Faculty of Economic Sciences

November 8, 2024

Solving and Estimating Non-Linear HANK Models With Machine Learning

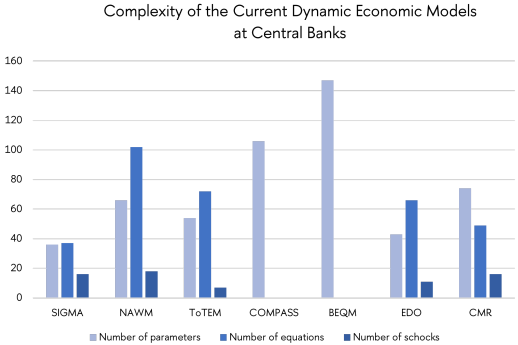
Project Proposal

Tatiana Sorokina, BST193

tssorokina@edu.hse.ru

Problem Formulation

- Central banks use DSGE models for counterfactual and policy analysis
  - Response to Lucas (1976) and Sims (1980) critique
- Inclusion of:
  - zero lower bound;
  - financial frictions;
  - agent heterogeneity;
  - and etc.
- Future DSGE models will include:
  - aggregate uncertainty and non-linearities;
  - HA facing idiosyncratic risks



$\Rightarrow$**A need for efficient global model solution methods**

Advantages of the Neural Network Framework for Model Solution

How can Machine Learning be helpful?

Machine Learning techniques became a tool for solving large heterogeneous-agent (HA) models. We build our **neural network framework** by combining simulation and projection approaches with Reinforcement Learning

Main features:

- models are solved in a non-linearised form
  - ability to keep such non-linearities as zero lower bound and occasionally binding constraints;
- global solution method, which handles easily small- and medium-scale models;
- economic agents adapt their behaviour to make an optimal decision in an evolving environment;
- easier model parameter estimation - *the model only needs to be solved once*

Neural Network Algorithm Description (1)

Building on Maliar et al. (2021) and Kase et al. (2022), the models are solved in a form of **Euler-equation residual minimisation**:

0. **If RA framework** $\rightarrow$ aggregate decision functions include RA behaviour
   If HA framework $\rightarrow$ continuum of agents is approximated with L agents, aggregate (economy-level) and individual (HA-level) decision functions are separated
1. **Individual and aggregate decision functions are approximated with neural networks:**

$$\mathbb{X}_t = \left\{ \mathbb{X}_t^A, \{\mathbb{X}_t^i\}_{i=1}^L \right\} \sim \left\{ \varphi_{NN}^A(\mathbb{S}_t, \mathbb{M}_t, \tilde{\Theta}|\bar{\Theta}), \left\{ \varphi_{NN}^I(\mathbb{S}_t, \mathbb{M}_t, \mathbb{S}_t^i, \mathbb{M}_t^A, \tilde{\Theta}|\bar{\Theta}) \right\}_{i=1}^L \right\},$$

- $\mathbb{S}_t = \{\mathbb{S}_t^A, \{\mathbb{S}_t^i\}_{i=1}^L\}$ and $\mathbb{M}_t = \{\mathbb{M}_t^A, \{\mathbb{M}_t^i\}_{i=1}^L\}$ are vectors of endogenous and exogenous state variables, divised into aggregate and individual parts
- $\bar{\Theta}$ and $\tilde{\Theta}$ are the vectors of calibrated and estimated model parameters respectively

Neural Network Algorithm Description (2)

The models are solved in a form of Euler-equation residual minimisation:

1. Individual and aggregate decision functions are approximated with neural networks
2. **Loss function for the neural network training is defined as a weighted sum of equilibrium conditions:**
   - aggregate and individual Euler equation residuals,
   - market clearing conditions
   - differentiable representation of the complementary slackness conditions

   All-in-One (AiO) expectation parameter (Maliar et al., 2021) and simulation procedure are used instead of iterating over a rigid grid
   - General idea of the AiO expectation parameter is the representation:

$$\mathbb{E}_{\mathbb{M}_t,\mathbb{S}_t,\epsilon_1}[f(\mathbb{M}_t,\mathbb{S}_t,\epsilon_1)]\cdot\mathbb{E}_{\mathbb{M}_t,\mathbb{S}_t,\epsilon_2}[f(\mathbb{M}_t,\mathbb{S}_t,\epsilon_2)]=\mathbb{E}_{(\mathbb{M}_t,\mathbb{S}_t,\epsilon_1,\epsilon_2)}[f(\mathbb{M}_t,\mathbb{S}_t,\epsilon_1)\cdot f(\mathbb{M}_t,\mathbb{S}_t,\epsilon_2)]$$

   where $f(\cdot)$ is an optimality condition, ϵ_1 and ϵ_2 are independently drawn shocks

