DISCUSSION PAPER SERIES

DP15346

DEEP LEARNING CLASSIFICATION: MODELING DISCRETE LABOR CHOICE

Lilia Maliar and Serguei Maliar

MONETARY ECONOMICS AND FLUCTUATIONS

ISSN 0265-8003

DEEP LEARNING CLASSIFICATION: MODELING DISCRETE LABOR CHOICE

Lilia Maliar and Serguei Maliar

Discussion Paper DP15346
Published 07 October 2020
Submitted 06 October 2020

Centre for Economic Policy Research
33 Great Sutton Street, London EC1V 0DX, UK
Tel: +44 (0)20 7183 8801
www.cepr.org

DEEP LEARNING CLASSIFICATION: MODELING DISCRETE LABOR CHOICE

Abstract

We introduce a deep learning classification (DLC) method for analyzing equilibrium in discrete-continuous choice dynamic models. As an illustration, we apply the DLC method to solve a version of Krusell and Smith's (1998) heterogeneous-agent model with incomplete markets, borrowing constraint and indivisible labor choice. The novel feature of our analysis is that we construct discontinuous decision functions that tell us when the agent switches from one employment state to another, conditional on the economy's state. We use deep learning not only to characterize the discrete indivisible choice but also to perform model reduction and to deal with multicollinearity. Our TensorFlow-based implementation of DLC is tractable in models with thousands of state variables.

JEL Classification: N/A

Keywords: deep learning, neural network, logistic regression, classification, discrete choice, Indivisible labor, intensive and extensive margins

Lilia Maliar - lmaliar@gc.cuny.edu
The Graduate Center, CUNY, CEPR and CEPR

Serguei Maliar - smaliar@scu.edu
Santa Clara University

Acknowledgements
Lilia Maliar and Serguei Maliar acknowledge the support from NSF grants SES-1949413 and SES-1949430, respectively. We thank Marc Maliar for his help with writing a TensorFlow code for producing numerical solutions.

Deep Learning Classification: Modeling Discrete Labor Choice*

Lilia Maliar† Serguei Maliar‡

October 6, 2020

Abstract

We introduce a deep learning classification (DLC) method for analyzing equilibrium in discrete-continuos choice dynamic models. As an illustration, we apply the DLC method to solve a version of Krusell and Smith's (1998) heterogeneous-agent model with incomplete markets, borrowing constraint and indivisible labor choice. The novel feature of our analysis is that we construct discontinuous decision functions that tell us when the agent switches from one employment state to another, conditional on the economy's state. We use deep learning not only to characterize the discrete indivisible choice but also to perform model reduction and to deal with multicollinearity. Our TensorFlow-based implementation of DLC is tractable in models with thousands of state variables.

Key Words : artificial intelligence, machine learning, deep learning, neural network, logistic regression, softmax regression, classification, image recognition, discrete choice, indivisible labor, intensive and extensive margins

*Lilia Maliar and Serguei Maliar acknowledge the support from NSF grants SES-1949413 and SES-1949430, respectively. We thank Marc Maliar for his help with writing a TensorFlow code for producing numerical solutions.

†The Graduate Center, City University of New York, USA and CEPR.

‡Santa Clara University

1 Introduction

Macroeconomic models are generally built on the assumption of continuous-set choices. For example, an agent can distribute wealth in any proportion between consumption and savings or she can distribute time endowment in any proportion between work and leisure. But certain economic choices are discrete: the agent can either buy a house or not, be either employed or not, retire or not, etc. The previous literature emphasizes the importance of discrete nature of choices for explaining certain economic phenomena, including indivisibility of labor (Hansen (1985), Rogerson (1988), Chang and Kim (2007), Prescott, Rogerson and Wallentius (2009), Chang, Kim, Kwon and Rogerson (2019)), retirement (Iskhakov, Jørgensen, Rust and Schjerning (2017)), sovereign default (Arellano (2008), Chatterjee, Corbae, Nakajima and Ríos-Rull (2007)). However, the progress in modeling discrete choices is still limited, in part, because they are more challenging to analyze numerically than continuous-set choices.

In the present paper, we introduce a deep learning classification (DLC) method that can be used to solve dynamic economic models with continuous-discrete choices. Our analysis relies on the same techniques that led to ground breaking application in data science (e.g., image and speech recognition). As an illustration, let us consider an image-recognition problem in which a machine classifies images into, let's say, cats, dogs and sheep. We parameterize the probabilities of the three classes with a deep (multilayer) neural network to which we apply a softmax function – a generalization of a logistic (sigmoid) function for the multiclass problems. We feed into the machine a collection of images; and we train the machine to maximize the likelihood function (equivalently, to minimize the cross-entropy loss) to ensure a correct classification; see Goodfellow, Bengio and Courville (2016) for a survey of classification techniques in data science.

Let us now show how the same idea can be used to study equilibrium in a heterogeneous-agent model with indivisible labor choice. An agent wants to be employed if her wage is higher than a certain reservation wage but wages are not known until all agents fix their employment choices (because wages depend on aggregate labor). To construct a solution to this fixed point problem, we assume that an agent assesses a probability that her wage will be higher than her reservation wage and chooses to be employed if this probability is larger than 1/2. We parameterize the state-contingent probability function with a deep neural network. Once the probability functions are fixed for all agents, we can compute wages and "validate" the agents' decisions, namely, we can check if the agents would have made the same labor choices if wages were known at the moment of their choices. We then train the machine to maximize the likelihood function that matches ex-ante and ex-post labor choices for all heterogenous agents.

The proposed DLC method can be generalized to include any finite number of discrete choices (not just two). As an illustration, we extend our economy to include three employment states, namely, unemployment, part-time employment and full-time employment. In that case, we parameterize the three probability functions corresponding to unemployed, part-time employed and full-time employed by deep neural networks, and we train the machine to predict these states by maximizing the likelihood function for the softmax regression instead of the logistic regression.

As an illustration of the DLC method, we solve a version of Krusell and Smith's (1998) model in which the agents face indivisible labor choices. In fact, the studied model is computationally challenging even in the absence of discrete choice. It features heterogeneous agents, incomplete markets and borrowing constraints. The state space includes thousands of state variables of heterogenous agents and is prohibitively large. To make the model tractable, Krusell and Smith (1998) approximated the state space of each agent by her own state variables and one or few aggregate moments of the wealth distribution – this reduces the state space to 5-6 state variables; see Den Haan (2010) for a review of other methods for reducing the state space. A distinctive feature of our DLC method is that it does not rely on moments or other reduced representations of the state space but works with the actual state space consisting of all individual and aggregate state variables – we let the deep neural network to choose how to condense large sets of the state variables into much smaller sets of features. Our TensorFlow code is tractable in models with thousands of state variables.

Our model builds on important contributions of Chang and Kim (2007) and Chang et al. (2019). The former paper extends Krusell and Smith's (1998) analysis to include indivisible labor choice by constructing value functions of employed and unemployed agents. This approach reduces the discrete choice problem to the analysis of two continuous-choice value functions. The latter paper offers a more simple and tractable way of modeling indivisibile-labor choice by discretizing the first-order conditions of the associated divisible labor model, namely, the agent decides on how many hours to work but if the chosen hours fall below a certain level, the agent becomes unemployed. In turn, this approach reduces the model with discrete choices to a familiar setup with continuous labor choices and occasionally binding constraints. The main novelty of our analysis is that we approximate discrete choices of agents by using classification techniques instead of relying on continuous choice representations. That is, we construct decision functions that tell us when the agent switches from one discrete choice to another, conditional on the economy's state.

We compare the predictions of the heterogeneous-agent models with indivisible labor with those of the corresponding divisible labor model, as well as the representative-agent model. We find that the introduction of indivisible labor helps us correct some shortcomings of the divisible labor model, in particular, on labor markets. One improvement we observe is that the volatility of labor increases relatively to the output; in that respect, we are similar to the indivisible labor framework of Rogerson (1994) and Hansen (1993). Another improvement is a reduction in the correlation between labor and wages which is excessively high in the representative-agent model with divisible labor; this implication is an outcome of both the assumptions of indivisible labor and heterogeneous agents. As for distributional implications, the predictions of our heterogeneous-agent model with indivisible labor are similar to those of the models studied by Chang and Kim (2007) and Chang et al. (2019). First, the assumption of indivisible labor increases the degrees of inequality, helping to bring the model closer to the data. Furthermore, unlike in the divisible labor model, the degrees of income and wealth inequalities in the indivisible labor economics are less sensitive to variations in the coefficient of risk aversion. Overall, we conclude that the assumption of indivisible labor alone is not sufficient to produce empirically relevant degrees of income and wealth inequality in the model.

Our DLC method is related to recent papers on deep learning, including Duarte (2018), Villa and Valaitis (2019), Fernández-Villaverde, Hurtado, and Nuño (2019), Azinović, Luca and Scheidegger (2019), Lepetyuk, Maliar and Maliar (2020) and especially, Maliar, Maliar and Winant (2018, 2019). However, this literature does not analyze models with discrete choices, which is the main subject of the present paper. From the other side, there are numerous methods in econometrics for estimating discrete-choice models but these methods are limited to statistical applications; see Train (2009) for a review. One method that is designed to deal with discrete choices in dynamic environment is an endogenous grid method with taste shocks by Iskhakov et al. (2017); see also Iskhakov and Keane (2020) for an application of this method for estimating a partial equilibrium model with discrete labour supply on Australian data. In the context of Carroll's (2005) analysis, these papers suggest to apply logistic smoothing to the kinks by transferring the problem into the choice probability space via the taste shocks. The main conceptual difference of our analysis from those papers is that we do not attempt to smooth the kinks but try to accurately approximate such kinks by using the-state-of-the-art deep learning classification method.

The rest of the paper is as follows: In Section 2, we set up the Krusell and Smith (1998) model with divisible labor choice; in Section 3, we solve the model with indivisible labor choice; in Section 4, we analyze the model with full- and part-time employment; in Section 5, we compare the aggregate and distributional predictions of divisible and indivisible labor models; and finally, in Section 6, we conclude.

