Machine Learning Projection Methods for Macro-Finance Models

Alessandro Villa and Vytautas Valaitis*

February 14, 2018

Abstract

This paper develops a global solution method to solve large state space macro-finance models using machine learning. Our new method, an artificial neural network expectation algorithm, is not only considerably faster but also as precise and more scalable than the standard parametrized expectations algorithm. We illustrate the advantages of the method in an optimal fiscal policy problem with a large and highly multicollinear set of state variables. We solve the model with multiple maturities and Epstein-Zin preferences and explain in detail how to use the artificial neural network in this context. To conclude, we suggest two relevant applications in asset pricing and learning, and financial intermediation with endogenous default.

Keywords: Machine Learning, Projection Methods, Optimal Fiscal Policy.

JEL classification: C63, E32, E37, E62, G12.

As noted in Mullainathan & Spiess (2017), applying machine learning to economics requires finding relevant tasks and, specifically, problems where prediction plays a crucial role. In this particular application, we are interested in efficiently and accurately generating predictions for various expectation terms. We find that an artificial neural network arrives at

*Villa: Duke University, alessandro.villa@ duke.edu, Valaitis: Duke University, vytautas.valaitis@ duke.edu.

the same solution as the one calculated by standard methods, but in a lesser time and in a more scalable way.

In general, machine learning techniques are particularly useful when dealing with a large number of state variables and when the explicit relation between the predictors and predicted variables is not known. In this paper, we suggest that machine learning should not be considered only for its possible econometrics applications but also as a handy computational tool to solve large state space problems. In this last context, machine learning can be used for optimization or in the context of global approximation methods.

On the one hand, consider two mainstream and widely used dynamic programming (DP) methods in economics: policy function and value function iteration. Thanks to the introduction of machine learning, methods like neuro-dynamic programming allow to attack larger state space problems through the combination of DP techniques, simulation, and iterative procedures to better approximate optimal cost (or policy) functions.

On the other hand, global approximation methods are gaining more and more popularity over local ones. Global methods often rely on the projection of relevant policies on the information set. In this paper, we present an application of projection methods and machine learning in solving an optimal fiscal policy problem. In particular, we use an artificial neural network to solve a Ramsey taxation problem with incomplete markets. We consider this application particularly relevant for several reasons. Firstly, the state space that needs to be considered explodes in function of the length of the maturities and the number of bonds. Secondly, this class of problems includes forward-looking constraints and the commonly used recursive representation (Bellman equation) cannot be adopted in this case. Following Marcet & Marimon (2011), we formulate the recursive Lagrangian to solve for the time-inconsistent optimal contract under full commitment. When markets are incomplete the Ramsey planner needs to keep track of all the promises made in the previous periods. This requires to add extra state variables, which in turn increase the state space and creates history dependence. Thirdly, the state variables tend to be highly multicollinear, requiring further extensions to the standard techniques: the standard Parametrized Expectations Algorithm (PEA) is replaced by the so called Condensed PEA. Our method becomes particularly useful as the model becomes increasingly complex: the length of the longest maturity and the number of debt instruments increase. A closely related paper to ours is Duarte (2018) who proposes an application of machine learning to macro-finance models. In particular he uses neuro-dynamic programming to replace polynomials in approximating the value function and solves the Hamilton-Jacobi-Bellman equation. Our method is different from his in multiple ways. First, rather than approximating the value function, we approximate the expected value terms. Secondly, our method does not rely on a continuous time representation of the model and lastly, it does not require to build a grid and does not suffer from the curse of dimensionality. The use of grid free methods comes at a cost: the state variables tend to be highly collinear. However artificial neural networks prove to be robust to multicollinearity.

The paper is organized as follows. Section 1 presents a baseline version of a model where government can issue only one type of non state-contingent bond. Section 2 extends the model to a generic N non-state contingent bonds with different maturities and Epstein-Zin preferences. Section 3 outlines a generic solution method for a generic N bonds model. We report results for the one, two and three bonds cases comparing our method with the standard approach. We also report the solution for the two bonds case with Epstein-Zin preferences. Section 4 suggests other two relevant applications in finance and macroeconomics. Section 5 concludes.

1 Illustrative case: one bond economy

1.1 Setting

The model we consider in this section is the one proposed in Aiyagari et al. (2002). It is a version of the stochastic neoclassical model with incomplete markets and a Ramsey planner. The economy is populated by a representative household that has preferences over consumption and leisure and maximizes the expected lifetime utility:

$$\mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t [u(c_t) + v(l_t)]$$

Subject to the budget constraint:

$$p_t b_t^N + c_t = (1 - \tau_t)(1 - l_t) + b_{t-1}^N$$

Where the superscript on b_t indicates that it is a N-periods maturity bond. In each period the aggregate endowment in the economy is A units that can be used for consumption, leisure and government expenditure. This leads to the aggregate resource constraint $c_t + g_t = A - l_t$, where $A - l_t$ is the period's GDP. The government needs to finance an exogenous stream of government expenditure $\{g_t\}_{t=0}^{\infty}$. It does that by setting proportional labor taxes τ_t and by issuing non state contingent bonds with maturity of N periods, sold at the price p_t which, at the optimum, it coincides with the household's stochastic discount factor. This gives the following budget constraint for the government:

$$g_t + b_{t-1}^N = \tau_t(1 - l_t) + p_t b_t^N$$

For simplification we assume that government can buy back and reissue the entire stock of the outstanding debt in each period, also known as the buyback assumption in the literature. The purpose of the government is to solve the Ramsey taxation problem: set taxes and issue debt to maximize welfare over the competitive equilibrium outcomes. Using the Primal approach and assuming upper and lower bound for government debt, we can express the government's problem as:

$$\max_{\{c_t\}_{t=0}^{\infty},\{b_t\}_{t=0}^{\infty}} \mathbb{E}_0 \sum_t \beta^t \left[u(c_t) + v(A - c_t - g_t)\right]$$

Subject to a sequence of time t measurability constraints[1]:

$$b_t^{(N)} \beta^N u_{c,t+N} - b_{t-1}^N \beta^{N-1} u_{c,t-1+N} - g_t u_{c,t} + (u_{c,t} - v_{l,t})(g_t + c_t) = 0$$

And borrowing limits:

$$\frac{\bar{M}_N}{\beta^N} \geq b_t^N \quad \frac{\underline{M}_N}{\beta^N} \leq b_t^N$$

