Estimating Nonlinear Heterogeneous Agents Models with Neural Networks

Hanno Kase
University of Minnesota

Leonardo Melosi
FED Chicago, CEPR

Matthias Rottner
Deutsche Bundesbank

July 21, 2022

Joint BoE, ECB and DAFM (King's College London) Conference: Advanced analytics: new methods and applications for macroeconomic policy

Introduction

- HANK models have gained more and more prominence
  - Role of borrowing limits, social inequality, transmission of monetary policy
- Such models are hard to handle due to their elevated complexity
  - Heterogeneous agents facing idiosyncratic risks
  - Aggregate uncertainty and nonlinearities
- Forces to study tractable approximations and limits the empirical analysis
  - Loss of interesting features such as ZLB or stochastic volatility

⇒ New approach based on machine learning to estimate complex models

- Estimation of a HANK model in its nonlinear specification

Estimation with Neural Networks

- Neural networks (NN) are the fundamental building block of our approach
  - → Neural networks can tame curse-of-dimensionality and are very scalable
- Why is it challenging to estimate complex models?
  1. It is infeasible to solve such complex models sufficiently often

  → We exploit the scalability of NN to solve the model only ONCE
    - Treat parameters as pseudo state variables and adapt the NN training

  2. It is very costly to evaluate the likelihood with a Monte Carlo filter repeatedly

  → We develop a particle filter trained neural network approach
    - Training of additional NN to provide the outcome of the filter

Proofs of Concept

1. Neural network based solution vs. analytical one for a simple model
   - Solution based on our extended neural network with pseudo state variables
   - Laboratory model is a version of the linearized 3 equation NK model

   ⇒ Extended neural network coincides with true solution

2. Neural network based estimation vs. conventional one for a nonlinear model
   - Focus on Bayesian estimation with neural networks
   - Laboratory model is a RANK model with a zero lower bound

   ⇒ The estimation results are very similar

Estimating a Non-Linear HANK Model

- Nonlinear Heterogeneous Agent New Keynesian model as laboratory
  - Idiosyncratic income risk and borrowing limit for households
  - Several aggregate shocks, backward looking components and ZLB
  - Estimation includes 12 parameters
    - No restriction on the parameters that can be considered
  - Identification of parameters related to idiosyncratic risk is weak

$\Rightarrow$ Our estimation procedure can recover the true-data generating process

$\Rightarrow$ Capturing idiosyncratic and aggregate risk simultaneously is important

  - Interactions between nonlinearities, aggregate uncertainty and heterogeneity

Literature

- Neural Networks in Macroeconomic Modeling
  - Fernandez-Villaverde et al. (2020), Chen et al. (2021), Maliar, Maliar and Winant (2021), Azinovic et al. (2022)

  $\Rightarrow$ Neural network based likelihood estimation procedure

- HANK models, Aggregate Uncertainty and Nonlinearities
  - Reiter (2009), Ahn et al. (2018), Boppart et al.(2018), Auclert et al. (2021), Winberry (2021), Gorodnichenko et al. (2021), Fernandez-Villaverde et al.

  $\Rightarrow$ Strategy that exploits neural networks to estimate HANK models

- Estimation of HANK models
  - Auclert et al. (2020,2021), Bayer et al. (2019), Lee (2021)

  $\Rightarrow$ Estimation of nonlinear HANK model with individual and aggregate risk

Challenge for Estimation and Neural Networks

- Estimation of complex nonlinear models
  - E.g., HANK models with aggregate and individual nonlinearities
- Requires the repetition of two expensive steps again and again
  1. **Solve the model for a considered parameters combination**
  2. Evaluate the fit of the model with the data (with a particle filter)
- Seems to render estimation of complex models infeasible

⇒ A neural networks approach to overcome these issues

  - Proofs of concept and estimation of a HANK model

Class of Models

- Interested in solving DSGE models in its nonlinear specification
  - State variables $\mathbb{S}_t$, shocks ν_t and structural parameters Θ
- Dynamics of model can be summarized as (nonlinear) transition equation

$$\mathbb{S}_t = f\left(\mathbb{S}_{t-1}, \nu_t; \Theta\right),$$

where function f is generally unknown and needs to be obtained numerically

- Heterogeneity: Approximate distribution with finite number agents L More

Key trick: Pseudo State Variables

- Decisive step to solve these models is to obtain policy functions $\psi(\cdot)$:

$$\psi_t = \psi(\mathbb{S}_t | \Theta),$$

- Key trick: Solve the policy function over an entire parameter range
- Divide the parameters in two subsets

$$\Theta = \{\tilde{\Theta}, \bar{\Theta}\},$$

where $\tilde{\Theta}$ is the set of parameters to be estimated and $\bar{\Theta}$ is the set of parameters to be calibrated