2 Krusell-Smith model with divisible labor choice

We start by considering a version of the Krusell-Smith (1998) model in which labor choice is divisible. Such a model will be useful as a basis for constructing the indivisible labor model.

Consumer side. The economy consists of a set of heterogeneous agents $i = 1, ..., \ell$ that are identical in fundamentals, but differ in dimensions of productivity and capital holdings. The agents experience idiosyncratic productivity shocks and the economy experiences aggregate shock. Each agent i solve

$$\max_{\left\{c_t^i,k_{t+1}^i,n_t^i\right\}_{t=0}^{\infty}} E_0\left[\sum_{t=0}^{\infty}\beta^t u\left(c_t^i,1-n_t^i\right)\right] \tag{1}$$

$$\text{s.t. } c_t^i+k_{t+1}^i=R_t k_t^i+W_t z_t^i n_t^i, \tag{2}$$

$$\ln z_{t+1}^i=\rho_z \ln z_t^i+\sigma_z\epsilon_t^i \text{ with } \epsilon_t^i\sim\mathcal{N}(0,1), \tag{3}$$

$$k_{t+1}^i\geq\overline{k}, \tag{4}$$

where c_t^i, n_t^i, k_t^i and z_t^i are consumption, leisure, labor, capital and agent's productivity; $\beta\in(0,1)$; $\rho_z\in(-1,1)$ and $\sigma_z\geq 0$; and initial condition $\left(k_0^i,z_0^i\right)$ is given. The capital choice is restricted by a borrowing limit $\overline{k}\leq 0$, and the total time endowment is normalized to one so that the term $1-n_t^i$ represents leisure.

Production side. The production side of the economy is described by a Cobb-Douglas production function $\exp(z_t)\,k_t^{\alpha-1}h_t^{1-\alpha}$, where $k_t=\sum_{i=1}^{\ell}k_t^i$ is aggregate capital, $h_t=\sum_{i=1}^{\ell}z_t^i n_t^i$ is aggregate efficiency labor, and z_t is an aggregate productivity shock following

$$\ln z_{t+1}=\rho\ln z_t+\sigma\epsilon_t \text{ with } \epsilon_t\sim\mathcal{N}(0,1), \tag{5}$$

where $\rho\in(-1,1)$ and $\sigma\geq 0$. The interest rate R_t and wage W_t are given by

$$R_t=1-d+\exp(z_t)\,\alpha k_t^{\alpha-1}h_t^{1-\alpha} \text{ and } W_t=\exp(z_t)(1-\alpha)\,k_t^{\alpha}h_t^{-\alpha}, \tag{6}$$

where $d\in(0,1]$.

Intertemporal choice. The Kuhn-Tucker condition of the agent's problem (1)–(4) with respect to capital is

$$\mu_t^i\delta_t^i=0, \tag{7}$$

where $\mu_t^i\geq 0$ is the Lagrange multiplier associated with the borrowing constraint (4) and $\delta_t^i\equiv\ k_{t+1}^i-\overline{k}\geq 0$ is the distance to the borrowing limit, satisfying the Euler equation,

$$\mu_t^i\equiv\ u_1\left(c_t^i,n_t^i\right)-\beta E_t\left[u_1\left(c_{t+1}^i,n_{t+1}^i\right)R_t\right], \tag{8}$$

where u_1 denotes a first-order partial derivative of function u with respect to the first argument. Whenever $\delta_t^i\geq 0$, the agent is not at the borrowing limit $k_t^i>\overline{k}$, so the Euler equation must hold with equality leading to $\mu_t^i=0$, and whenever the Euler equation does not hold with equality, it must be that the agent is at the borrowing constraint $\delta_t^i=0$.

Intratemporal choice. We assume that the utility function in (1) takes the form

$$u(c,n)=\frac{c^{1-\gamma}-1}{1-\gamma}+B\frac{(1-n)^{1-\eta}-1}{1-\eta}, \tag{9}$$

where γ, $\eta\geq 0$. The labor choice is characterized by a FOC of (1)–(4) with respect to labor, which under the utility function (9) is

$$n_t^i=1-\left[\frac{c_i^{-\gamma}W_t\exp\left(z_t^i\right)}{B}\right]^{-1/\eta}, \tag{10}$$

where the labor choice is perfectly divisible, i.e., the agent can choose any $n_t^i\in[0,1]$.

Deep learning solution procedure. The state space includes the state variables of all agents, as well as aggregate productivity, $(\{k_t^i, z_t^i\}_{i=1}^{\ell}, z_t)$ which is $2\ell+1$ state variables in total; for example, with $\ell = 1,000$ heterogeneous agents, the state space has $2,001$ state variables. To deal with so large number of state variables, we rely on a combination of techniques introduced in Maliar et al. (2018, 2019), including i) stochastic simulation that allows us to restrict attention to the ergodic set in which the solution "lives"; ii) multilayer neural networks that perform model reduction and help deal with multicollinearity; iii) a (batch) stochastic gradient descent method that reduces the number of function evaluations by operating on random grids; iv) a Fischer-Burmeister function that effectively approximates the kink; v) and most importantly, "all-in-one expectation operator" that allows us to approximate high-dimensional integrals with just 2 random draws on each iteration. The key contribution of the present paper is to show how the above techniques can be adapted to the analysis of models with discrete-continuous dynamic choices.

We solve for two decision functions in terms of state variables, namely, the labor choice n_t^i and the fraction of wealth $w_t^i \equiv R_t k_t^i + W_t z_t^i n_t^i$ that goes to consumption $\frac{c_t^i}{w_t^i}$,

$$\left\{n_t^i, \frac{c_t^i}{w_t^i}\right\} = \sigma\left(\zeta_0 + \varphi\left(k_t^i, z_t^i, \{k_t^i, z_t^i\}_{i=1}^{\ell}, z_t; \theta\right)\right), \tag{11}$$

where $\varphi(\cdot)$ is a multilayer neural network parameterized by a vector of coefficients θ (weights and biases), $\sigma(x) = \frac{1}{1+e^{-x}}$ is a sigmoid function which ensures that both $\frac{c_t^i}{w_t^i}$ and n_t^i are bounded to be in an interval $[0,1]$, and ζ_0 is a constant term.[1] We train the machine by using a stochastic gradient descent method until approximation (11) satisfies all model's equations (2)–(10). Since the agents are identical in fundamentals, the resulting two $2\ell+1$–dimensional decision functions are sufficient to characterize the choices of all ℓ heterogeneous agents.

Two remarkable properties of deep learning help us deal with the curse of dimensionality: First, the neural network performs the model reduction: it extracts information from thousands of state variables in the input layer and condenses it into a much smaller number of features in the hidden layer, (for example, to 64 features), and those features are used as the state variables for producing the decision variables in the output layer. Second, the deep neural network can learn to ignore collinear variables, in particular, the individual state variables k_t^i, z_t^i appear in approximation (11) both as the state variables of agent i and as a part of the distribution but such perfect multicollinearity does not create numerical problems. Further computational details are elaborated in Appendix A. In the main text, we focus on a numerical construction of labor choice, which is the main interest of the present paper.

To approximate the labor decision function, we proceed in three steps:

- **Generating an employment decision:** we obtain n_t^i from the assumed neural-network function of state variables.
- **Verifying the employment decision:** we find $h_t = \sum_{i=1}^{\ell} z_t^i n_t^i$ and W_t from (6), and we use (10) to find $\widehat{n}_t^i$.
- **Training the machine:** we train the machine with respect to θ with the aim of reducing a distance between n_t^i and $\widehat{n}_t^i$ by using a least-squares type of criterion $\min_{\theta} \sum_{i=1}^{\ell} \left(n_t^i - \widehat{n}_t^i\right)^2$.

Later in the paper, we will see how these steps can be implemented in the economy with indivisible labor.

Numerical results for the divisible labor model. For numerical analysis, we assume $\alpha = 0.36$, $d = 0.08$, $\beta = 0.96$; $\rho = 0.9$; $\sigma = 0.1$; $\rho_z = 0.9$; $\sigma_z = 0.21$; and $\overline{k} = 0$. We perform training using the

[1] In addition, we also paramterize and approximate the Lagrange multiplier μ_t^i associated with the borrowing constraint. This is needed for making stochastic gradient unbiased; see Maliar et al. (2018, 2019) for more details.

$ADAM$ stochastic gradient descent method with the batch size of 100 and the learning rate of 0.001. We fix the number of iterations (which is also a simulation length) to be $K = 100,000$. In Figure 1, we show the solution with $\ell = 300$ heterogeneous agents under $\gamma = 1$ and $\eta = 1$.

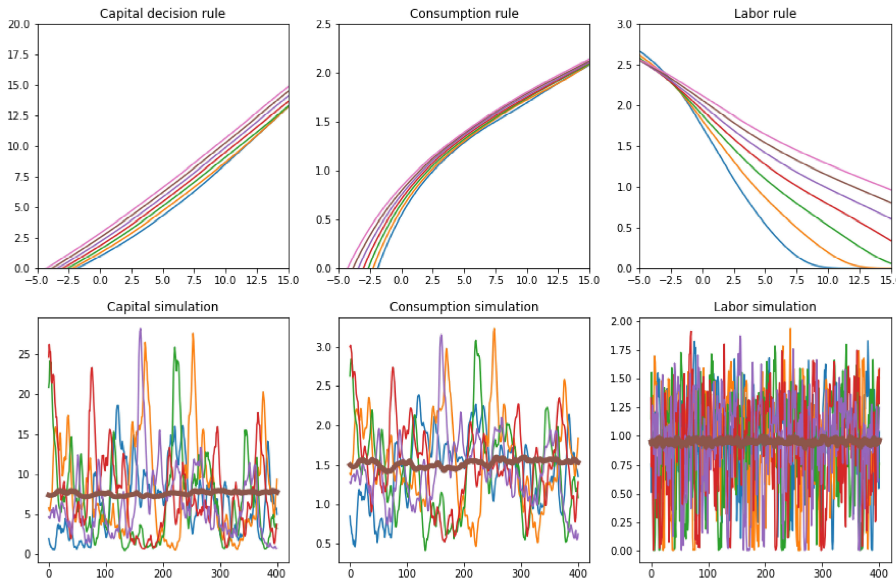

Figure 1: Divisible labor model under $\gamma = 1$ and $\eta = 1$.

