525	0.525
deposits	0.707	0.724	0.744	0.747	0.723	0.703	0.704
loans	0.894	0.905	0.921	0.938	0.946	0.949	0.932
consumption	0.781	0.790	0.812	0.819	0.774	0.762	0.765
investment	0.576	0.578	0.581	0.627	0.645	0.651	0.615

Conducting Economic Experiments

- Introduce change in the environment or the interaction rules
- 8000 additional training period to adapt to new scenario
- 500 'test' periods simulated 3 times
- Results from the test simulations aggregated to evaluate the changes in economic aggregates

Introducing deposit guarantees

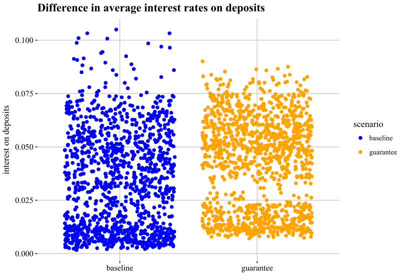


Figure: Average interest rate on deposits

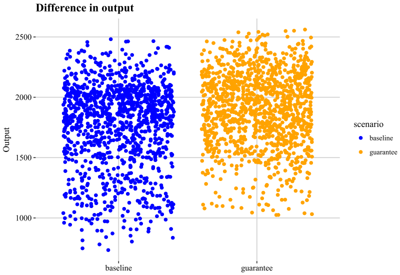


Figure: Average per-period output

Changing the required reserve ratio

Series	4%	8%	12%	16%	20%
output	2220	2108	2153	2116	2083
deposits	4323	4145	6172	8651	5134
loans	1596	1314	1425	1368	1265
consumption	1557	1573	1326	1500	1256
investment	364	340	341	341	307
deposit interest	0.027	0.042	0.034	0.028	0.043
loan interest	0.030	0.061	0.049	0.064	0.052

Areas for future research

- Policy Convergence
- Agent Sophistication
- Choosing hyper-parameters
- Continuous actions spaces
- Concurrent simulation
- Policy interpretation
- Causal Links

Q&A

Thank You!

`github.com/demirev/banks_rl`