- Treat the (subset of) parameters as pseudo state variables

$$\psi_t = \psi(\mathbb{S}_t, \tilde{\Theta} | \bar{\Theta}),$$

$\Rightarrow$ Policy functions depend now on the state variables and the parameters

Example: Linearized NK model

- Small off-the-shelf linearized three equation NK model with TFP shock
  - Features an analytical solution

$$\hat{X} = E_t\hat{X}_{t+1} - \sigma^{-1}\left(\phi_\Pi\hat{\Pi}_t + \phi_Y\hat{X}_t - E_t\hat{\Pi}_{t+1} - \hat{R}_t^F\right)$$
$$\hat{\Pi}_t = \kappa\hat{X}_t + \beta E_t\hat{\Pi}_{t+1}$$
$$\hat{R}_t^F = \rho_A\hat{R}_{t-1}^F + \sigma(\rho_A - 1)\omega\sigma_A\epsilon_t^A$$

where $\hat{X}_t$ is the output gap, $\hat{\Pi}$ is inflation, R_t^F is the risk free rate and ϵ_t^A is a TFP shock

Example: Solution to Linearized NK Model

- Solution to equation system depends on state variables and parameters

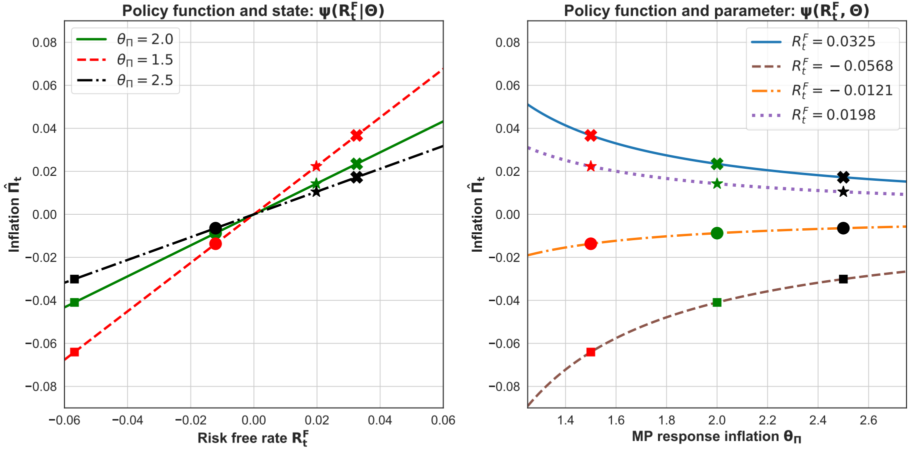
$$\begin{pmatrix} \hat{X}_t \\ \hat{\Pi}_t \end{pmatrix} = \psi \left(\underbrace{\hat{R}_t^F}_{\text{State } S_t}, \underbrace{\beta, \sigma, \eta, \phi, \theta_\Pi, \theta_Y, \rho_A, \sigma_A}_{\text{Parameters } \tilde{\Theta}} \right).$$

- The analytical solution is given as (method of undetermined coefficients)

$$\hat{X}_t = \frac{1 - \beta\rho_A}{(\sigma(1-\rho_A) + \theta_Y)(1-\beta\rho_A) + \kappa(\theta_\Pi - \rho_A)} \hat{R}_t^F,$$
$$\hat{\Pi}_t = \frac{\kappa}{(\sigma(1-\rho_A) + \theta_Y)(1-\beta\rho_A) + \kappa(\theta_\Pi - \rho_A)} \hat{R}_t^F.$$

Graphical Characterization: Policy Function

- Policy function over the parameter space $\theta_\Pi \in [1.25, 2.75]$



Challenge

- But, how can we solve such policy functions for models that feature jointly
  1. Pseudo state variables for estimation $\tilde{\Theta}$
  2. Nonlinear dynamics (e.g. zero lower bound, borrowing limits)
  3. Heterogeneous agents
- Most numerical techniques are not well suited due to curse of dimensionality