In the top row, we show the individual decision rules as a function of capital. We see that next-period capital k^i_{t+1} and consumption c^i_t increase in the current capital k^i_t, while labor n^i_t is decreasing meaning that the agent chooses to enjoy more leisure $1 - n^i_t$. The seven lines in each graph correspond to seven different productivity states representing a mean $\pm$ 1, 2 and 3 standard deviations; they show that k^i_{t+1}, c^i_t and n^i_t are all increasing with productivity z^i_t. We observe a soft kink in the consumption function in the area where the agent reaches the borrowing limit.

In the bottom row, we show simulated series for 5 agents over time (we do not show all agents to avoid the clutter). As expected, fluctuations in individual capital of agents are significantly larger than the fluctuations in their consumption and labor. In the bottom row, we also show simulation for the corresponding aggregate variables k_{t+1}, c_t and n_t. The volatility of the aggregate variables (see the thick lines) is typical for the real business cycle models and is considerably lower than that of the individual variables. We will quantify the business cycle and distributive properties of the model in Section 5 after we present the version of the model with indivisible labor.

Finally, we comment on two alternative solution methods that we could have used instead and that could have simplified finding equilibrium. First, we could have approximated numerically just one labor decision function instead of both consumption and labor choices. This point was emphasized in Maliar and Maliar (2005), who argue that given n^i_t, we can find h_t, W_t and hence, c^i_t, in a closed form, while given c^i_t, we need a numerical procedure to construct h_t, W_t and n^i_t – hence, it is better to parameterize n^i_t than c^i_t in a similar model. Second, we could have solved for one individual decision function (for example, c^i_t) and one aggregate variable (for example, W_t) in terms of state variables since we can find n^i_t in a closed form, given c^i_t and W_t. We do not follow these approaches because they do not carry over to the model with indivisible labor.

3 Deep learning classification (DLC) method: logistic regression

In the divisible-labor case, the optimal labor choice must satisfy FOC (10), so the agent chooses labor by considering just the current period variables. However, the same is not true for the indivisible labor model in which the agent chooses to be employed or unemployed depending on which of the two choices leads to a higher continuation value. Chang and Kim (2007) use this approach to analyze Krusell and Smith (1998) model with indivisible labor by constructing separate value functions for employed and unemployed agents. This approach reduces the problem of approximating a discrete labor choice to that of approximating two continuous-choice value functions.

Prescott et al. (2009) proposes a different approach for modeling indivisibility of labor by considering intensive and extensive margins and by analyzing a "discretized" version of the FOC (10); in turn, Chang et al. (2019) implement this approach in the context of Krusell and Smith (1998) model. Specifically, these papers assume that the agent's labor choice is divisible and is characterized by the divisible-labor FOC (10) as long as it is above a given threshold $\overline{n}_f$ but the labor choice is zero (i.e., the agent becomes unemployed) whenever it falls below $\overline{n}_f$. Thus, the labor supply has a "kink" at the threshold level $\overline{n}_f$,

$$n_t^i = \begin{cases} 1 - \left[\frac{c_i^{-\gamma} W_t \exp\left(z_t^i\right)}{B}\right]^{-1/\eta} \geq \overline{n}_f, \\ 0 \text{ otherwise.} \end{cases} \tag{12}$$

In turn, this approach reduces the problem of constructing discrete-labor choice to the problem of approximating continuous-labor choice with occasionally binding constraints. Our main novelty is that we show how to approximate the discrete labor choice without relying on continuous choice representations. The decision functions we construct show the moment when the agent switches from one discrete choice to another, contingent on state.

To introduce our DLC method, we borrow from Prescott et al. (2009) and Chang et al. (2019) the idea of discretizing the FOCs of the divisible labor model, however, we go a step further and make the labor choice entirely indivisible by assuming that n_t^i can take just two values 0 (unemployed) and 1 (employed):

$$n_t^i = \begin{cases} 1 \text{ if } 1 - \left[\frac{c_i^{-\gamma} W_t \exp\left(z_t^i\right)}{B}\right]^{-1/\eta} \geq \overline{n}_f, \\ 0 \text{ otherwise.} \end{cases} \tag{13}$$

There are two complications we face in the model with indivisible labor choice compared to that with divisible labor choice: First, not only labor but also consumption c_i jumps when an agent switches between the employed and unemployed states (this is because in equilibrium, the employed agent accepts a consumption cut to enjoy more leisure in the unemployed state). Second, the aggregate wage W_t is unknown in the moment when the agent decides on the employment because it depends on the labor choices of all heterogeneous agents $h_t = \sum_{i=1}^{\ell} z_t^i n_t^i$ via (6) (that is, the individual labor choices must satisfy the market clearing condition (6)). We will deal with these complications by using logistic regression – a popular machine-learning technique for classification.

Generating an employment decision: Let us parameterize the employment decision boundary by a deep learning neural network with

$$x\left(s_t^i; \theta\right) = 0, \tag{14}$$

where $s_t^i \equiv \left(\left\{k_t^i, z_t^i\right\}_{i=1}^{\ell}, z_t\right)$ is the agent's state, and θ is a vector of coefficients (weights and biases) of the neural network. On the boundary $x\left(s^i; \theta\right) = 0$, the agent is indifferent between being employed and unemployed. When $x\left(s^i; \theta\right) \geq 0$ and $x\left(s^i; \theta\right) < 0$, the agent chooses to be employed $n^i = 1$ and unemployed $n^i = 0$, respectively (for the rest of the section, we omit the time subscript for expositional convenience).

We characterize the decision boundary in terms of employment probability by using logistic function. Specifically, we assume that the agent chooses to become employed with the probability p, which means that the probability of being unemployed is $1 - p$. The probabilities are related to the decision boundary (14) with a sigmoid (logistic) function

$$\ln \frac{p}{1-p} = x\left(s^i; \theta\right) \qquad \Rightarrow \qquad p = \frac{1}{1+e^{-x}} \equiv p\left(s^i; \theta\right), \tag{15}$$

The sigmoid function has the properties of cumulative density function, namely, if $x \to \infty$, then $p \to 1$; if $x \to -\infty$, then $p \to 0$; and if $x = 0$, then $p = \frac{1}{2}$. The latter threshold value corresponds to zero of decision boundary (14) and separates the employed and unemployed choices.[2] We assume that the agent chooses the employment status that corresponds to the larger probability, i.e., if $p\left(s^i;\theta\right) \geq \frac{1}{2}$, then the agent chooses $n^i = 1$; and otherwise, she chooses $n^i = 0$.

Verifying the employment decision: Once the labor choices of all agents are fixed, we compute $h_t = \sum_{i=1}^{\ell} z_t^i n^i$ and W_t from (6) and we find the employed choices $\widehat{n}_t^i$ satisfying the discretized FOC (13). Now the agents can check if the choices n^i implied by their probability function $p\left(s^i;\theta\right)$ coincide with the choices $\widehat{n}^i$ that are implied by the discretized FOC (13).

Training the machine: Generally, n^i and $\widehat{n}^i$ are not the same so we need a numerical procedure that brings them close to each other. That is, we need to find the value of θ such that if $\widehat{n}^i = 1$, then $p\left(s^i;\theta\right) \to 1$, and if $\widehat{n}^i = 0$, then $p\left(s^i;\theta\right) \to 0$. Note that this objective function can be summarized with a single expression $(p)^{\widehat{n}}(1-p)^{1-\widehat{n}}$ (because $\widehat{n} = 1$ implies $(p)^1(1-p)^0 = p$ and $\widehat{n} = 0$ implies $(p)^0(1-p)^1 = 1-p$). Using this expression, we can form a likelihood function to be maximized with respect to θ

$$\ln L(\theta) = \ln \prod_{i=1}^{\ell} \left(p\left(s^i;\theta\right)\right)^{\widehat{n}^i} \left(1 - p\left(s^i;\theta\right)\right)^{1-\widehat{n}^i} = \sum_{i=1}^{\ell} \left[\widehat{n}^i \ln\left(p\left(s^i;\theta\right)\right) + \left(1-\widehat{n}^i\right)\ln\left(1-p\left(s^i;\theta\right)\right)\right]. \quad (16)$$

The goal of training is to find a probability function $p\left(s^i;\theta\right)$ that satisfies the fixed-point property such that the values of n^i induced by $p\left(s^i;\theta\right)$ coincide with the values of $\widehat{n}^i$ implied by the discretized FOC (13) for each possible state. In effect, the procedures for solving for divisible and indivisible labor are similar except that we parameterize the probability of employment instead of the labor supply and that we maximize the likelihood function instead of the least-squares criterion.[3] Thus, the way in which we model discrete labor choice is the same as the one used for the canonical classification problem in machine learning.[4]

The solution to the model with discrete labor choice produced by the DLC method is illustrated in Figure 2. The decision rules in the top row are qualitatively similar to those in the divisible labor model, however, the changes in labor happen in discrete jumps and they induce the corresponding discrete jumps in the consumption and capital choices (as before seven lines correspond to seven individual productivity levels). The exact moment when the agent switches from the employed to unemployed states depends on the productivity level. More productive agents remain employed for much larger capital levels than low productive agents.

The simulated series for the individual capital and consumption in Figure 2 are similar to those in the divisible labor model in Figure 1, while labor fluctuates between the employed and unemployed states in discrete jumps. Finally, the fluctuations in aggregate series are again typical for the real business cycle

[2] Note that if we parameterize the probability function with a neural network directly instead of the sigmoid function, the resulting probability function can be outside the interval $[0,1]$.

[3] One can be tempted to minimize the least-squares style criteria $\sum_{i=1}^{\ell}\left(n_t^i - \widehat{n}_t^i\right)^2$ instead of maximizing the likelihood function but that would not lead to efficient training (as the cost function will have many local minima due to the assumption of a sigmoid function).