The optimality conditions are:

$$u_{c,t} - v_{l,t} + \lambda_t(u_{cc,t}c_t + u_{c,t} + v_{ll,t}(c_t + g_t) - v_{l,t}) + u_{cc,t}(\lambda_{t-N} - \lambda_{t-N+1})b_{t-N}^N = 0 \tag{1}$$

$$\lambda_t = \frac{\mathbb{E}_t(u_{c,t+N}\lambda_{t+1})}{\mathbb{E}_t(u_{c,t+N})} \tag{2}$$

$$b_t^N \beta^N \mathbb{E}_t(u_{c,t+N}) = b_{t-1}^N \beta^{N-1} \mathbb{E}_t(u_{c,t+N-1}) - g_t u_{c,t} - (u_{c,t} - v_{l,t})(g_t + c_t) \tag{3}$$

[1] See AMSS (2002) for details on how to use the recursive lagrangian approach in this context

Where λ_t is the multiplier on time t period measurability constraint. By issuing debt at time t, the government commits to increase taxes, or reissue debt at time $t+N$. Such past actions must be taken into account by the government, when it sets taxes at any of the periods between t and $t+N$. That is why all the lags of the state variables up to N form a state space. More formally, the Ramsey planner's relevant state variable vector X_t is:

$$X_t = \left\{ g_t, \{\lambda_{t-i}\}_{i=1}^N, \{b_{t-i}^i\}_{i=1}^N \right\}$$

The focus and the main contribution of this paper is on the solution method. Therefore, we abstain from performing a calibration and, instead, assign reasonable values to model parameters which can be found in Table 6 in Appendix D. For now, we use a standard CRRA utility for both consumption and leisure.

1.2 Solution

Because the state space includes $2N+1$ variables, which are highly multicollinear, the model is hardly solvable using standard methods, such as Parametrized Expectations Algorithm (PEA). In general, the expected value terms in the first order conditions need to be approximated using some functions of a core set[2] of the state variables. The model can be solved by iterating on the optimality conditions and updating the approximating functions for the unknown expected terms until the approximated expected value becomes consistent with the dynamics of the system. That is: parametrize the expected value terms with some functions of the state variables, iterate on the system of stochastic difference equations (1)-(3) for a large time horizon T, perform regressions of equations (4)-(6) (given the model generated data) and obtain the new coefficients for the approximating functions. Iterate till convergence: when the predicted dynamic of the system doesn't change anymore and the predicted values on the expectations are accurate enough.

More formally, the solution involves approximating the following expectations as a func-

[2]A subset of the entire information set is selected to avoid multicollinearity

tion of the state variables:

$$\mathbb{E}_t(u_{c,t+N}) \simeq f_1(g_t, \{\lambda_{t-i}\}_{i=1}^N, \{b_{t-i}^i\}_{i=1}^N) \quad (4)$$

$$\mathbb{E}_t(u_{c,t+N-1}) \simeq f_2(g_t, \{\lambda_{t-i}\}_{i=1}^N, \{b_{t-i}^i\}_{i=1}^N) \quad (5)$$

$$\mathbb{E}_t(u_{c,t+N}\lambda_{t+1}) \simeq f_3(g_t, \{\lambda_{t-i}\}_{i=1}^N, \{b_{t-i}^i\}_{i=1}^N) \quad (6)$$

In the next subsections we briefly describe the existing algorithm in literature and our approach, concluding with a comparison between them.

1.2.1 Condensed PEA

The solution method existing in literature, capable to deal with the multicollinearity issue is called Condensed PEA. The following is a brief high level description of how it works (see Faraglia et al. (2014b) for more details):

1. Parametrize the 3 expectations in equations (1)-(3) as functions of a subset of state variables (called core set) and given an initial guess of the polynomials parameters[3]
   - Set the bounds for the bond (see Maliar and Maliar (2003))
   - Simulate the model given the parameters
   - Using the simulated data, run a regression of each element on the core variables to get the new parameters values
   - Iterate and stop when the prediction matches the simulated data
2. Regress the remaining state variables on the core set and save the residuals. Then regress the realized values of the 3 expectations on the core set and the saved residuals. Add these residuals, multiplied by the estimated coefficients, to the core set and go back to point 1 till convergence on the path of debt is reached

This method keeps extracting orthogonal components from the information set, similarly to the Principle Component Analysis (PCA), but the number of factors does not have to

[3] Initial parameters can be given by a simulated sequence with $\{b_t\}_{t=0}^T$

be decided ex-ante. According to our practice, commonly used methods to deal with multicollinearity, such as PCA or Ridge and Lasso regressions, could not converge to any reliable solution.

1.2.2 Solving with Artificial Neural Networks (ANN)

We explore the possibility of using machine learning in this context because of two salient features of this class of algorithms: 1. Robustness to multicollinearity. 2. Non-parametric nature, which does not require commitment to a pre-specified functional form. In particular, we decided to use an ANN with the characteristics specified in Table 1. Neural networks are mainly based on the simulation of properly linked artificial neurons, which receive stimuli (inputs) and elaborate them. Each input, connected to each neuron, is multiplied by a specific weight. All the results of these multiplications are added and, if the sum exceeds a certain threshold, the neuron activates by activating its output (neurons are modeled as transfer functions). The weight indicates the synaptic efficacy of the input line and serves to quantify its importance. A very important input will have a high weight, while a less important input will have a lower weight. Neural networks can feature multiple layers and each layer can have a specific number of inputs. Increasing the number of layers and neurons increases the capacity of the network to learn but, at the same time, can potentially lead to overfitting and slow down computation in the learning phase. We explain how to choose the number of neurons optimally later in this section. After weights are assigned with initial values, data can be divided into two groups: training set and validation set. The training set is used for the learning phase, where the network produces outputs, given initial weights, and compare its outputs with the values in the training set. The error resulting from this comparison can be used by a training algorithm to adjust the weights in order to match the data in the training set. An epoch is one complete presentation of the training set to the neural network. The learning rate represents the speed at which the neural network learns. After having been trained, the network enters a validation phase. In this phase outputs are produced using the inputs associated with the data points in the validation set and given the weights computed during the learning phase. Produced outputs and realized ones are compared, for example by calculating the Mean Squared Error (MSE) which represents the prediction power of the neural network out-of-sample.