Neural Network Algorithm Description (3)

The models are solved in a form of Euler-equation residual minimisation:

1. Individual and aggregate decision functions are approximated with neural networks
2. Loss function for the neural network training is defined as a weighted sum of equilibrium conditions
3. **The neural network is trained using stochastic optimisation:**
   - state variables $\{\mathbb{S}_t, \mathbb{M}_t\}$ are drawn from the ergodic state;
   - shocks $\{\epsilon_1, \epsilon_2\}$ are drawn according to their distribution;
   - estimated parameters $\tilde{\Theta}$ are drawn from the predefined model parameter space;
   - the neural network weights are updated using anti-gradient of the empirical loss function;
   - economy is simulated forward to generate a new draw of the state variables

Example 1: Numerical Analysis of a Consumption-Saving Problem

Optimisation Problem

Agent is choosing their current consumption and next-period wealth $\{c_t, w_{t+1}\}_{t=0}^{\infty}$ in a following problem with a borrowing constraint:

$$
\begin{gathered}
\max_{\{c_t, w_{t+1}\}_{t=0}^{\infty}} \mathbb{E}_t \left[\sum_{\tau=t}^{\infty} B_\tau \beta^{\tau-t} \frac{1}{1-\gamma} \left(c_\tau^{1-\gamma} - 1 \right) \right] \\
\text{s.t. } w_{t+1} = (w_t - c_t) \cdot \bar{R} \cdot R_{t+1} + Y_{t+1}, \\
c_t \leq w_t;
\end{gathered}
$$

Logarithms of real return on investment R_t shock, transitory Q_t and permanent P_t production components ($Y_t = Q_t \cdot P_t$), and preference shock B_t follow AR(1) process with normally-distributed independent shocks $\{\epsilon_r, \epsilon_q, \epsilon_p, \epsilon_b\} \sim N(0, \Sigma)$

Example 1: Numerical Analysis of a Consumption-Saving Problem

Parametrisaton

1. **Individual decision function is approximated with a neural network:**

$$\begin{pmatrix} A \\ B \end{pmatrix} = \varphi_{NN}\left(\underbrace{w_t}_{\mathbb{S}_t}, \underbrace{r_t, p_t, q_t, b_t}_{\mathbb{M}_t}, \underbrace{\beta, \gamma, \bar{r}, \rho_r, \sigma_r, \rho_p, \sigma_p, \rho_q, \sigma_q, \rho_b, \sigma_b}_{\tilde{\Theta}} \right)$$

$$A \equiv \frac{c_t}{w_t},\ 0 \leq A \leq 1;\ B \equiv \beta \bar{R} \mathbb{E}_\epsilon \left[\frac{u'(c_{t+1})}{u'(c_t)} \cdot \frac{B_{t+1} \cdot R_{t+1}}{B_t} \right] > 0$$

- 500 simultaneous batch-economies
- 10,000 iterations
- 3 fully-connected layers with a SiLU (Elfwing et al., 2017) activation function

2. **Loss function for the neural network training is defined as a weighted sum of equilibrium conditions:** (here, v is the weight of the complementary slackness condition):

$$F_1(\mathbb{S}_t, \mathbb{M}_t, \epsilon, \tilde{\Theta}) = \beta \bar{R} \mathbb{E}_\epsilon \left[\frac{u'\left(c_{t+1}\right)}{u'(c_t)} \cdot \frac{B_{t+1} \cdot R_{t+1}}{B_t} \right] - B;$$