[4] For example, suppose we want to classify the tumors into malignant and benign, denoted by $n^i = 1$ and $n^i = 0$, respectively, conditional on the number of tests s^i such as the tumor size, the blood test and the age and gender of the patient. Then, we parameterize the decision boundary (14), construct the probability function (15) and maximize the likelihood function (16) with respect to θ on the set of data points $\left\{s^i\right\}$ for which it is known whether the tumor is malignant or benign (analogue of our condition (13)). After training, the machine produces the probabilities for new patients that their tumors are malignant or benign conditional on their tests. Again, a successful training implies that p is close to either zero or one and those probabilities lead to correct inferences about tumor malignancy.

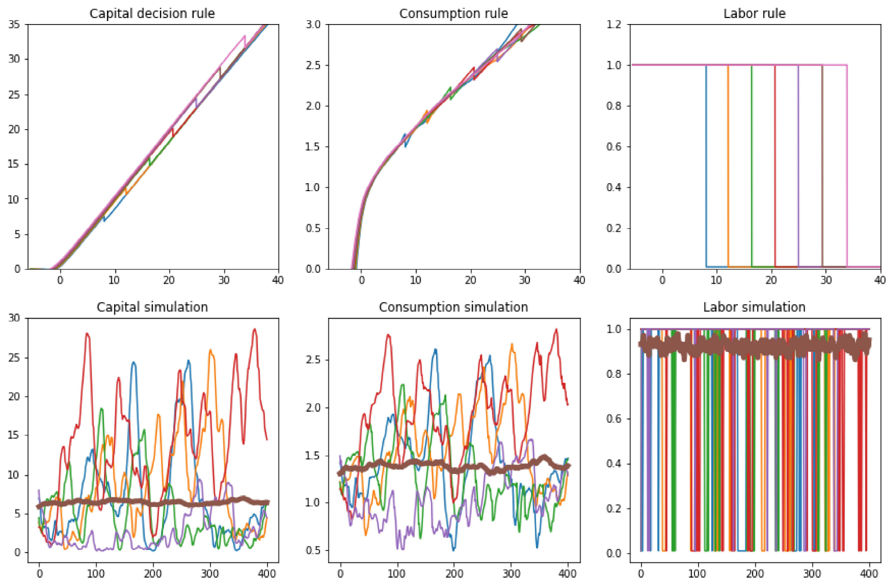

Figure 2: Indivisible labor model under $\gamma = 1$ and $\eta = 1$.

models except that we can observe that aggregate labor changes in small discrete increments which are a consequence of discrete jumps in individual labor.

Iskhakov et al. (2017) introduce another method for solving dynamic models with discrete-continuous choices, namely, an endogenous grid method with taste shocks, so it is interesting to compare our analysis to theirs. There are three main differences: First, our DLC method is designed to approximate sharp kinks in policy functions like those shown in the figure, while Iskhakov et al. (2017) suggest to smooth the kinks by introducing supplementary preference shocks which transform the discrete-choice problem into the choice probability space. Second, in our application, our discrete-choice decisions depend only on the economy's state, while in their application, such decisions depend both on state and time, namely, the agent has to decide which time period to retire. The presence of time among the argument of the choice function feature creates "secondary" discrete shocks that propagate across time domain; such secondary shocks are absent in our application. Finally, the problems that we solve have much larger dimensionality than those studied in Iskhakov et al. (2017), but their application in more challenging in another respect: they estimate the model's parameters which requires solving the model a large number of times. It would be interesting to see how our DLC method performs in the context of their application but this lies beyond the scope of the present paper.

4 DLC method for multiclass problems: softmax regression

We next extend the Krusell-Smith (1998) model to include a possibility of part-time employment, which is another case emphasized by Chang et al. (2019). However, we again differ from their analysis in that we do not assume intensive and extensive margins but consider a discrete choice between three different employment states. Specifically, the agent chooses full-time employment, $n_t^i = 1$, whenever her labor choices implied by the FOC of divisible labor model (10) is above a threshold $\overline{n}_f$; she chooses part-time employment, $n_t^i = \frac{1}{2}$, whenever it belongs to the interval $[\overline{n}_p, \overline{n}_f]$; and she chooses unemployment whenever it falls below the part-time employment threshold $\overline{n}_p$, i.e.,

$$n_t^i = \begin{cases} 1 \text{ if } 1 - \left[\frac{c_i^{-\gamma} W_t \exp\left(z_t^i\right)}{B}\right]^{-1/\eta} \geq \overline{n}_f, \\ \frac{1}{2} \text{ if } 1 - \left[\frac{c_i^{-\gamma} W_t \exp\left(z_t^i\right)}{B}\right]^{-1/\eta} \in \left[\overline{n}_p, \overline{n}_f\right], \\ 0 \text{ otherwise.} \end{cases} \tag{17}$$

To introduce the DLC method, we again assume that the agent learns the equilibrium by making employment choices with some probabilities without knowing the aggregate wage and by verifying optimality of these decisions ex-post.

Generating an employment decision: The key assumption of multiclass classification analysis is the hypothesis of an independence of irrelevance alternatives which postulates that the agent's choice between two alternatives is independent of the presence of other alternatives.[5] In our model, that means that the choice of the agent between full- and part-time employment is unaffected by the alternative of being unemployment (i.e, the choices between full-time employment and unemployment and between part-time employment and unemployment). This leads us to pairwise comparisons. There are three of them but we need to consider just two since the third one follows by a normalization of the probabilities to one. In particular, we assume

$$\ln \frac{\Pr\left(n^i = 1\right)}{\Pr\left(n^i = 0\right)} = \varphi\left(s^i, \theta_1\right) \qquad \text{and} \qquad \ln \frac{\Pr\left(n^i = \frac{1}{2}\right)}{\Pr\left(n^i = 0\right)} = \varphi\left(s^i, \theta_2\right),$$

where $\varphi\left(s^i, \theta_1\right)$ and $\varphi\left(s^i, \theta_2\right)$ are neural networks that represent decision boundaries between the choices of 1 and 0 and of $\frac{1}{2}$ and 0, respectively (we again omit the time subscript). By taking into account that the probabilities must add up to one, we obtain

$$\Pr\left(n^i = 1\right) = \frac{1}{\Delta} e^{\varphi\left(s^i, \theta_1\right)}, \qquad \Pr\left(n^i = \frac{1}{2}\right) = \frac{1}{\Delta} e^{\varphi\left(s^i, \theta_2\right)}, \qquad \Pr\left(n^i = 0\right) = \frac{1}{\Delta},$$

where $\Delta \equiv e^{\varphi\left(s^i, \theta_1\right)} + e^{\varphi\left(s^i, \theta_2\right)} + 1$ is a numerair that normalizes the probabilities to one.

The advantage of the above representation is that one can approximate three probability functions by using only two neural networks. However, for numerical computations, we will use another representation that treats all probabilities symmetrically and requires three probability functions – the so-called softmax function – to satisfy

$$\Pr\left(n^i = \overline{n}_j\right) = \frac{e^{\varphi\left(s^i, \theta_j\right)}}{e^{\varphi\left(s^i, \theta_1\right)} + e^{\varphi\left(s^i, \theta_2\right)} + e^{\varphi\left(s^i, \theta_3\right)}}, \tag{18}$$

where $\overline{n}_j \in \left\{1, \frac{1}{2}, 0\right\}$. The softmax probabilities are not uniquely defined in the sense that if we multiply the numerator and denominator by the same number, we get the same probabilities. To make employment decisions, we fix θ_1, θ_2 and θ_3, and compute the corresponding probability (18) for each given agent s^i, and we choose the employment status $\overline{n}_j$ for which the probability $\Pr\left(n^i = \overline{n}_j\right)$ is the largest of three.

Verifying the employment decision: Compute $h_t = \sum_{i=1}^{\ell} z_t^i n^i$ and W_t from (6) and find the employment choice $\widehat{n}_t^i$ satisfying the optimality condition (17). Thus, the agents can check if the choices they made by using the assumed probabilities n^i coincide with the choices $\widehat{n}^i$ implied by the discretized FOC (17).

Training the machine: We form three likelihood functions L_j, $j = 1, 2, 3$ that correspond to three

[5] It is easy to find real-life situations in which this assumption is violated, for example, the voting preferences between two politicians can be affected by the presence or absence of another politician as an alternative.

outcomes $\overline{n}_j \in \left\{1, \frac{1}{2}, 0\right\}$:

$$\ln L_j\left(\theta_1, \theta_2, \theta_3\right) = \sum_{i=1}^{\ell} \mathcal{I}_j^i \ln\left(\frac{e^{\varphi\left(s^i, \theta_j\right)}}{e^{\varphi\left(s^i, \theta_1\right)} + e^{\varphi\left(s^i, \theta_2\right)} + e^{\varphi\left(s^i, \theta_3\right)}}\right) + \left(1 - \mathcal{I}_j^i\right) \ln\left(1 - \frac{e^{\varphi\left(s^i, \theta_j\right)}}{e^{\varphi\left(s^i, \theta_1\right)} + e^{\varphi\left(s^i, \theta_2\right)} + e^{\varphi\left(s^i, \theta_3\right)}}\right), \quad (19)$$

where $\mathcal{I}_j^i$ is an indicator function such that $\mathcal{I}_j^i = 1$ if outcome j occurs. We train the model to maximize these three likelihood functions.[6]

The DLC training procedure is essentially the same as for the model with just employed and unemployed agents: First, assume θ_1, θ_2 and θ_3, compute the corresponding probability (18), choose $n_t^i \in \left\{0, \frac{1}{2}, 1\right\}$ depending on which probability is the largest. Second, compute $h_t = \sum_{i=1}^{\ell} z_t^i n_t^i$ and W_t from (6), find the employed choices $\widehat{n}_t^i$ satisfying the discretized FOC (17). Finally, train the machine to maximize the likelihood functions (19) with respect to θ_1, θ_2 and θ_3. Ideally, the outcome of training is the three probability functions (18) that satisfy the fixed-point property such that the values of n_t^i induced by these probabilities coincide with the values $\widehat{n}_t^i$ implied by the discretized FOC (17) for each possible state.