Parameter	Value
Hidden layers	1
Neurons	7
Transfer function	Hyperbolic tangent sigmoid
Training algorithm	Gradient descent with adaptive learning rate backpropagation
Learning rate	0.001
Ratio to increase learning rate	1.01
Ratio to decrease learning rate	0.9
Epochs	1000

Table 1: ANN specifications

We chose to operate with only one layer since a single-hidden layer ANN is faster and, at the same time, it is perfectly capable of catching all the non linearities we might need in this application. The number of neurons is calibrated in our simulations to avoid overfitting. In particular, given the parameters in Table 6, the number of neurons has been chosen such that the MSE calculated out-of-sample (on the validation set) is minimized. This is to maximize the prediction power of the ANN and, at the same time, have the highest number of neurons that provides the best fit on the training set. In general, more neurons always provide a better fit in-sample. The problem is the fitting out-of-sample which, after a certain point, stops to improve and starts to diverge. We implement this intuition choosing the maximum number of neurons such that the fitting out-of-sample is not worsening. The learning rate is chosen to be small on purpose. Intuitively, the learning rate describes how quickly an ANN abandons old beliefs for new ones. In this particular application we are concerned in obtaining a stable solution, which should fluctuate as little as possible between iterations (given the unstable nature of the problem) and, therefore, we decided to keep the learning rate small. We allow for adaptive learning with asymmetric increasing/decreasing ratio, privileging the decreasing side. This is to be conservative and create continuity among solutions calculated in different iterations.

The following points summarize how the solution algorithm works in this case:

- Start with an initial guess for the weights of the ANN (initializing sequences are generated as described above)
- Use the entire information set $\left\{g_t, \{\mu_{t-1}\}_{i=1}^N, \{b_{t-i}^i\}_{i=1}^N\right\}$ as an input of the ANN and get predictions for $\mathbb{E}_t(u_{c,t+N})$, $\mathbb{E}_t(u_{c,t+N-1})$ and $\mathbb{E}_t(u_{c,t+N}\lambda_{t+1})$.
- Combine predictions and optimality conditions to generate an implied model dynamics
- Use the model generated data to perform supervised learning of the ANN
- Start again from the beginning with the newly trained ANN till convergence is reached (predictions match with the implied dynamics)

The main advantage of this method comes from the possibility to feed the entire information set to the ANN. The regression approach needs to perform additional cycles in order to find a core set of regressors able to deliver enough prediction power to avoid inaccuracies due to multicollinearity. In the ANN approach, neurons connected to redundant information are likely to switch off automatically (weights go to zero), or to adjust their importance in accordance to the weights of the other neurons connected to the same redundant information. These adjustments, which automatically happen during the learning phase, offer the possibility to deal more efficiently with an increasingly big information set, as shown in the following example.

1.2.3 Performance comparison

We report an example where we solve the same problem in the same computational environment: one bond with 10 periods maturity, same exogenous sequence $\{g_t\}_{t=0}^T$ (generated as white noise with positive constant mean) and parameters values as reported in Table 6 in Appendix D.

	ANN		C. PEA	
Projected term	*Residual*	$Residual_{\%}$	*Residual*	$Residual_{\%}$
$\mathbb{E}_t(u_{c,t+N}\lambda_{t+1})$	0.037	2.3%	0.041	2.7%
$\mathbb{E}_t(u_{c,t+N})$	0.155	2.2%	0.181	2.6%
$\mathbb{E}_t(u_{c,t+N-1})$	0.155	2.2%	0.185	2.6%
Time	801s		14080s	

Table 2: Bonds bounds are the natural ones (in this case around 10 times the GDP). Both methods converge to comparable solutions: The mean absolute deviation between the debt path calculated through C.PEA and ANN is 0.050. $Residual = \frac{1}{T}\sum\left|Y_i - \hat{E}_i\right|$ and $Residual_{\%} = \frac{1}{T}\sum\left|\frac{Y_i-\hat{E}_i}{Y_i}\right|$.

As shown in Table 3, the ANN approach provides slightly more precise forecasts on all three expectation terms. The key message from this table is that, given a comparable prediction precision and a comparable solution, the ANN approach takes around 95% less time. The main reason is that the Condensed PEA needs to try different combinations of core regressors, whereas the ANN needs to iterate only one time digesting the entire information set at once. Moreover, at each step of the Maliar bounds, or at each refinement step once the Maliar bounds are completely open, the Condensed PEA approach requires to run 3 separate regressions whereas the ANN approach requires only one training phase for all three predictions. Using the same ANN to predict the three outputs at once is not only faster but might also help catching correlations between predicted terms. On the other hand, it requires more time to train the ANN than to run a regression and the training time increases with the number of layers. However, a single-layer ANN is perfectly capable to capture the nonlinearities in this model. In summary, the ANN approach required 1 single iteration on the information set which took 801s, whereas the Condensed PEA approach required 12 iterations on the information set (to find the right combination of regressors) and, each of them, required on average around 1173s.

1.3 An example with a predictable g process

For illustration purposes, we present our solution to a model where government expenditure follows an AR(1) process. As shown in Figure 1, the government debt dynamic is closely linked to the expenditure process - periods when both are decreasing coincide. At the same time, b_t is more persistent[4] than g_t and there is a tendency for the government to accumulate assets in the long-run as found in Aiyagari et al. (2002). The dynamics of other model variables, as well as the prediction errors, are presented in Figures 6 and 7 in Appendix A. As shown in Figure 3, the ANN offers particularly good predictions for the $\mathbb{E}_t[u_{c,t+N}\lambda_{t+1}]$ term. Even though the forecasts of the terms involving only marginal utility leave room for higher residuals. This fact is reasonable since the rational expectation for consumption N periods ahead is the steady state.

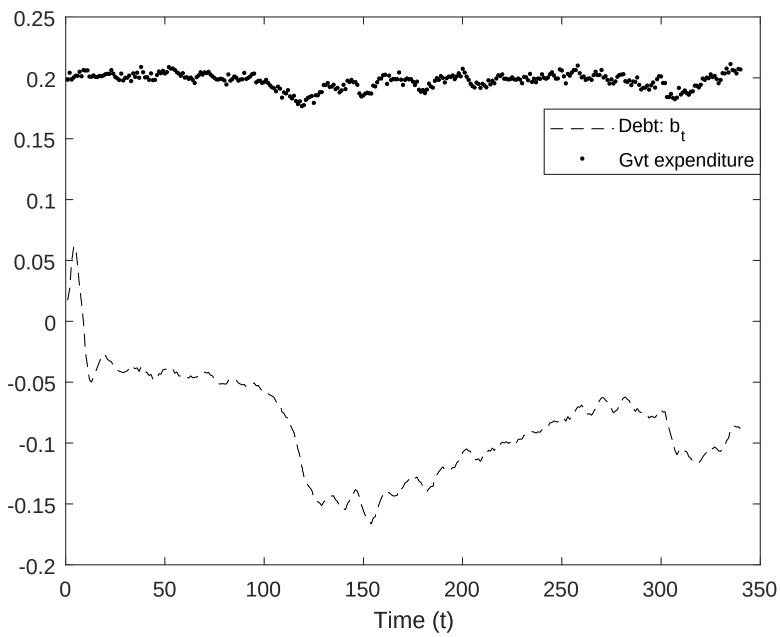

Figure 1: Solution to the one bond model using neural network. g_t follows AR(1) process

In general, we would like to emphasize several important observations from our computational results. First, while increasing the length of the maturity increases the number of relevant state variables, the time it takes to solve the model remains almost unchanged. This

[4] Even in other solutions when g_t is i.i.d. process , b_t is not. This is consistent with the results found in AMSS (2002)

is, once again, because the ANN can use all the state variables at once even if they are highly correlated. In contrast, the regression approach requires an increasing number of iterations to find the best set of orthogonal elements of the state variables. In this sense this solution method is more scalable, and we can use it to solve more complicated models with multiple maturities as showed in the next section. Second, we discovered that the ANN approach tends to be more robust to different specifications of the g_t process - which is probably due to its non-parametric nature which can handle jumps and non-linearities induced by variation in the exogenous process.