$\Rightarrow$ NN tame the curse of dimensionality and are universal approximators More

- Can handle high-dimensional input (e.g. many parameters, shocks, agents)
- Can resolve local features accurately (e.g. nonlinear features)
- Can capture irregularly shaped domain

Remarkable Features of Neural Networks

1. Universal approximation theorem (Hornik et al. 1989, Cybenko 1989)
   - Sufficient wide NN can approximate any finite-dimensional function with any desired non zero error

   $\Rightarrow$ NN can be used to solve macroeconomic models

2. Scalability and curse of dimensionality (Barron, 1993, Bach, 2017)
   - NN handle high dimensional problems much better than classical function approximators

   $\Rightarrow$ Scalability allows to handle models with a large number of states

$+$ Extraordinary efficiency of modern machine learning software and hardware

Extended Neural Network-Based Solution Method

- We use NN to solve the extended policy functions $\psi_{NN}(\mathbb{S}_t, \tilde{\Theta}|\bar{\Theta})$
  - Minimization of the Euler residual (loss function)
  - Training over thousands of iteration and number of economies (batch size B)
  - Training over parameter space and stochastic solution domain
- Adjust NN training to solve policy function over entire parameter space

$$\tilde{\Theta} = \left\{ \left[\tilde{\Theta}^{\underline{1}}, \tilde{\Theta}^{\overline{1}}\right], \left[\tilde{\Theta}^{\underline{2}}, \tilde{\Theta}^{\overline{2}}\right], \ldots, \left[\tilde{\Theta}^{\underline{P}}, \tilde{\Theta}^{\overline{P}}\right] \right\}$$

  - We draw new parameters for each economy in each iteration
  - Combined with a simulation step to adjust solution domain for each draw

⇒ Extended neural network provides solution over entire parameter space

Example: Linearized NK Model

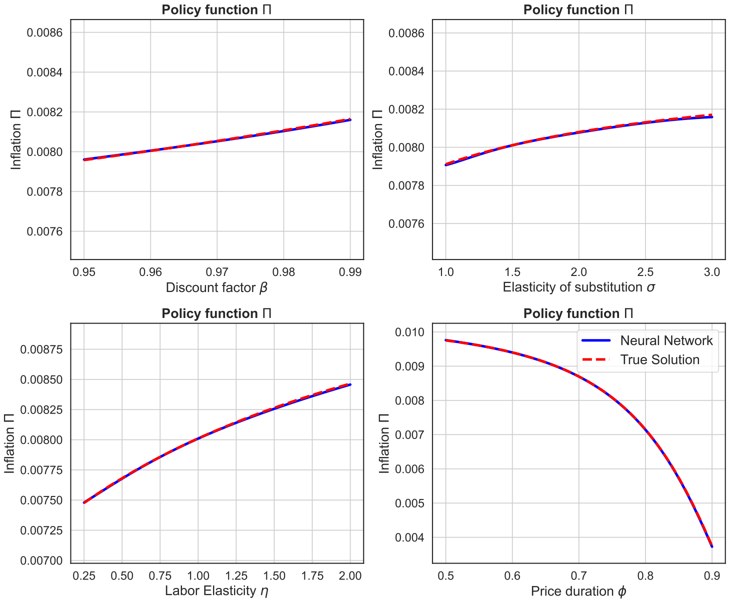
- We are interested in finding $\begin{pmatrix} \hat{X}_t \\ \hat{\Pi}_t \end{pmatrix} = \psi_{NN}\left(\hat{R}^F_t, \beta, \sigma, \eta, \phi, \theta_\Pi, \theta_Y, \rho_A, \sigma_A\right)$
- Minimization of the squared residual error (loss function)

$$
\begin{aligned}
err_1 =& \hat{X} - \left(E_t \hat{X}_{t+1} - \sigma^{-1}\left(\phi_\Pi \hat{\Pi}_t + \phi_Y \hat{X}_t - E_t \hat{\Pi}_{t+1} - \hat{R}^F_t\right)\right) \\
err_2 =& \hat{\Pi}_t - \left(\kappa \hat{X}_t + \beta E_t \hat{\Pi}_{t+1}\right)
\end{aligned}
$$