The numerical solution to the model with three employment states produced by the DLC method is shown in Figure 3.

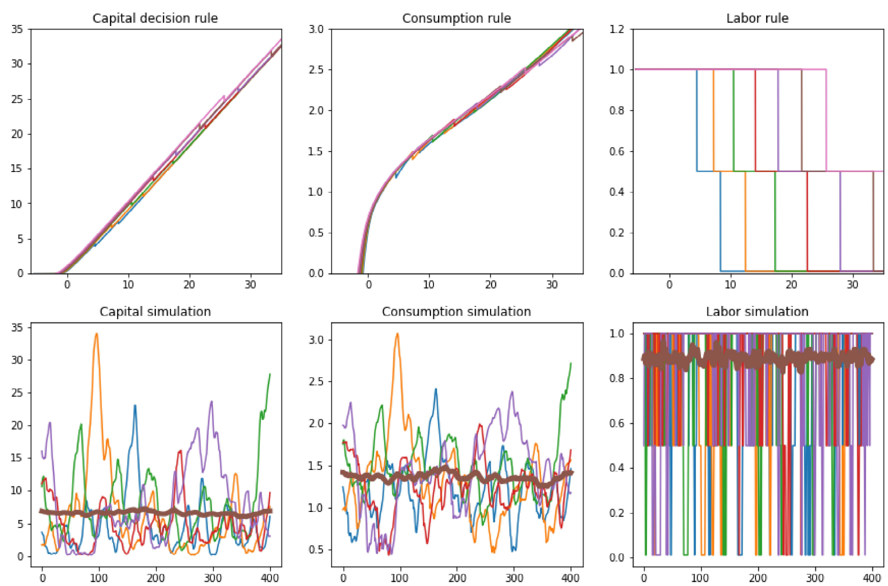

Figure 3: Model with full- and part-time employment under $\gamma = 1$ and $\eta = 1$.

The decision rules in the top row now experience two jumps instead of one (as in Figure 2) which correspond to switches between the full and partial employment, and the partial employment and unemployment (for each of the seven productivity levels). Switches occur later for more productive agents than for less productive agents. This is because the opportunity cost of leisure is higher for more productive agents.

The simulated series for the individual capital and consumption in the bottom row are similar to those in the previous figure but the individual labor switches in two discrete steps between full employment,

[6] In machine learning, a likelihood function with a negative sign is referred to as a "cross entropy loss" because of a connection with information theory. Thus, maximizing the likelihood function is equivalent to minimizing the cross-entropy loss.

partial employment and unemployment. Finally, in the bottom row, we show the fluctuations of the aggregate variables (see the thick lines) where we can distinguish how discrete changes in individual labor are transformed into discrete changes in aggregate labor.

We should finally remark that to solve the model with three employment states, it would be possible to use just one probability function similar to the one we used in the model with two probability states $p\left(s^i;\theta\right)$. We just need to split it into three intervals $\left[0,\overline{p}_p\right]$, $\left[\overline{p}_p,\overline{p}_f\right]$ and $\left[\overline{p}_f,1\right]$ that lead to the labor choices in the corresponding intervals $[0,\overline{n}_p]$, $[\overline{n}_p,\overline{n}_f]$ and $[\overline{n}_f,1]$. However, this approach is not always working well in practice, and it is not applicable to more general non-monotone classification problems.[7] In contrast, the softmax function approach we describe here is general, flexible and can be used to solve a variety of classification problems with any number of possible outcome.

5 Assessing the role of heterogeneity and labor choice

We discuss the aggregate and distributional implications of indivisible labor choice in Sections 5.1 and 5.2, respectively.

5.1 Aggregate implications

In Table 1, we provide selected business cycle statistics for the studied divisible- and indivisible-labor economies, as well as for the associated representative-agent model with divisible labor.

Table 1. Selected business cycle statistics.

	RA			HA Divisible labor			HA Indivisible labor			HA Full/Part time		
	$\eta=\frac{1}{5}$	$\eta=1$	$\eta=5$	$\eta=\frac{1}{5}$	$\eta=1$	$\eta=5$	$\eta=\frac{1}{5}$	$\eta=1$	$\eta=5$	$\eta=\frac{1}{5}$	$\eta=1$	$\eta=5$
$std(y)$	0.046	0.044	0.035	0.045	0.033	0.043	0.037	0.040	0.031	0.034	0.039	0.036
$\frac{std(c)}{std(y)}$	0.784	0.872	0.922	0.725	0.898	0.860	0.880	0.903	0.856	0.913	0.862	0.858
$\frac{std(n)}{std(y)}$	0.318	0.153	0.044	0.648	0.616	0.143	0.931	0.627	0.078	1.044	0.654	0.225
$\frac{std(h)}{std(y)}$	0.318	0.153	0.044	0.769	0.389	0.100	0.685	0.402	0.032	0.761	0.439	0.134
$\frac{std(i)}{std(y)}$	2.081	1.751	1.668	1.527	1.875	1.721	2.089	1.794	1.830	2.164	1.866	1.729
$\frac{std(y/n)}{std(y)}$	0.789	0.885	0.960	0.935	1.071	0.953	1.055	1.021	1.002	1.091	0.972	0.975
$corr(c,y)$	0.931	0.953	0.951	0.778	0.876	0.942	0.829	0.911	0.926	0.807	0.895	0.940
$corr(n,y)$	0.754	0.783	0.897	0.419	0.193	0.387	0.410	0.283	0.008	0.433	0.369	0.223
$corr(h,y)$	0.754	0.783	0.897	0.647	0.377	0.425	0.457	0.321	0.014	0.461	0.367	0.244
$corr(i,y)$	0.897	0.877	0.836	0.951	0.817	0.886	0.805	0.826	0.866	0.772	0.849	0.886
$corr(\frac{y}{n},n)$	0.964	0.994	1.000	0.779	0.824	0.990	0.584	0.805	0.997	0.500	0.778	0.974

Note: The statistics are computed across 100 simulations, each of which is length 1,000 periods; RA and HA refer to representative- and heterogeneous-agent models.

The business cycle statistics of all studied economies are typical for the real business cycle models. The volatility of output is somewhat higher than that in the US economy because the individual shocks contribute to the volatility of aggregate variables. To adjust for this effect, we report the ratio of volatilities of other variables relative to that of output.

It is well known that the representative-agent divisible-labor model is capable of accounting for stylized features of consumption-saving behavior over the US business cycle, however, has difficulties in reproducing the labor market statistics. In particular, it underpredicts the volatility of labor and overpredicts the correlation between labor and output and that between the labor and wage.

[7] For example, we cannot assign numerical values to classify musicians, artists and writers.

Concerning the first problem, the seminal works of Rogerson (1994) and Hansen (1993) had shown that the introduction of indivisible labor helps increase the volatility of labor. Under the assumption of complete markets, agents trade employment insurances, and the economy behaves as if the utility is linear in labor, which magnifies labor fluctuations compared to the economy where agents are risk averse with respect to labor choice. Chang et al. (2019) consider the model with intensive and extensive margins and find that this effect is quantitatively important, namely, the volatility of labor ranges between 30% and 60% of the output. Our analysis for the models with indivisible labor led to similar predictions for the benchmark case $\eta = 1$, however for $\eta = 5$, the volatility is excessively low; Chang et al. (2019) do not get so low volatility because they do not consider so high degrees of risk aversion, as we do.

Concerning excessively high correlation of labor variables, we can see in the table that the assumption of indivisible labor reduces the correlation $corr(n, y)$ from 0.7–0.8 to 0.4–0.2 relative to the representative-agent model. A similar reduction is observed for the correlation between efficiency labor and output. The reduction in correlation is due to the assumption of heterogeneity: in the representative-agent economy, a higher wage induces higher labor efforts, whereas in the heterogeneous-agent model, the efforts depend also on the individual productivity and the level of wealth which can offset some of the wage effect; see Maliar and Maliar (2003) for a related discussion.

5.2 Distributional implications

In Table 2, we summarize the distributional statistics produced by the heterogeneous-agent models.

Table 2. Distributional implications of the studied models.

	HA Divisible labor			HA Indivisible labor			HA Full/Part time		
	$\eta = \frac{1}{5}$	$\eta = 1$	$\eta = 5$	$\eta = \frac{1}{5}$	$\eta = 1$	$\eta = 5$	$\eta = \frac{1}{5}$	$\eta = 1$	$\eta = 5$
Income distribution									
Top 1%	0.015	0.036	0.032	0.033	0.032	0.029	0.031	0.032	0.030
Top 20%	0.270	0.398	0.374	0.385	0.368	0.358	0.371	0.371	0.359
Top 40%	0.520	0.636	0.610	0.630	0.606	0.595	0.612	0.610	0.594
Bottom 20%	0.082	0.071	0.083	0.068	0.081	0.088	0.072	0.079	0.089
Bottom 40%	0.254	0.193	0.214	0.193	0.214	0.224	0.205	0.211	0.226
Gini	0.190	0.327	0.292	0.320	0.288	0.272	0.300	0.294	0.271
Wealth distribution									
Top 1%	0.013	0.038	0.043	0.042	0.043	0.042	0.037	0.042	0.041
Top 20%	0.243	0.422	0.461	0.453	0.446	0.447	0.407	0.447	0.445
Top 40%	0.470	0.666	0.708	0.705	0.691	0.696	0.648	0.690	0.689
Bottom 20%	0.133	0.059	0.043	0.041	0.047	0.045	0.064	0.048	0.048
Bottom 40%	0.319	0.169	0.137	0.135	0.148	0.143	0.181	0.149	0.150
Gini	0.111	0.365	0.420	0.416	0.401	0.405	0.344	0.400	0.398

Note: The statistics are computed across 100 simulations, each of which is length 1,000 periods; RA and HA refer to representative- and heterogeneous-agent models.