2 General case: model with N bonds and Epstein-Zin preferences

2.1 Adding Epstein-Zin preferences

Matching the relevant moments in the bonds data can be difficult with the model just considered. Usual practice is to introduce Epstein-Zin preferences to separate risk-aversion and elasticity of intertemporal substitution parameters.

In this model the representative household has preferences:

$$V_t = [(1-\beta)U(c_t, l_t)^{1-\rho} + \beta(\mathbb{E}_t V_{t+1}^{1-\gamma})^{\frac{1-\rho}{1-\gamma}}]^{\frac{1}{1-\rho}}$$

where $l_t = 1 - h_t$. Subject to a budget constraint:

$$c_t + p_t b_{t+1} = b_t + (1-\tau_t)h_t$$

A one-period bond price p_t is the expected value of the stochastic discount factor (SDF):

$$p_t = \beta \mathbb{E}_t \mathcal{M}_t(V_{t+1}) \left(\frac{U_{t+1}}{U_t}\right)^{-\rho} \frac{U_{c,t+1}}{U_{c,t}}$$

where $\mathcal{M}_t(V_{t+1}) = \left(\frac{V_{t+1}}{\mathcal{R}_t(V_{t+1})}\right)^{\rho-\gamma}$ (see Appendix B for more details).

2.2 Implementability and measurability constraints

Furthermore, we extend the model to include a generic number of bonds N to study the optimal maturity structure. In this setting, the government can decide to issue non-state contingent securities b_t^i with maturity i and we assume full buy-back.

The government budget constraint is:

$$\sum_{i=1}^{N} p_{i-1,t} b_t^i = \tau_t h_t - g_t + \sum_{i=1}^{N} p_{i,t} b_{t+1}^i$$

Combining the technology constraint, $c_t + g_t = h_t$, with the household's labor optimality condition, $1 - \tau_t = U_{l,t}/U_{c,t}$, yields an expression for surplus:

$$s_t = \tau_t h_t - g_t = c_t - (1 - \tau_t) h_t = c_t - \frac{U_{l,t}}{U_{c,t}}(c_t + g_t)$$

The government problem is essentially identical to before, except that now bond prices are discounted with a SDF that contains the ratio of the agent's continuation value and its certainty equivalent:

$$\sum_{i=1}^{N} b_t^i \mathbb{E}_t \beta^{i-1} \mathcal{M}_t(V_{t+i-1}) \left(\frac{U_{t+i-1}}{U_t}\right)^{-\rho} \frac{U_{c,t+i-1}}{U_{c,t}} = s_t + \sum_{i=1}^{N} b_{t+1}^i \mathbb{E}_t \beta^i \mathcal{M}_t(V_{t+i}) \left(\frac{U_{t+i}}{U_t}\right)^{-\rho} \frac{U_{c,t+i}}{U_{c,t}}$$

The computational complexity to solve this problem increases significantly; besides that all the lagged values of b_t^i, up to its maturity, become relevant state variables, additional state variables are required to keep track of the recursive utility constraint in the household problem. Epstein-Zin preferences does not just complicate the problem introducing more state variables. An additional layer of complexity comes from the non-convexities in the implementability constraint, as mentioned in (Karantounias, 2017). First order conditions methods, in this context, might lead to a wrong solution. In order to address this issue, we solve the system of first order conditions starting from many different initial points and evaluating welfare at each corresponding solution.

2.2.1 Sequential formulation of the Ramsey problem

The primal approach to the Ramsey problem yields the following recursive Lagrangian:

$$\max_{\{c_t, b_{t+1}^i, \mu_t, V_t\}_{t=0}^{\infty}} \mathcal{L} = V_0 + \mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t \Bigg\{ \mu_t \Big(U_t^{-\rho} U_{c,t} s_t + \sum_{i=1}^{N} \mathbb{E}_t \beta^i b_{t+1}^i \mathcal{M}_t(V_{t+i}) U_{t+i}^{-\rho} U_{c,t+i} - \sum_{i=1}^{N} \mathbb{E}_t \beta^{i-1} b_t^i U_{t+i-1}^{-\rho} U_{c,t+i-1} \mathcal{M}_t(V_{t+i-1}) \Big) + \sum_{i=1}^{N} \xi_{U,t}^i (B^U - b_{t+1}^i) + \sum_{i=1}^{N} \xi_{L,t}^i (b_{t+1}^i - B^L) \Bigg\}$$

Subject to:

$$V_t = [(1-\beta)U(c_t, 1-c_t-g_t)^{1-\rho} + \beta(\mathbb{E}_t V_{t+1}^{1-\gamma})^{\frac{1-\rho}{1-\gamma}}]^{\frac{1}{1-\rho}}$$

The first order condition with respect to b^i_{t+1}, yields the following intertemporal expression for the promise keeping Lagrange multiplier μ:

$$\mu_t = \left[\mathbb{E}_t \mathcal{M}_t(V_{t+i}) U_{t+i}^{-\rho} U_{c,t+i}\right]^{-1} \left[\mathbb{E}_t \mu_{t+1} \mathcal{M}_{t+1}(V_{t+i}) U_{t+i}^{-\rho} U_{c,t+i} - \frac{\xi_t^U}{\beta^i} + \frac{\xi_t^L}{\beta^i}\right]$$