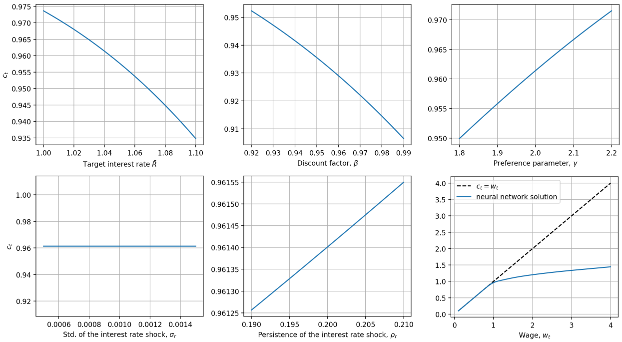
$$F_2\left(\mathbb{S}_t, \mathbb{M}_t, \tilde{\Theta}\right) = F^{FB}(1 - A, 1 - B),$$

$$Loss(\varphi_{NN}) = F_1\left(\mathbb{S}_t, \mathbb{M}_t, \epsilon_1, \tilde{\Theta}\right) F_1\left(\mathbb{S}_t, \mathbb{M}_t, \epsilon_2, \tilde{\Theta}\right) + v\left(F_2\left(\mathbb{S}_t, \mathbb{M}_t, \tilde{\Theta}\right)\right)^2$$

3. **The neural network is trained using stochastic optimisation:**
   - optimiser Adam (Kingma and Ba, 2017)

Example 1: Numerical Analysis of a Consumption-Saving Problem

Addressing the "black box" critique of the neural networks



Consumption elasticity depending on the values of: target interest rate $\bar{R}$, discounting factor β, preference parameter γ, standard deviation σ_r and persistence of the interest rate shock ρ_r. The last figure depicts the consumption decision rule conditioned on the initial wealth endowment w_t

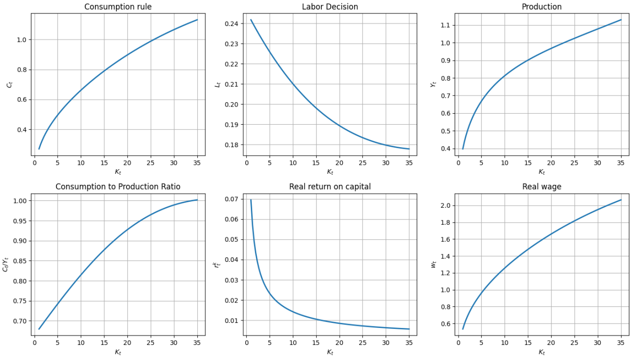
Let us consider a **New Keynesian DSGE** model with the following equilibrium:[1]

Intertemporal consumption substitution	$1-\mu_t = \beta \mathbb{E}_t \frac{C_t}{C_{t+1}}(1+r^k_{t+1}-\delta),$
Consumption-labour substitution	$w_t = \frac{1-\gamma}{\gamma}\cdot\frac{C_t}{1-L_t}$
Marginal real cost of capital	$r^k_t = \alpha\frac{\zeta-1}{\zeta}\cdot\frac{Y_t}{K_t}$
Marginal real cost of labour	$w_t = (1-\alpha)\frac{\zeta-1}{\zeta}\cdot\frac{Y_t}{L_t}$
Production function for the intermediate good	$Y_t = A_t K_t^{\alpha} L_t^{(1-\alpha)}$
The capital transition from the household problem	$K_{t+1} = (1-\delta)K_t + I_t$
Economy budget constraint	$Y_t = C_t + I_t$
Transition function for the TFP	$\ln A_t = (1-\rho_A)\bar{A} + \rho_A \ln A_{t-1} + \epsilon^A_t$

[1] source: the HSE course on Business Cycles by Oxana Malakhovskaya (2021)

Example 2: NK DSGE model with Monopolistic Competition

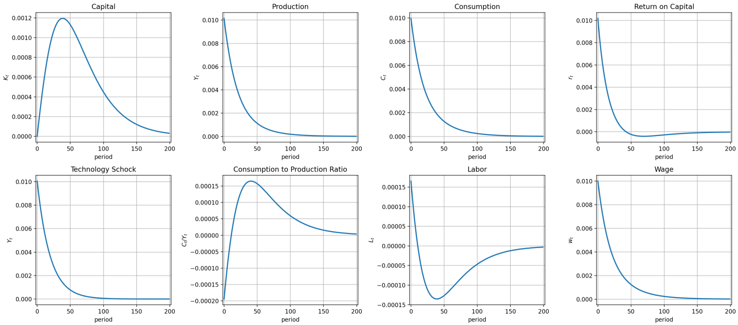
Neural network solution



Neural network approximations of the control variables $\mathbb{X}_t = \{C_t, L_t, Y_t, \frac{C_t}{Y_t}, r_t^k, w_t\} \sim \varphi_{NN}$ depending on the initial capital endowment $K_t \in [10^{-8}, 50]$, *ceteris paribus*