Overall, the distributional implications of the studied models with indivisible labor are similar to those of the heterogeneous-agent model with intensive and extensive margins studied in Chang et al. (2019). A robust distributional implication of this class of models is that they underpredict the degree of inequality relative to the US economy; see, e.g., Aiyagari (1993) for the corresponding statistics on the US economy data. We see that the introduction of indivisible labor helps mitigate this problem and increase the degrees of inequality, in particular, the share of wealth that belongs to the top one percent of the population. Unlike the divisible labor model, the indivisible labor models are characterized by the degrees of income and wealth inequalities that do not significantly depend on the inverse of elasticity of intertemporal substitution of labor. However, the assumption of indivisible labor alone is not sufficient to produce the empirically relevant degrees of income and wealth inequalities as in the data.

6 Conclusion

This paper shows how to use deep learning classification approach borrowed from data science for modeling discrete choices in dynamic economic models. A combination of the state-of-the-art machine learning techniques makes the proposed method tractable in problems with very high dimensionality – hundreds of heterogeneous agents. We investigate just one example – discrete labor choice – but the proposed deep learning classification method has a variety of potential applications such as sovereign default models, models with retirement, and models with indivisible commodities, in particular, housing.

References

[1] Arellano, C. (2008). Default risk and income fluctuations in emerging economies. American Economic Review, 98(3), 690–712.

[2] Aiyagari, R. (1994). Uninsured idiosyncratic risk and aggregate saving, Quarterly Journal of Economics 109(3), 659-684.

[3] Azinovic, M., G. Luca and S. Scheidegger (2020). Deep equilibrium nets. SSRN: https://ssrn.com/abstract=3393482.

[4] Carroll, K. (2005). The method of endogenous grid points for solving dynamic stochastic optimal problems, Economic letters 91, 312-320.

[5] Chang, Y., and S. Kim (2007). Heterogeneity and Aggregation: Implications for Labor-Market Fluctuations, American Economic Review 97 (2007), 1939–56.

[6] Chang, Y., Kim, S.-B., Kwon, K. and R. Rogerson (2019). 2018 Klein lecture: individual and aggregate labor supply in heterogeneous agent economies with intensive and extensive margins. International Economic Review, Vol. 60 (1).

[7] Chatterjee, S., D. Corbae, M. Nakajima, and J. V. Ríos-Rull, (2007). A Quantitative Theory of Unsecured Consumer Credit with Risk of Default. Econometrica 75 (November): 1525-1589.

[8] Den Haan, W., (2010), Comparison of solutions to the incomplete markets model with aggregate uncertainty. Journal of Economic Dynamics and Control 34, 4–27.

[9] Duffy, J. and P. McNelis, (2001). Approximating and simulating the real business cycle model: parameterized expectations, neural networks, and the genetic algorithm, Journal of Economic Dynamics and Control 25(9), 1273-1303.

[10] Fernández-Villaverde, J., S. Hurtado, and G. Nuño, (2018). Financial frictions and the wealth distribution. Manuscript.

[11] Goodfellow, I., Y. Bengio, and A. Courville, (2016). Deep learning. Massachusetts Institute Technology Press.

[12] Hansen, G. (1985), Indivisible labor and the business cycle, Journal of Monetary Economics 16: 309-328.

[13] Iskhakov, F., Jørgensen, T., Rust, J. and B. Schjerning (2017). The endogenous grid method for discrete-continuous dynamic choice models with (or without) taste shocks. Quantitative Economics, 8(2):317–365 (lead article).

[14] Iskhakov, F. and M. Keane (2020). Effects of Taxes and Safety Net Pensions on Life-Cycle Labor Supply, Savings and Human Capital: the Case of Australia, Journal of Econometrics, 2020.

[15] Judd, K. L., L. Maliar, and S. Maliar, (2011). Numerically stable and accurate stochastic simulation approaches for solving dynamic models. Quantitative Economics 2, 173–210.

[16] Lepetyuk, V., L. Maliar and S. Maliar, (2020). When the U.S. catches a cold, Canada sneezes: a lower-bound tale told by deep learning. Journal of Economic Dynamics and Control, forthcoming.

[17] Krusell, P., and A. Smith, (1998). Income and wealth heterogeneity in the macroeconomy. Journal of Political Economy 106, 868–896.

[18] Maliar, L., and S. Maliar, (2003). The Representative consumer in the neoclassical growth model with idiosyncratic shocks. Review of Economic Dynamics 6, 362-380.

[19] Maliar, L., and S. Maliar, (2005). Parameterized expectations algorithm: how to solve for labor easily. Computational Economics 25, 269–274.

[20] Maliar, L., Maliar, S., and P. Winant, (2018). Deep learning for solving dynamic economic models. CEF-2018 presentation, https://www.youtube.com/watch?v=u7_CdytTEe8&feature=youtu.be.

[21] Maliar, L., Maliar, S., and P. Winant, (2019). Will artificial intelligence replace computational economists any time soon? CEPR working paper DP 14024.

[22] Prescott, E.C, Rogerson, R. and J. Wallentius (2009). Lifetime Aggregate Labor Supply with Endogenous Workweek Length, Review of Economic Dynamics 12 (2009), 23-36.

[23] Rogerson, R. (1988), Indivisible labor, lotteries and equilibrium, Journal of Monetary Economics 21: 3-16.

[24] Villa, A. and V. Valaitis (2019). Machine learning projection methods for macro-finance models, Manuscript.

Appendix A

In Sections A1, A2 and A3, we present the deep learning solution algorithms for the versions of Krusell and Smith (1998) model with divisible labor choice, indivisible labor choice, and part-time and full-time employment, respectively.

Our algorithm follows Maliar et al. (2018, 2019), except that we allow for divisible and indivisible labor choices. It alternates between the solution and simulation steps:

1. *Draw initial aggregate productivity* z_0 *and initial distributions* $\{K_0, Z_0\} \equiv \left\{k_0^i, z_0^i\right\}_{i=1}^{\ell}$*;*

2. *Construct labor* $\left\{n_0^i\right\}_{i=1}^{\ell}$ *using neural network approximation;*

3. *Compute the prices* R_0 *and* W_0*;*

4. *Compute consumption* $\left\{c_0^i\right\}_{i=1}^{\ell}$ *using the neural network approximation and find* $\left\{k_1^i\right\}_{i=1}^{\ell}$ *from the budget constraints;*

5. *Draw the uncorrelated shocks and train the neural network to satisfy the optimality conditions for* ℓ *agents;*

6. *Perform forward simulation to produce next-period individual and aggregate productivity* z_1 *and*$\{Z_1\} = \left\{z_1^i\right\}_{i=1}^{\ell}$.

Proceed iteratively until the convergence is achieved.

As the machine is trained and the panel is simulated, the decision functions are refined jointly with the ergodic distribution.[8] The studied method is similar in spirit to Krusell and Smith's (1998) one but is simpler conceptually as it does not involve construction of separate approximation of the law of motion for aggregate variables using the reduced state space. We just simulate the panel of heterogeneous agents, and we use the resulting distributions to infer both the individual and aggregate quantities and prices as the economy evolves over time.

[8] Since random variables are autocorrelated in our model, the stochastic gradient is correlated over time and hence, it is biased. To reduce the bias, we train the model on cross-sections which are sufficiently separated in time instead of using all consecutive periods.

A1. Solution method for divisible labor model

Algorithm 1: Deep learning for divisible labor model.
Step 0: (Initialization). Construct initial state of the economy $\left(\left\{k_0^i, z_0^i\right\}_{i=1}^{\ell}, z_0\right)$ and parameterize three decision functions by a neural network with three outputs $$\left\{n_t^i, \frac{c_t^i}{w_t^i}\right\} = \sigma\left(\zeta_0 + \varphi\left(k_t^i, z_t^i, \left\{k_t^i, z_t^i\right\}_{i=1}^{\ell}, z_t; \theta\right)\right),$$ $$\mu_t^i = \exp\left(\zeta_0 + \varphi\left(k_t^i, z_t^i, \left\{k_t^i, z_t^i\right\}_{i=1}^{\ell}, z_t; \theta\right)\right),$$ where $w_t^i \equiv R_t k_t^i + W_t \exp\left(z_t^i\right) n_t^i$ is wealth; μ_t^i is Lagrange multiplier associated with the borrowing constraint; $\varphi(\cdot)$ is a neural network; $\sigma(x) = \frac{1}{1+e^{-x}}$ is a sigmoid (logistic) function; ζ_0 is a constant; θ is a vector of coefficients (biases and weights).
Step 1: (Evaluation of decision functions).
Given state $\left(\left\{k_t^i, z_t^i\right\}_{i=1}^{\ell}, z_t\right)$, compute n_t^i, w_t^i, $\frac{c_t^i}{w_t^i}$ from the decision rules and find k_{t+1}^i from the budget constraint for all agents $i = 1, ..., \ell$.
Step 2: (Construction of Euler residuals).
Draw two random sets of individual productivity shocks $\Sigma_1 = \left(\epsilon_1^1, ..., \epsilon_1^{\ell}\right)$, $\Sigma_2 = \left(\epsilon_2^1, ..., \epsilon_2^{\ell}\right)$ and two aggregate shocks $\epsilon_{1,}$, ϵ_2. Construct the Euler residuals, $$\mathcal{R}_1 = \left\{\left[\Psi^{FB}\left(1 - \frac{w^i}{c^i}, 1 - \mu^i\right)\right]^2 + v\left[\frac{\beta u'\left(c^{i\prime}\left(\Sigma_1, \epsilon_{1,}\right)\right)}{u'(c^i)}\left(1 + R_t\right) - \mu^i\right]\left[\frac{\beta u'\left(c^{i\prime}\left(\Sigma_2, \epsilon_{2,}\right)\right)}{u'(c^i)}\left(1 + R_t\right) - \mu^i\right],$$ where $\Psi^{FB}(a, b) = a + b - \sqrt{a^2 + b^2}$ is a Fischer-Burmeister function.
Step 3*:* (Construction of labor-choice residuals).
Construct the residuals in the labor choice by $$\mathcal{R}_2 = \left(n_t^i - 1 - \left[\frac{c_i^{-\gamma} W_t \exp\left(z_t^i\right)}{B}\right]^{-1/\eta}\right)^2.$$
Step 4*:* (Training).
Train the neural network coefficients θ to minimize a weighted sum of two residuals $\mathcal{R}_1 + \nu \mathcal{R}_2$ across the agents $i = 1, ..., \ell$ and batches, where ν is an exogenous weight.
Step 5*:* (Simulation).
Move to $t + 1$ by using endogenous and exogenous variables obtained in Step 4 under $\Sigma_1 = \left(\epsilon_1^1, ..., \epsilon_1^{\ell}\right)$ and ϵ_1 as a next-period state $\left(\left\{k_{t+1}^i, z_{t+1}^i\right\}_{i=1}^{\ell}, z_{t+1}\right)$.