In order to calculate the first order condition with respect to c_t, it is necessary to calculate an expression for the derivative of welfare V_0 with respect to c_t. Note that V_0 contains all the consumption path from 0 throughout ∞. In Appendix B we explain how to find an expression for $\frac{\partial V_0}{\partial c_t(g^t)}$ [5], with which is possible to find the following optimality condition:

$$V_0^\rho (1-\beta)\mathcal{X}_{0,t} U_t^{-\rho} \frac{\partial U_t}{\partial c_t(g^t)} + \mu_t \left(\frac{\partial U_t^{-\rho} U_{c,t}}{\partial c_t(g^t)} s_t + \frac{\partial s_t}{\partial c_t} U_t^{-\rho} U_{c,t} \right) +$$
$$\frac{\partial U_t^{-\rho} U_{c,t}}{\partial c_t(g^t)} \sum_{i=1}^N \left(\mu_{t-i}\mathcal{M}_{t-i}(V_t) - \mu_{t-i+1}\mathcal{M}_{t-i+1}(V_t)\right) b^i_{t-i+1} + \lambda_t^V V_t^{-\rho}(1-\beta) U_t^{-\rho} \frac{\partial U_t}{c_t(g^t)} = 0$$

where λ_t^V is the time-t Lagrange multiplier associated with the recursive constraint and $\mathcal{X}_{t_1,t_2} \equiv \prod_{k=1}^{t_2-t_1} \mathcal{M}_{t_1+k-1}(V_{t_1+k})$ with $\mathcal{X}_{t_1,t_2} \equiv 1, \forall t_2 \leq t_1$. Note that $\mathcal{X}$ admits a recursive representation[6].

The first order condition with respect to V_t (see Appendix B) yields the following recursion for λ_t^V [7]:

$$\lambda_t^V = \sum_{i=1}^N \left(\mu_{t-i} \frac{\partial \mathcal{M}_{t-i}(V_t)}{\partial V_t(g^t)} - \mu_{t-i+1} \frac{\partial \mathcal{M}_{t-i+1}(V_t)}{\partial V_t(g^t)} \right) b^i_{t-i+1} U_{c,t} U_t^{-\rho} + \lambda_{t-1}^V \left(\frac{V_{t-1}}{V_t}\right)^\rho \mathcal{M}_{t-1}(V_t)$$

The first order condition with respect to μ_t just gives back the inter-temporal government budget constraint.

CRRA preferences When $\rho = \gamma$ Epstein-Zin preferences collapses into the CRRA case. It is easy to verify that the above otpimality conditions collapse to the following set of

[5] $\frac{\partial V_0}{\partial c_t(g^t)} = V_0^\rho \beta^t (1-\beta)\mathcal{X}_{0,t}\pi(g^t|g^0) U_t^{-\rho} \frac{\partial U_t}{\partial c_t(g^t)}$

[6] $\mathcal{X}_{t_1,t_2} \equiv \prod_{k=1}^{t_2-t_1} \mathcal{M}_{t_1+k-1}(V_{t_1+k}) = \mathcal{M}_{t_2-1}(V_{t_2}) \prod_{k=1}^{t_2-t_1-1} \mathcal{M}_{t_1+k-1}(V_{t_1+k}) = \mathcal{M}_{t_2-1}(V_{t_2})\mathcal{X}_{t_1,t_2-1}$

[7] Where (see Appendix B): $\frac{\partial \mathcal{M}_{t-i}(V_t)}{\partial V_t} = (\rho-\gamma)\frac{\mathcal{M}_{t-i}(V_t)}{V_t}\left[1 - \mathcal{M}_{t-i}(V_t)^{\frac{1-\gamma}{\rho-\gamma}} \pi(g_t|g^{t-i})\right]$

equations:

$$
\begin{aligned}
&c_t : u_{c,t} - v_{l,t} + \lambda_t \big[u_{c,t} - v_{l,t} + u_{cc,t} c + v_{ll,t}(c_t + g_t) \big] + \sum_{i=1}^{N} (\lambda_{t-i} - \lambda_{t-i+1}) b^i_{t-i} u_{cc,t} = 0 \\
&b^i_{t+1} : [\mathbb{E}_t u_{c,t+i}]^{-1} \left[\mathbb{E}_t \mu_{t+1} u_{c,t+i} - \frac{\xi^i_{U,t}}{\beta^i} + \frac{\xi^i_{L,t}}{\beta^i} \right] \\
&\lambda_t : \sum_{i=1}^{N} b^i_t \mathbb{E}_t \beta^{i-1} \frac{u_{c,t+i-1}}{u_{c,t}} = s_t + \sum_{i=1}^{N} b^i_{t+1} \mathbb{E}_t \beta^i \frac{u_{c,t+i}}{u_{c,t}}
\end{aligned}
$$

3 Computational strategy

For illustrative purposes we describe in detail the computational strategy when $\rho = \gamma$. In Appendix C, we describe a generalization of this strategy for the EZ case.

At every instant t the information set is $\mathcal{I}_t = \{g_t, \{\{b^i_{t-k}\}_{k=0}^{N-1}\}_{i=1}^{N}, \{\lambda_{t-k}\}_{k=1}^{N}\}$. Consider projections of $\mathbb{E}_t u_{c,t+i}$, $\mathbb{E}_t \lambda_{t+i} u_{c,t+i}$ and $\mathbb{E}_t u_{c,t+i-1}$ onto $\mathcal{I}_t$. We model these relationships using one single-layer artificial neural network $\mathcal{ANN}(\mathcal{I}_t)$ with the characteristics described in Table 1. For example, in the two bonds case there are six[8] terms to project and, if one bond matures in 1 period, they reduce to five[9]. In particular, if the long maturity is $N > 1$ then the terms to approximate are the following:

$$
\begin{aligned}
\mathcal{ANN}^i_1 &= \mathbb{E}_t u_{c,t+i} \quad \text{for } i = \{1, N\} \\
\mathcal{ANN}^i_2 &= \mathbb{E}_t \lambda_{t+i} u_{c,t+i} \quad \text{for } i = \{1, N\} \\
\mathcal{ANN}^i_3 &= \mathbb{E}_t u_{c,t+i-1} \quad \text{for } i = \{N\}
\end{aligned}
$$

The solution procedure is summarized by the following algorithm:
Given starting values $\lambda_{t-1} = 0$ and initial weights for $\mathcal{ANN}$ [10], simulate a sequence of $\{c_t\}$, $\{\lambda_t\}$ and $\{b^i_t\}$ as follows:

1. Impose the Maliar moving bounds on debt (these bounds are particularly important and need to be tight and open slowly since the ANN at the beginning can only make accurate predictions around zero debt - that is our initialization point). Proper penalty

[8] $\mathbb{E}_t(u_{c,t+N})$, $\mathbb{E}_t(u_{c,t+N-1})$, $\mathbb{E}_t(u_{c,t+N-1}\lambda_{t+1})$, $\mathbb{E}_t(u_{c,t+S})$, $\mathbb{E}_t(u_{c,t+S-1})$, $\mathbb{E}_t(u_{c,t+S}\lambda_{t+1})$

[9] $\mathbb{E}_t(u_{c,t+S-1})$ is just $u_{c,t}$

[10] The network can be initially trained imposing $\{b_t\} = 0$

functions are used instead of the ξ terms to avoid out of bound solutions, see Faraglia et al. (2014b) for more details. Use forward-states on the following i equations:

$$\forall i: \quad \lambda_t = \mathcal{ANN}_1^i(\mathcal{I}_t)^{-1}\left[\mathcal{ANN}_2^i(\mathcal{I}_t) - \frac{\xi_{U,t}^i}{\beta^i} + \frac{\xi_{L,t}^i}{\beta^i}\right]$$

Note that λ_t is now over identified. We tackle this problem by using the Forward-States approach as described in Faraglia et al. (2014b). This involves approximating the expected value terms with the state variables that are relevant at period $t+1$ and invoking the law of iterated expectations[11].