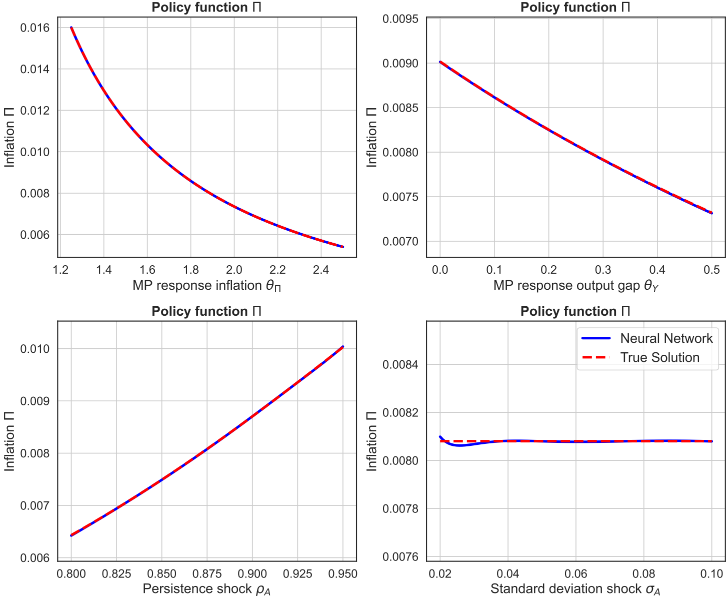
- Training NN over 100000 iterations and batch size of 500 economies
- Stochastic solution from simulating $\hat{R}^F_t = \rho_A \hat{R}^F_{t-1} + \sigma(\rho_A - 1)\omega\sigma_A\epsilon^A_t$
- We train the extended NN by drawing from the bounded parameter space

Parameters		LB	UB	Parameters		LB	UB
β	Discount factor	0.95	0.99	θ_Π	MP inflation response	1.25	2.5
σ	Relative risk aver.	1	3	θ_Y	MP output response	0.0	0.5
η	Inverse Frisch elas.	1	4	ρ_A	Persistence TFP shock	0.8	0.95
φ	Price duration	0.5	0.9	σ_A	Std. dev. TFP shock	0.02	0.1

Neural Network: Inflation over the Parameter Space



NN: Inflation over the Parameter Space (cont'd)



Challenge for Estimation and Neural Networks

- Estimation of complex nonlinear models
  - E.g., HANK models with aggregate and individual nonlinearities
- Requires the repetition of two expensive steps again and again
  1. ~~**Solve the model for a considered parameters combination**~~
  2. **Evaluate the fit of the model with the data (with a particle filter)**
- Seems to render estimation of complex models infeasible

⇒ A neural networks approach to overcome these issues
  - Proofs of concept and estimation of a HANK model

Neural Network Particle Filter

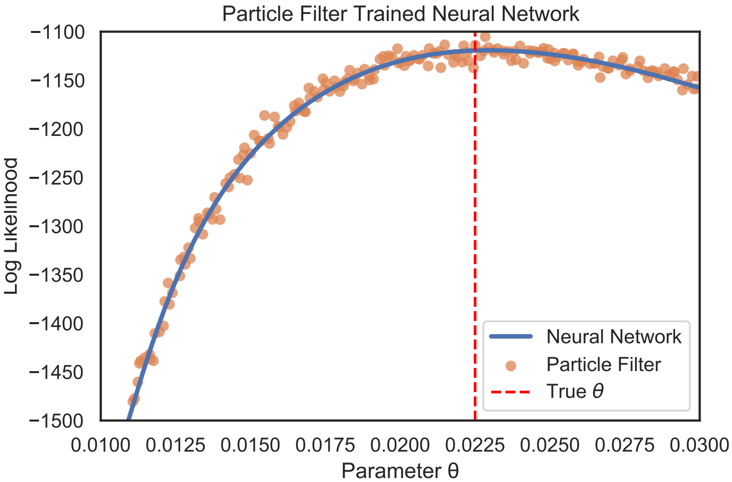
- We need to evaluate the fit of the model with the data More
  - Particle filter calculates the likelihood of the nonlinear model
  - Calculation is noisy and can be time consuming $\Rightarrow$ Bottleneck
- Goal: Additional neural network that gives directly output of particle filter
  - Create a dataset of parameter values and corresponding likelihoods
    - Run the particle filter for randomly drawn values from the parameter space
  - Train an additional neural network with this dataset