Example 2: NK DSGE model with Monopolistic Competition

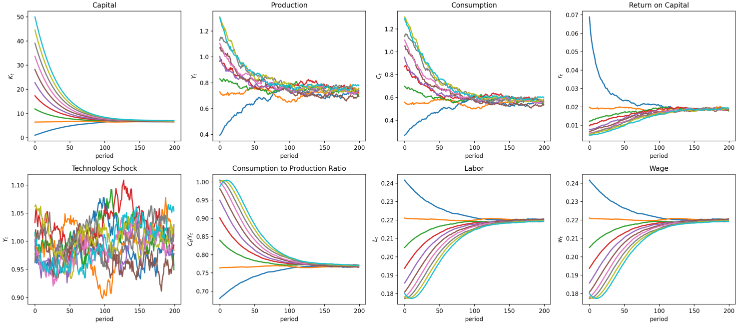
Stability and interpretation of the solution



The impulse-response functions for the state variables $\{K_t, A_t\}$ and control variables $\{L_t, C_t, Y_t, w_t, r_t^k\}$. Capital endowments are distributed uniformly $K_t \sim Uniform[10^{-8}; 50]$; the estimated parameter values of the TFP shock are fixed at $\tilde{\Theta} = \{\rho_A, \sigma_A\} = \{0.95, 0.01\}$; other model parameters are fixed at calibrated values

Example 2: NK DSGE model with Monopolistic Competition

Simulation of the Economy



Simulation of the economies in the New Keynesian DSGE model with monopolistic competition. Capital endowments $K_t \in [10^{-8}; 50]$ are distributed over the state space, economies differ in capital endowments and shock realisations. Other model parameters are fixed at calibrated values

Discussion on the Algorithm Performance in the HA Framework (1)

We attempted to solve two Krussel-Smith style HA models with aggregate risks:

- small one-sector HA model from Maliar et al. (2021)
  - households differ only in productivity and capital endowment;
- medium-size HANK model from Kase et al. (2022)
  - households differ in preferences, productivity, and employment
  - same fundamental parameters and aggregate policy variables
  - monetary and fiscal authority, intermediate and final goods production
  - occasionally binding constraints and zero lower bound

Discussion on the Algorithm Performance in the HA Framework (2)

Explanations for the results obtained

The neural network framework did not produce a stable solution:

- the shapes of the agents' approximated decision functions kept changing after 100,000 training steps regardless of the loss function and the neural network parametrisation;
- in a simulation exercise:
  - real return on investments and inflation increased tenfold;
  - aggregate consumption, labour supply, and production reduced to zero;
  - individual bond holdings converged to a constant value

Large[2] models hold considerable local extrema in the loss function. **Attempted treatments:**

- training the neural network for more iterations;
- using different initial weights of the neural network layers;
- experimenting with neural network architecture and optimisation parameters;
- adjusting the training loss function - *gives space for further research*

[2] in terms of state space dimensionality

Discussion on the Algorithm Performance in the HA Framework (3)

Explanations for the results obtained

Based on experiments we formulated the **drawbacks of our algorithm**:

- augmenting the state space with estimated model parameters $\tilde{\Theta}$ requires exponentially more training data points and augmented solution time;
- some model parameter combinations are incompatible, but the neural network still learns from these points;
- the AiO formulation of the optimisation problem admits unstable behaviour in a complex non-linear setting with many shocks

Closing Remarks

Neural network-based algorithms in their current development state generate reliable solutions for small-scale RA models but cannot produce stable results for high-dimensional models with non-linearities

Proposed treatments to improve the neural network solution convergence:

- filtering out the incompatible model parameter combinations with active learning;
- neural network loss function adjustments:
  - more rigid approximations of the market equilibrium conditions;
  - selecting appropriate weights for the equilibrium conditions in the loss function;
  - replacing Euler equation method with Bellman;
- exploration of the All-in-One expectation parameter stability