The equations to solve are:

$$\forall i: \quad \lambda_t = \left[\mathbb{E}_t\mathcal{ANN}_1^i(\mathcal{I}_{t+1})\right]^{-1}\left[\mathbb{E}_t\mathcal{ANN}_2^i(\mathcal{I}_{t+1}) - \frac{\xi_{U,t}^i}{\beta^i} + \frac{\xi_{L,t}^i}{\beta^i}\right]$$

2. Choose T big enough and find $\{c_t\}$ and $\{b_{t+1}^i\}$ that solve the following system of $2T$ equations:

   i. $$u_{c,t} - v_{l,t} + \lambda_t\left[u_{c,t} - v_{l,t} + u_{cc,t}c + v_{ll,t}(c_t + g_t)\right] + \sum_{i=1}^{N}(\lambda_{t-i} - \lambda_{t-i+1})b_{t-i}^i u_{cc,t} = 0$$

   ii. $$\sum_{i=1}^{N} b_t^i \beta^{i-1}\mathcal{ANN}_3^i(\mathcal{I}_t) = U_t U_{c,t} s_t + \sum_{i=1}^{N} b_{t+1}^i \beta^i \mathcal{ANN}_1^i(\mathcal{I}_t)$$

3. If the solution error is large, or a reliable solution could not be found, the algorithm automatically restores the previous period ANN and tries to proceed with a reduced Maliar bound [12].

4. If the solution calculated shrinking the bound at iteration $i-1$ is not satisfactory, the algorithm does not go back another iteration but uses the same ANN and tries to lower the $Bound_{i-1}$ again towards $Bound_{i-2}$. Once a reliable solution is found, the algorithm proceeds to calculate the solution for iteration i again, but with $Bound_i = Bound_{i-1} + (Bound_{i-1} - Bound_{i-2})$. In this way, if an error is detected multiple times

[11] For a detailed description of the procedure using polynomial regressions see Faraglia et al. (2014a) or Faraglia et al. (2014b). Here we follow the same logic using the neural network

[12] If the unreliable solution has been detected in iteration i the algorithm restore the $i-1$ environment and tries to proceed with $Bound_{i-1} = \alpha * Bound_{i-1} + (1-\alpha) * Bound_{i-2}$

we guarantee that both $Bound_i$ and $Bound_{i-1}$ keep shrinking toward $Bound_{i-2}$ and there must exist a point close enough to $Bound_{i-2}$ such that the system can be reliably solved with both $Bound_{i-1}$ and $Bound_i$.

5. If the solution found at iteration i is satisfactory, the ANN enters the learning phase supervised by the implied model dynamics, the Maliar bounds are increased and a new iteration starts again.

Keep repeating until the ANN prediction errors converge below a certain small threshold and the simulated sequences of $\{b_t^i\}$, and c_t do not change[13].

3.1 Two bonds: CRRA and EZ

In table 3 we compare computation times required to solve the two bonds problem when the expectations are parametrized using an ANN against the Condensed PEA method. In particular, we solve the model with an AR(1) process for g_t with persistence parameter of 0.8 and with a constant such that the mean of g_t is 0.2. We run the code under the same computational environment and parametrization, the only difference being the way it approximates the expectation and we solve the model for 10 different realizations of g_t using both methods. The numbers in table 3 are the means over these simulations.

[13]There is no need to check λ_t which can be backed out analytically from the first order condition for c_t

	ANN		C. PEA	
Projected term	*Residual*	$Residual_{\%}$	*Residual*	$Residual_{\%}$
$\mathbb{E}_t(u_{c,t+N}\lambda_{t+1})$	0.017	1.1%	0.009	0.5%
$\mathbb{E}_t(u_{c,t+N})$	0.056	0.8%	0.035	0.5%
$\mathbb{E}_t(u_{c,t+N-1})$	0.049	0.7%	0.035	0.5%
$\mathbb{E}_t(u_{c,t+S})$	0.047	0.7%	0.009	0.5%
$\mathbb{E}_t(u_{c,t+S}\lambda_{t+1})$	0.019	1.2%	0.037	0.5%
Time	2263s		17474s	

Table 3: Bonds bounds are 100% the GDP. Mean absolute deviation between the debt path calculated through C.PEA and ANN is 0.199 for long bond and 0.147 for short bond. $Residual \equiv \frac{1}{T}\sum\left|Y_i - \hat{E}_i\right|$ and $Residual_{\%} \equiv \frac{1}{T}\sum\left|\frac{Y_i-\hat{E}_i}{Y_i}\right|$

Compared to one bond case now there are even more possible parameter combinations the condensed PEA needs to explore. In summary, methods converged to roughly the same solutions, but Condensed PEA took on average 17474 seconds, whereas ANN took 2263 seconds.

For illustration purposes we solve the model with both CRRA and Epstein-Zin preferences using the same process for government expenditure. Bottom panel of Figure 2 shows the simulated path of the model with CRRA preferences. The two bonds move in opposite directions: it is optimal for the government to borrow money using the long-term bond and lending money using the short-term bond as in Angeletos (2002). Moreover, dynamics of the two bonds are highly negatively correlated and, like in (Buerra and Nicolini, 2004) the positions are large and volatile.