$\Rightarrow$ Particle filter trained neural network approach

- Surrogate model that provides mapping from parameters to likelihood

Graphical Characterization: NN based Particle Filter

- Particle filter trained neural network
  - Use particle filter to create data, which then can used to train neural network



Challenge for Estimation and Neural Networks

- Estimation of complex nonlinear models
  - E.g., HANK models with aggregate and individual nonlinearities
- Requires the repetition of two expensive steps again and again
  1. ~~**Solve the model for a considered parameters combination**~~
  2. ~~**Evaluate the fit of the model with the data (with a particle filter)**~~
- Seems to render estimation of complex models infeasible

⇒ A neural networks approach to overcome these issues

  - **Proofs of concept and estimation of a HANK model**

Comparison to Conventional Estimation

- Estimation of a nonlinear model with neural networks
  - RANK model with zero lower bound in its fully nonlinear specification
  - True data-generating process to provide controlled environment
- Neural network based Bayesian estimation
  - Extended neural network, surrogate particle filter, RWMH algorithm
- Conventional methods follow Herbst and Schorfheide (2015)
  - Solve model with global methods, particle filter, RWMH algorithm

⇒ Estimation results are very similar and recover true data-generating process

More

Estimation of Nonlinear HANK with Neural Networks

- HANK with individual and aggregate nonlinearities
  - Households face idiosyncratic income risk s_t^i and a borrowing limit $\underline{B}$

$$E_0 \sum_{t=0}^{\infty} \beta^t \exp(\zeta_t^D) \left[\left(\frac{1}{1-\sigma} \right) (C_t^i - hC_{t-1})^{1-\sigma} - \chi \left(\frac{1}{1+\eta} \right) (H_t^i)^{1+\eta} \right]$$

$$\text{s.t. } C_t^i + B_t^i = W_t s_t^i H_t^i + \frac{R_{t-1}}{\Pi_t} B_{t-1}^i - T_t^i + Div_t^i$$

$$B_t \geq \underline{B}$$

  where idiosyncratic risk follows an AR(1) process: $s_t^i = \rho_s s_{t-1}^i + \sigma_s \epsilon_t^i$

  - Aggregate shocks: preference ζ^D, growth rate g_t and monetary policy mp_t
  - Consumption habit h and persistence in the monetary policy rule ρ_R
  - Monetary policy is constrained by the zero lower bound

$$R_t = \max \left[1, \left(R_{t-1}^N \right)^{\rho_R} \left(R \left(\frac{\Pi_t}{\Pi} \right)^{\theta_\Pi} \left(\frac{Y_t}{Z_t Y} \right)^{\theta_Y} \right)^{1-\rho_R} \exp(mp_t) \right]$$

Setup and Training of Neural Networks

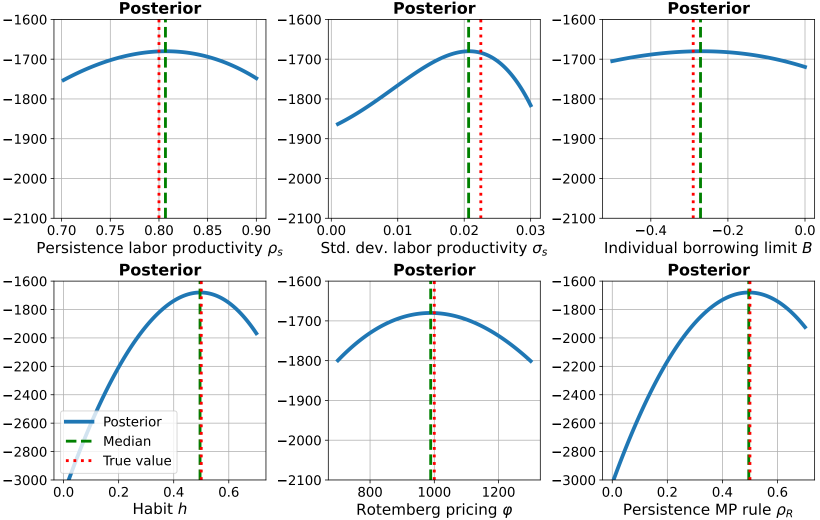
- We are interested in finding the policy functions over the parameter range
  - Aggregate PFs: Inflation and wage
  - Individual PFs: Labor choice and multiplier on borrowing constraint
- Model features 217 state variables
  - 200 individual, 5 aggregate and 12 pseudo (parameters) states
  - Approximation of continuum with 100 agents
- Training NN over 200000 iterations and batch size of 100 economies
  - Minimization of the squared residual error of 205 equations
- Estimation, likelihood and particle filter
  - Observation equation connects output growth, inflation and interest rate
  - NN based particle filter trained with 15000 likelihood points
  - Metropolis Hastings algorithm with 500000 draws