A2. Solution method for indivisible labor model

Algorithm 2: Deep learning for indivisible labor model.

Step 0: (Initialization). Construct initial state of the economy $\left(\left\{k_0^i, z_0^i\right\}_{i=1}^{\ell}, z_0\right)$ and parameterize the decision functions by

$$\left\{\ln \frac{p_t^i}{1-p_t^i}, \frac{c_t^i}{w_t^i}\right\} = \sigma\left(\zeta_0 + \varphi\left(k_t^i, z_t^i, \left\{k_t^i, z_t^i\right\}_{i=1}^{\ell}, z_t; \theta\right)\right),$$

$$\mu_t^i = \exp\left(\zeta_0 + \varphi\left(k_t^i, z_t^i, \left\{k_t^i, z_t^i\right\}_{i=1}^{\ell}, z_t; \theta\right)\right),$$

where p_t^i is the probability of being employed $n_t^i = 1$.

Step 1: (Evaluation of decision functions).

Given state $\left(\left\{k_t^i, z_t^i\right\}_{i=1}^{\ell}, z_t\right) \equiv s_t^i$, compute $n_t^i = \overline{n}$ if $p_t^i \geq \frac{1}{2}$ and $n_t^i = 0$ if $p_t^i < \frac{1}{2}$. Compute w_t^i and $\frac{c_t^i}{w_t^i}$ from the decision rules and find k_{t+1}^i from the budget constraint for all agents $i = 1, ..., \ell$.

Step 2: (Construction of Euler residuals). ...

Step 3*:* (Construction of labor-choice residuals).

Construct the residuals in the labor choice by

$$\mathcal{R}_2 = \left[\sum_{i=1}^{\ell} \widehat{n}_t^i \ln\left(p\left(s_t^i; \theta\right)\right) + \left(1 - \widehat{n}_t^i\right) \ln\left(1 - p\left(s_t^i; \theta\right)\right)\right]^2$$

$$\text{where } \widehat{n}_t^i = \begin{cases} 1 \text{ if } L - \left[\frac{c_i^{-\gamma} W_t \exp\left(z_t^i\right)}{B}\right]^{-1/\eta} \geq \overline{n}_f \\ 0 \text{ otherwise.} \end{cases}.$$

...

A3. Solution method for the model with full and part time employment

Algorithm 3: Deep learning for the model with full and partial employment.
Step 0: (Initialization). Construct initial state of the economy $\left(\left\{k_0^i, z_0^i\right\}_{i=1}^{\ell}, z_0\right)$ and parameterize the decision functions by $$\left\{\ln \frac{p_t^i(1)}{\Delta_t^i}, \ln \frac{p_t^i(\frac{1}{2})}{\Delta_t^i}, \ln \frac{p_t^i(0)}{\Delta_t^i}, \frac{c_t^i}{w_t^i}\right\} = \sigma\left(\zeta_0 + \varphi\left(k_t^i, z_t^i, \left\{k_t^i, z_t^i\right\}_{i=1}^{\ell}, z_t; \theta\right)\right),$$ where $p_t^i(1)$, $p_t^i\left(\frac{1}{2}\right)$ and $p_t^i(0)$ are the probabilities to be full- and part-time employed and unemployed, respectively, and $\Delta_t^i \equiv p_t^i(1) + p_t^i\left(\frac{1}{2}\right) + p_t^i(0)$ is a normalization of probability to one.
Step 1: (Evaluation of decision functions).
Given state $\left(\left\{k_t^i, z_t^i\right\}_{i=1}^{\ell}, z_t\right) \equiv s_t^i,$, set $n_t^i = 1$, $n_t^i = \frac{1}{2}$ and $n_t^i = 0$ depending on which probability $p_t^i(1)$, $p_t^i\left(\frac{1}{2}\right)$ and $p_t^i(0)$ is the largest. Compute w_t^i, $\frac{c_t^i}{w_t^i}$ from the decision rules and find k_{t+1}^i from the budget constraint for all agents $i = 1, \ldots \ell$.
Step 2: (Construction of Euler residuals). ...
Step 3*:* **(Construction of labor-choice residuals).**
Construct the residuals in the labor choice by $$\mathcal{R}_2 = \sum_{j=1}^{3}\left[\sum_{i=1}^{\ell} \widehat{n}_t^i \ln\left(\frac{e^{\varphi\left(s^i,\theta_j\right)}}{e^{\varphi\left(s^i,\theta_1\right)}+e^{\varphi\left(s^i,\theta_2\right)}+e^{\varphi\left(s^i,\theta_3\right)}}\right) + \left(1-\widehat{n}_t^i\right)\ln\left(1-\frac{e^{\varphi\left(s^i,\theta_j\right)}}{e^{\varphi\left(s^i,\theta_1\right)}+e^{\varphi\left(s^i,\theta_2\right)}+e^{\varphi\left(s^i,\theta_3\right)}}\right)\right]^2$$ $$\text{where } \widehat{n}_t^i = \begin{cases} 1 \text{ if } L - \left[\frac{c_i^{-\gamma} W_t \exp\left(z_t^i\right)}{B}\right]^{-1/\eta} \geq \overline{n}_f \\ \frac{1}{2} \text{ if } L - \left[\frac{c_i^{-\gamma} W_t \exp\left(z_t^i\right)}{B}\right]^{-1/\eta} \in \left[\overline{n}_p, \overline{n}_f\right] \\ 0 \text{ otherwise} \end{cases} ..$$ ...

A4. Remarkable features of deep learning that help to deal with the curse of dimensionality

Maliar et al. (2018, 2019) argue that three remarkable features of deep learning approach help us deal with the curse of dimensionality. We discuss these features below.

Ergodic-set domain. We solve the model on simulated series (ergodic set) instead of a rectangular-style domain that classical projection methods (like Smolyak) use. The volume of the rectangular domain is huge in high-dimensional problems, and it is prohibitively expensive to attain accurate approximation everywhere on such a huge domain. In contrast, only an infinitesimally small fraction of rectangular domain is generally visited in equilibrium in high-dimensional applications; see Judd et al. (2011) for a discussion. By solving the model on simulated series, we restrict attention to a relatively small ergodic-set domain in which the solution "lives" – this helps us deal with the curse of dimensionality.

Perfect multicollinearity. In the approximating function of the consumption share, we include the state variables of agent i twice $\varphi\left(k_t^i, z_t^i, \left\{k_t^i, z_t^i\right\}_{i=1}^{\ell}, z_t; \theta\right)$, namely, they enter both as variables of agent i and as an element of the distribution. This repetition implies perfect collinearity in explanatory variables, so that the inverse problem is not well defined. Such a multicollinearity would break down a conventional numerical method which solves the inverse problem but neural networks can learn to ignore redundant colinear variables, as we have shown earlier. Thus, even though it is possible to design a transformation that avoids a repetition of variables, it would require cumbersome and costly permutations, so we find it easier to keep the repeated variables.

Model reduction. We solve the models with hundreds of heterogeneous agents (and thus, state variables). How can the deep learning method deal with such a huge state space? In addition to the ability to handle multicollinearity, neural networks possess another remarkable property: they automatically perform the model reduction. When we supply a large number of state variables to the input layer, the neural network condenses the information into 64 neurons of two hidden layers, making it more abstract and compact. In a sense, this procedure is similar to a photo compression or principal component transformation when a large set of variables is condensed into a smaller set of principal components without losing essential information; see Goodfellow et al. (2016) for a discussion of neural networks.

Krusell and Smith (1998) found one specific model reduction that works extremely well for their model, namely, they approximate the distribution of state variables with a finite set of moments. They found that in their model, just one moment – a mean of wealth distribution m_t – is a sufficient statistic for capturing all relevant information, which reduces their state space to just four state variables (k_t^i, z_t^i, z_t, m_t).

If Krusell and Smith's (1998) construction is the most efficient representation of the state space, the neural network is likely to find this representation as an outcome of training. However, the neural network will automatically consider many other possible ways of extracting the information that is contained in the distribution $\left\{k_1^i, z_1^i\right\}_{i=1}^{\ell}$ and condensing it in a relatively small set of hidden layers. The output of the machine can look like moments or some other statistics – we will not always be able to understand how the machine handles the information in the hidden layers but this fact does not prevent us from using this remarkable technology in applications.

Appendix B.

In this appendix, we show sensitivity results with respect to the inverse of intertemporal elasticity of substitution of labor η for the models with divisible labor, indivisible labor model and full- and part-time employment.