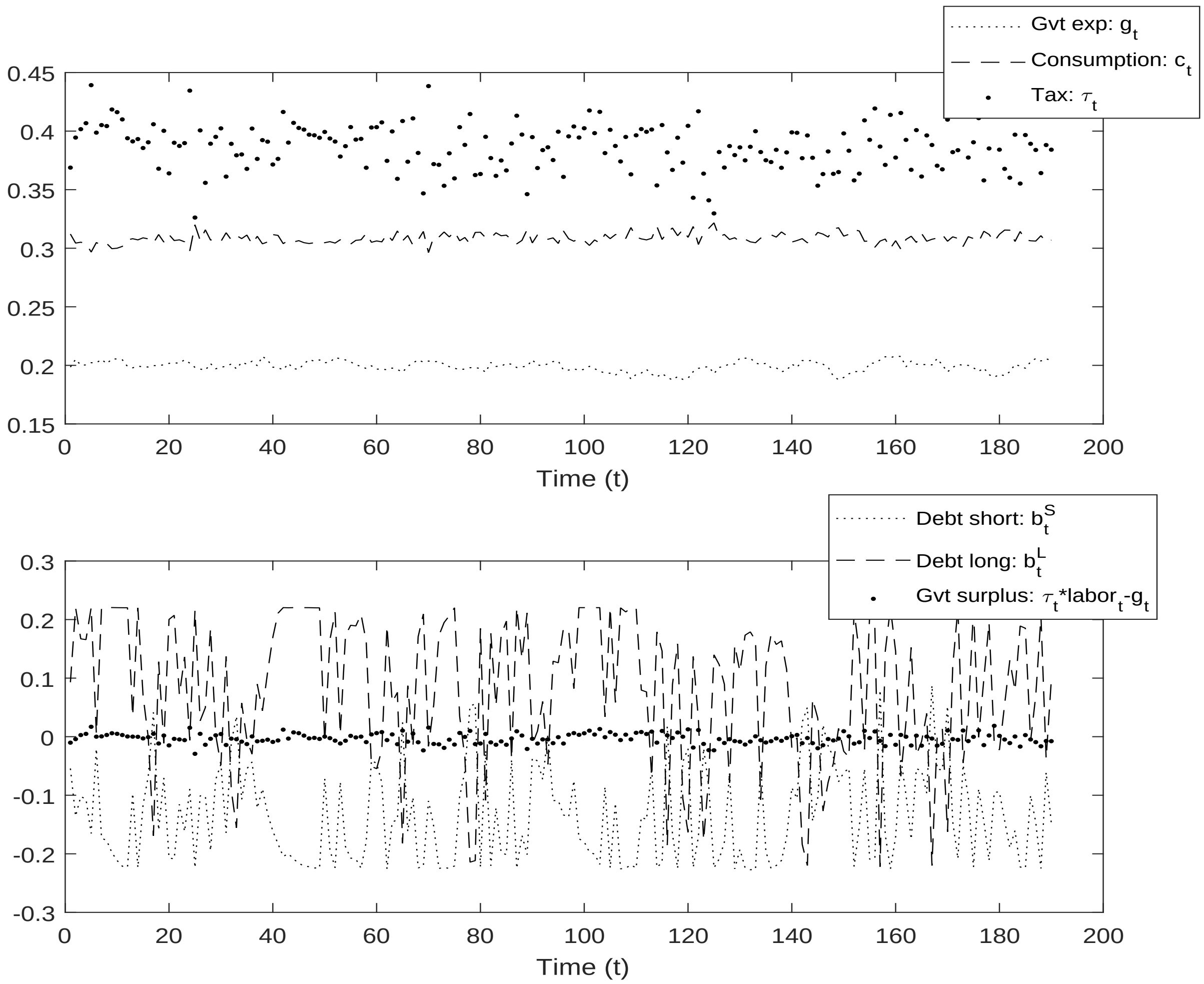

Figure 2: CRRA: Solution to the 2 bonds case. g is an exogenous process generated as an AR(1).

* When the preferences are Epstein-Zin, the optimal debt portfolio is no longer that volatile. Anmol Bhandari and Sargent (2017) took an important first step in this direction and use

a perturbation method around current level of government debt to solve and calibrate the model with two bonds. Their findings suggest that, after calibrating the model to match the data on bond returns, the optimal portfolio does not hold any short position and the allocation shares are around equal among different maturities. Our results using a global solution method is consistent with this idea. As shown in Figure 3, for exactly the same process of g_t and keeping the remaining parameters fixed, now bond holdings are both positive and small so that little re-balancing are required when an aggregate shock hits.