$\Rightarrow$ Bayesian estimation of HANK model in its nonlinear specification

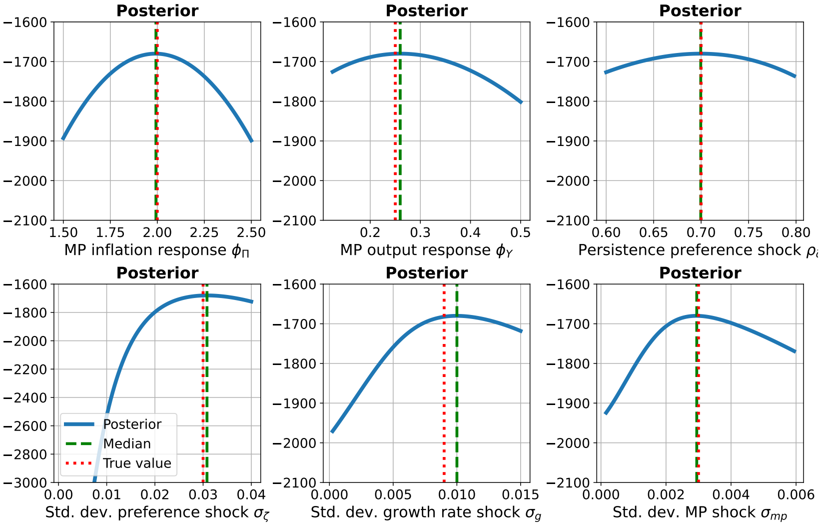
Estimation: Results and Priors

Estimation								
Par.	Prior					Neural Network		
	Type	Mean	Std	Lower Bound	Upper Bound	Posterior Median	5%	95%
Parameters related to idiosyncratic risk								
$\underline{B}$	Trc.N	-0.29	0.05	-0.5	0.0	-0.27	-0.36	-0.18
ρ_s	Trc.N	0.8	0.01	0.7	0.9	0.81	0.79	0.82
σ_s	Trc.N	2.25%	0.5%	0.01%	3.0%	2.07	1.87	2.26
Parameters related to aggregate risk								
h	Trc.N	0.5	0.01	0.0	0.7	0.50	0.48	0.51
φ	Trc.N	1000	25	700	1300	989	949	1028
ρ_r	Trc.N	0.5	0.01	0.0	0.7	0.50	0.48	0.51
θ_Π	Trc.N	2.0	0.025	1.5	2.5	2.00	1.95	2.03
θ_Y	Trc.N	0.25	0.025	0.125	0.5	0.26	0.23	0.29
ρ_ζ	Trc.N	0.7	0.025	0.6	0.8	0.70	0.68	0.72
σ_ζ	Trc.N	3.0%	0.25%	0.1%	4.0%	3.08%	2.92%	3.25%
σ_g	Trc.N	0.9%	0.1%	0.01%	1.5%	1.00%	0.92%	1.09%
σ_{mp}	Trc.N	0.3%	0.1%	0.01%	0.6%	0.29%	0.27%	0.32%

Posterior: Estimated Parameters and Idiosyncratic Risk



Posterior: Estimated Parameters and Aggregate Risk

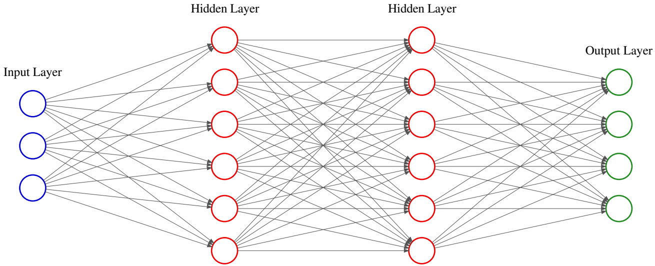


Conclusion

- Novel neural network based Bayesian estimation procedure
  - Extended neural network with pseudo state variables
  - Neural network based particle filter algorithm
  - Estimation of models with hundreds of state variables possible
- Estimation of a HANK with individual and aggregate nonlinearities
  - Two proofs of concept based on simpler models
- Techniques open up new exciting avenues for future research questions
  - Framework to think about monetary policy strategy and inequality