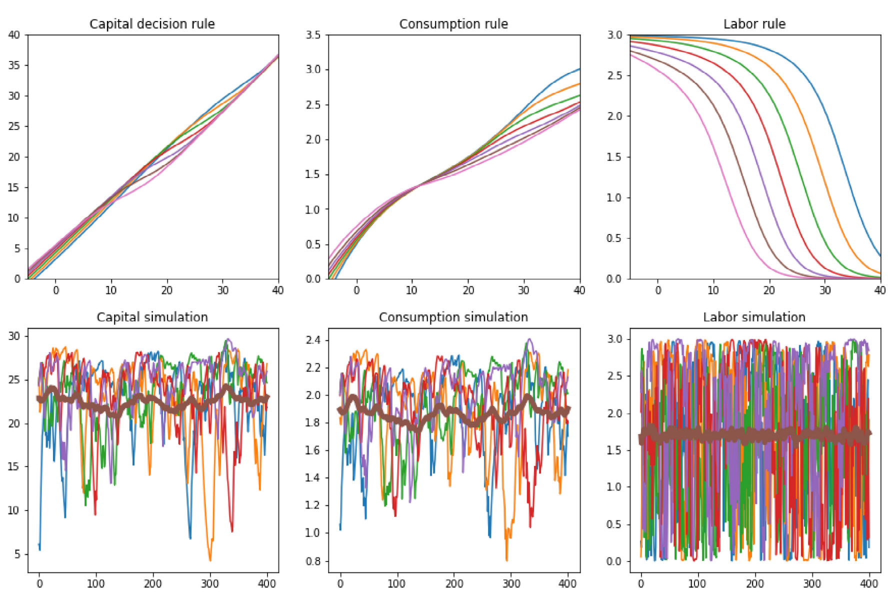

Figure 4: Divisible labor model under $\gamma = 1$ and $\eta = 0.2$.

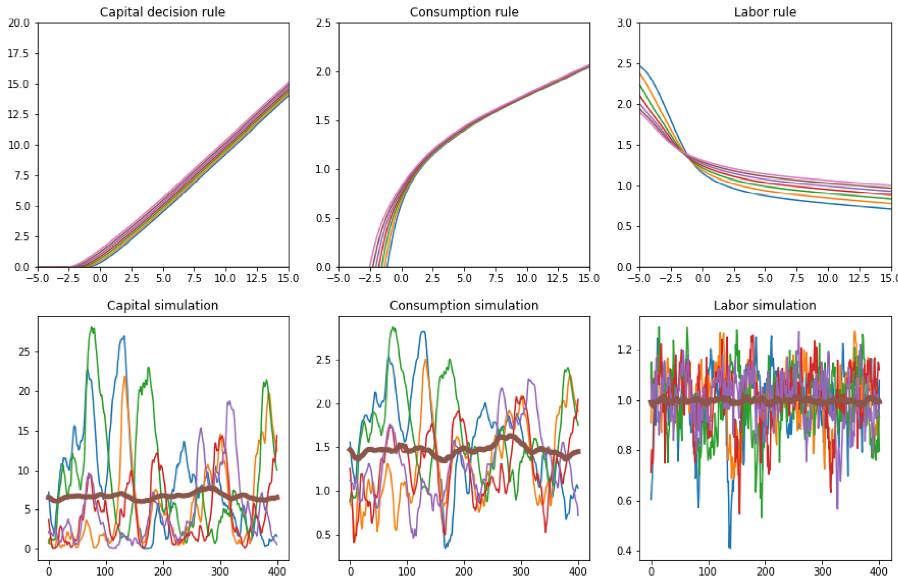

Figure 5: Divisible labor model under $\gamma = 1$ and $\eta = 5$.

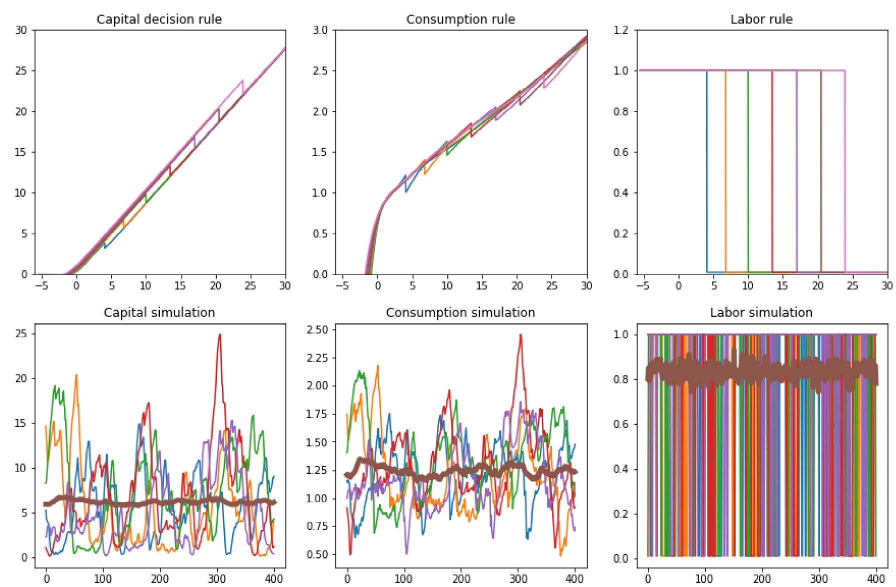

Figure 6: Indivisible labor model under $\gamma = 1$ and $\eta = 0.2$.

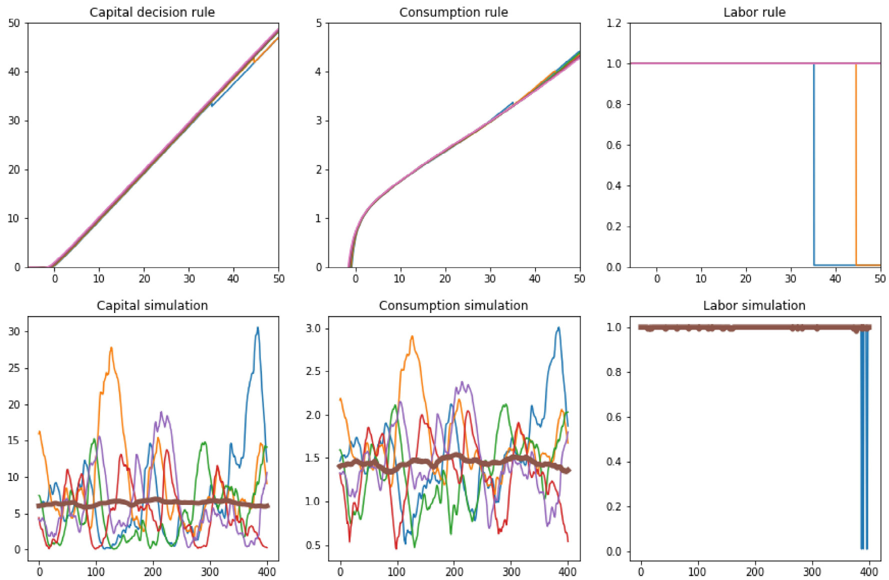

Figure 7: Indivisible labor model under $\gamma = 1$ and $\eta = 5$.

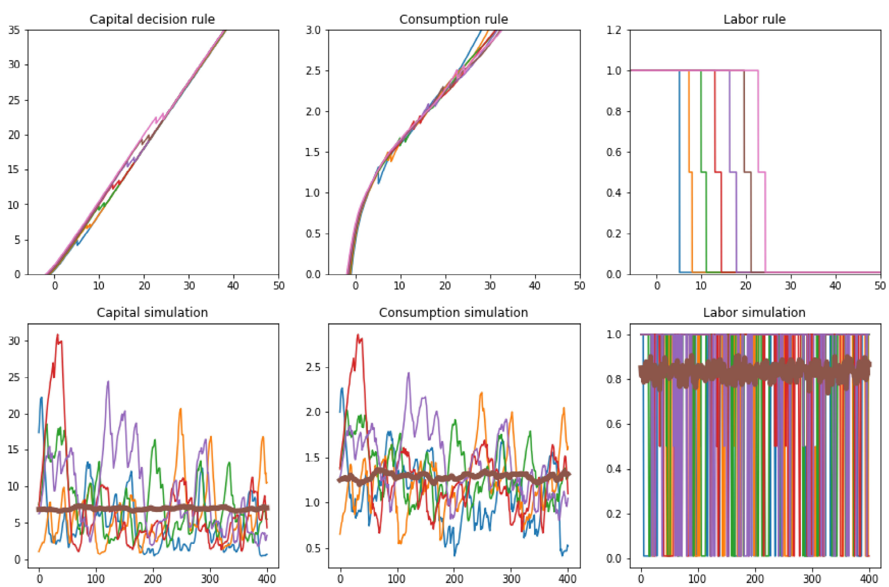

Figure 8: Model with full- and part-time employment under $\gamma = 1$ and $\eta = 0.2$.

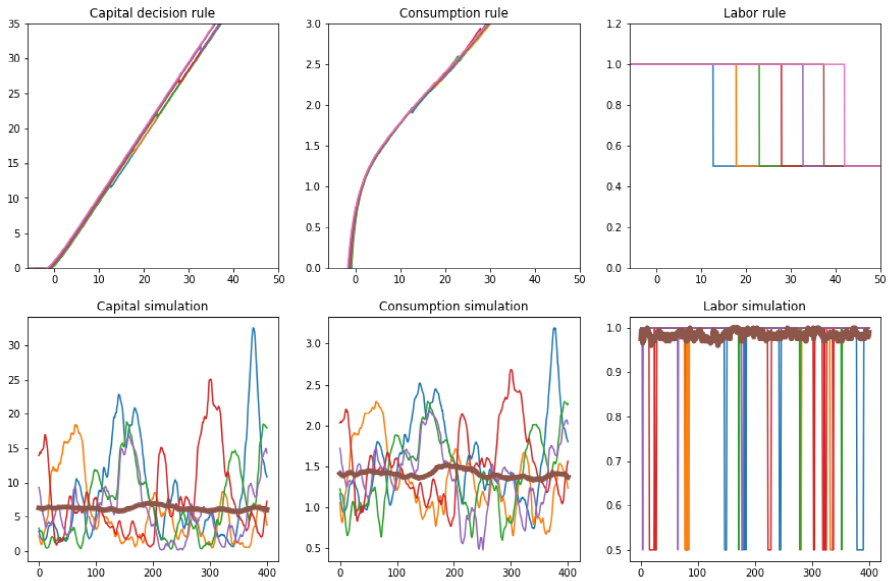

Figure 9: Model with full- and part-time employment under $\gamma = 1$ and $\eta = 5$.