Faculty of Economic Sciences

November 8, 2024

Solving and Estimating Non-Linear HANK Models With Machine Learning

Project Proposal

Tatiana Sorokina, BST193
tssorokina@edu.hse.ru

Referernces (1)

- Elfwing, S., Uchibe, E., & Doya, K. (2017, November 1). Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. Retrieved April 29, 2023, from http://arxiv.org/abs/1702.03118
- Kase, H., Melosi, L., & Rottner, M. (2022, June 14). Estimating nonlinear heterogeneous agents models with neural networks. https://doi.org/10.2139/ssrn.4138711
- Kingma, D. P., & Ba, J. (2017, January 29). Adam: A method for stochastic optimization. https://doi.org/10.48550/arXiv.1412.6980
- Lucas, R. E. (1976). Econometric policy evaluation: A critique. *Carnegie-Rochester Conference Series on Public Policy*, *1*, 19–46. https://doi.org/10.1016/S0167-2231(76)80003-6
- Malakhovskaya, O. (2021). *"business cycles"*. https://economics.hse.ru/en/det/courses/444510956.html
- Maliar, L., Maliar, S., & Winant, P. (2021). Deep learning for solving dynamic economic models.. *Journal of Monetary Economics*, *122*, 76–101. https://doi.org/10.1016/j.jmoneco.2021.07.004

Referernces (2)

Sims, C. A. (1980). Macroeconomics and reality [Publisher: [Wiley, Econometric Society]]. *Econometrica*, *48*(1), 1–48. https://doi.org/10.2307/1912017

Appendix (1)

Consumption-saving problem

Table: Estimated parameters of the consumption-saving problem

Parameter	Parameter name	Lower bound	Upper bound	Calibrated value
β	Discounting factor	0.92	0.99	0.9
γ	Preference parameter in the Cobb-Douglas utility function	1.8	2.2	2.0
$\bar{R}$	Target interest rate	1.0	1.1	1.04
ρ_r	Persistence of the interest rate shock	0.19	0.21	0.2
σ_r	Std. of the interest rate shock	0.0005	0.0015	0.001
ρ_p	Persistence of the permanent production component shock	0.92	0.97	0.999
σ_p	Std. of the permanent production component shock	0.00005	0.00015	0.0001
ρ_q	Persistence of the transitory production component shock	0.92	0.99	0.9
σ_q	Std. of the transitory production component shock	0.0005	0.0015	0.001
ρ_b	Persistence of the preference shock	0.18	0.22	0.2
σ_b	Std. of the preference shock	0.0005	0.0015	0.001

Appendix (2)

NK DSGE model

Table: Estimated and calibrated parameters of a simple New Keynesian model with monopolistic competition

Parameter	Parameter name	Lower bound	Upper bound	Calibrated value
β	Discounting factor	-	-	0.99
γ	Normalisation constant in the utility function	-	-	0.4
α	Output elasticity of capital in the production function	-	-	0.35
δ	Depreciation rate	-	-	0.025
ζ	Elasticity of the intermediary good substitution	-	-	2.0
$\bar{A}$	Determined component of the total factor productivity shock	-	-	1.0
ρ_A	Persistence of the total factor productivity shock	0.9	0.99	0.95
σ_A	Std. of the total factor productivity shock	0.005	0.015	0.01

Appendix (3)

Parametrisation of the neural network for NK DSGE model

- The neural network parametrises decision on labour supply and the normalised Lagrange multiplier:

$$\begin{pmatrix} L_t \\ 1-\mu_t \end{pmatrix} = \varphi_{NN}\left(\underbrace{K_t}_{\mathbb{S}_t}, \underbrace{A_t}_{\mathbb{M}_t}, \underbrace{\rho_A, \sigma_A}_{\tilde{\Theta}} \Big| \underbrace{\beta, \gamma, \alpha, \delta, \zeta, \bar{A}}_{\bar{\Theta}}\right)$$

- Other control variables $\mathbb{X}_t = \left\{Y_t, C_t, K_{t+1}, w_t, r_t^k\right\}$ are derived according to equilibrium conditions
- The loss function for the neural network is comprised in line with consumption-saving problem