Primer on Neural Networks

- Deep learning uses deep neural networks (NN) as fundamental building block
- NN are mathematical function that maps some inputs into outputs
  - Composed of several layers with neutrons



- NN is trained with batch of data points to minimize a defined loss function

Back

Incorporation of Heterogeneity

- Heterogeneity usually assumes the existence of a continuum of agents
  - → Distribution of states and shocks is infinite

$$\int \mathbb{S}_t^i d\Omega \quad \text{and} \quad \int \nu_t^i d\Omega,$$

- We approximate the distribution with a finite number agents L

$$\left\{\mathbb{S}_t^i\right\}_{i=1}^L \quad \text{and} \quad \left\{\nu_t^i\right\}_{i=1}^L.$$

- The state variables and shock can be summarized as:

$$\mathbb{S}_t = \left\{\left\{\mathbb{S}_t^i\right\}_{i=1}^L, \mathbb{S}_t^A\right\} \quad \text{and} \quad \nu_t = \left\{\left\{\nu_t^i\right\}_{i=1}^L, \nu_t^A\right\},$$

- Individual and aggregate policy functions we adjust the policy functions

$$\psi_t^i = \psi^I(\mathbb{S}_t^i, \mathbb{S}_t|\bar{\Theta}) \quad \text{and} \quad \psi_t^A = \psi^A(\mathbb{S}_t|\bar{\Theta}).$$

Back

Particle Filter

- Observation equation connects the state variables with the observables $\mathbb{Y}_t$:

$$\mathbb{Y}_t = g(\mathbb{S}_t|\tilde{\Theta}) + u_t,$$

  where g is a function and u_t is a measurement error

- Particle filter determines the likelihood

$$\mathcal{L}\left(\mathbb{Y}_{1:T};\tilde{\Theta}\right) = \Omega^{PF}\left(\mathbb{Y}_{1:T};\tilde{\Theta}\right)$$

- Particle filter can be noisy and very time consuming for complex models
- Using a filter to calculate the likelihood is still a bottleneck Back

Nonlinear RANK Model with ZLB

- Off-the-shelf New Keynesian model
  - Shocks to households' preference to consumption
  - Price rigidities a la Rotemberg
  - Zero lower bound constraint on the nominal interest rate

$$R_t = \max\left[1, R\left(\frac{\Pi_t}{\Pi}\right)^{\theta_\Pi}\left(\frac{Y_t}{Z_t Y}\right)^{\theta_Y}\right]$$

- We are interested in solving and estimating it in its nonlinear specification

Back

Neural Network and Estimation

- Training NN over 100000 iterations and batch size of 200 economies
- We train the extended NN by drawing from the bounded parameter space
- Stochastic solution from simulating the model after each draw

- Observation equation with a sample size of 1000 periods

$$\begin{bmatrix} \text{Output Growth} \\ \text{Inflation} \\ \text{Interest Rate} \end{bmatrix} = \begin{bmatrix} 100 \ln \left(\frac{Y_t}{Y_{t-1}} \right) \\ 400 \ln \left(\Pi_t \right) \\ 400 \ln \left(R_t \right) \end{bmatrix} + u_t$$

- Estimation includes five structural parameters
- Priors are truncated normal densities
- 15000 data points to train neural network based particle filter

Estimation Results

Estimation							
Par.	Cal.	Neural Network			Conventional Approach		
	True Value	Posterior Median	5%	95%	Posterior Median	5%	95%
θ_Π	2.0	2.02	1.87	2.17	2.06	1.94	2.20
θ_Y	0.25	0.251	0.238	0.263	0.248	0.237	0.258
φ	1000	988.6	935.1	1036.7	973.7	911.2	1037.2
ρ_ζ	0.8	0.686	0.669	0.701	0.691	0.670	0.710
σ^ζ	0.02	0.020	0.020	0.021	0.020	0.019	0.020

- Neural network based estimation works very well
  - Posterior median is very close to the true value
- The bounds of neural network and conventional method are very similar
- Neural network method is much faster and much more scalable!

Bayesian Estimation with NN: Posterior