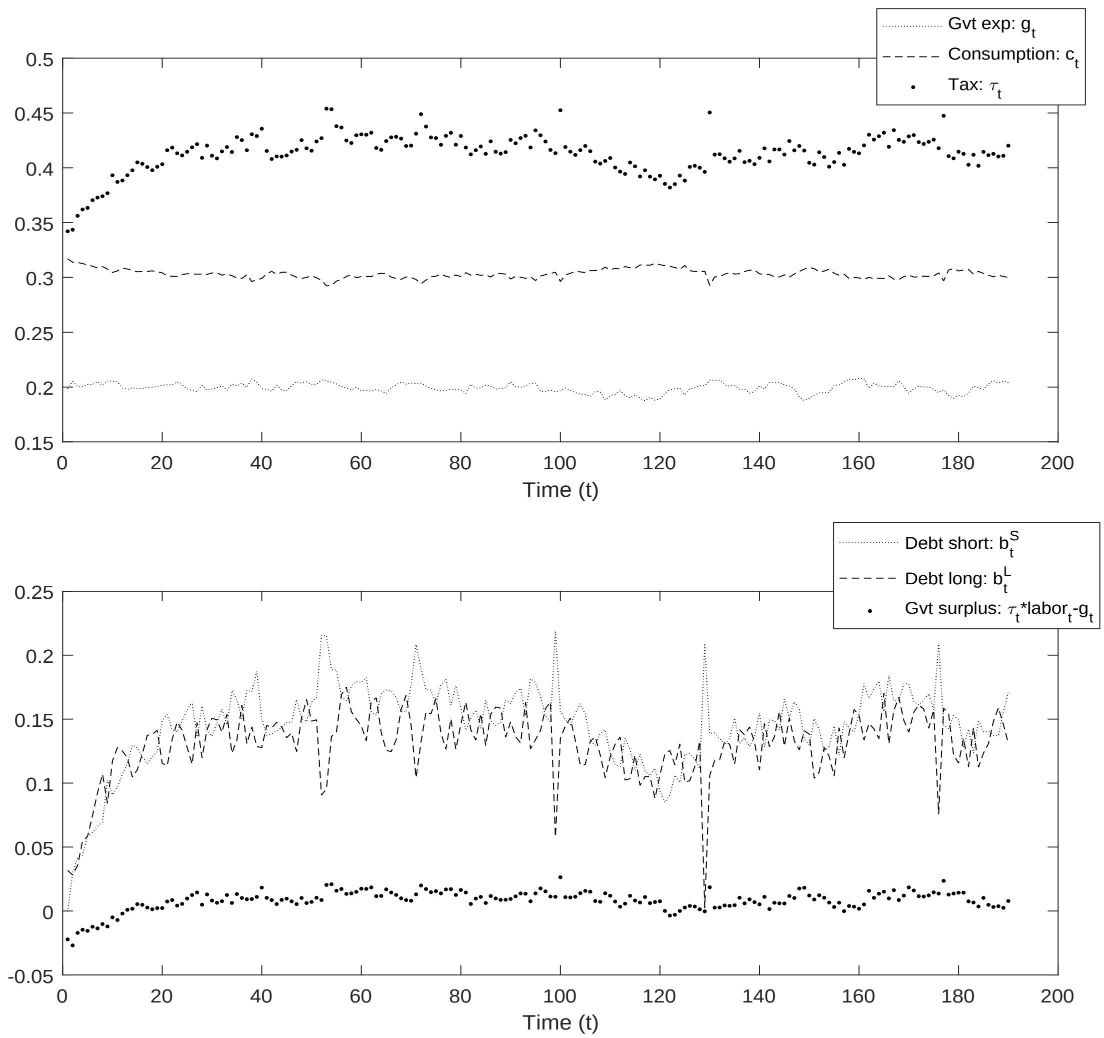

Figure 3: EZ: Solution to the 2 bonds case. g is an exogenous process generated as an AR(1).

There might be multiple interesting ways to match the data, for example Anmol Bhandari and Sargent (2017) use shocks on discount factor, labor efficiency and government expenses.

We believe that matching financial data in other ways could lead to different optimal debt dynamics. Our efficient computational method would be fast enough to run simulated method of moments.

3.2 The three bonds solution

We further explore the capabilities of the ANN by adding a medium term b_t^M bond to the model with CRRA preferences. First thing that changes compared to the 2 bond case is the correlation between different maturities. Solution to the two bonds case was showing almost perfectly negatively correlated maturities (correlation was -0.9935), whereas now different maturities are less negatively correlated as shown in the table 4. In particular, in the short-run the government positions in the three bonds are negatively correlated with each others, all around -0.5.

	$\rho(b^S, b^M)$	$\rho(b^S, b^L)$	$\rho(b^M, b^L)$
Short-run	-0.49	-0.52	-0.46
Long-run	-0.43	-0.18	-0.16

Table 4: Correlations (ρ) between the short (S), medium (M) and long (L) term securities.

In figure 4 we report the short-run (first 50 periods) dynamics we obtained solving the three bonds model. The long-run (1000 periods) solution is reported in Appendix A in figure 8. In the short run the debt positions are volatile, but not as much as in the case of two debt instruments. In the long-run the government tends to accumulate assets and slowly reduces taxes (from around 39% to 34%). As a consequence the short-term bond, used by the government to lend money reach the lower bound[14] and the government starts to use long-term bonds to compensate the fluctuations in government expenditure. As a result, position in the long-term bonds becomes even less negatively correlated with the holdings in the short and medium bonds, as seen in the bottom row of table 4

[14]both upper and lower bounds are set at 100% the GDP

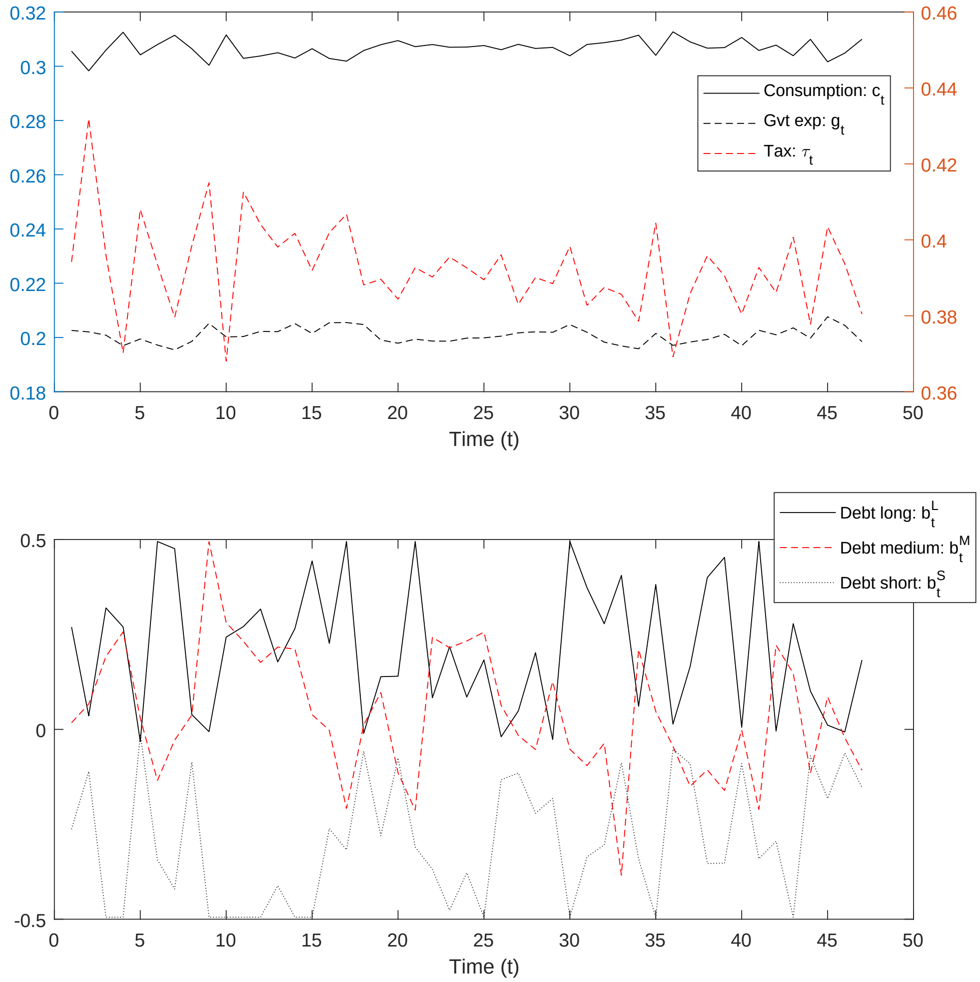

Figure 4: Short-term dynamic of the 3 bonds problem. Bounds are set at 100% the GDP and g follows an AR(1) process.

Lastly in figure 5 we present the short run predictions of the ANN. As can be seen from the figure, projections residuals are the smallest for the expectations involving the recursive multiplier λ_t. This result is reasonable as their values depend on the past debt, whereas consumption is largely unpredictable. The increase in the computation time going from 2 to 3 bonds is lower than going from 1 to 2 bonds. The reason is that the ANN method for the 2 and 3 bonds cases is qualitatively the same: it just has to consider more inputs and two more outputs. In table 5 we report the ANN prediction errors, which are very small.

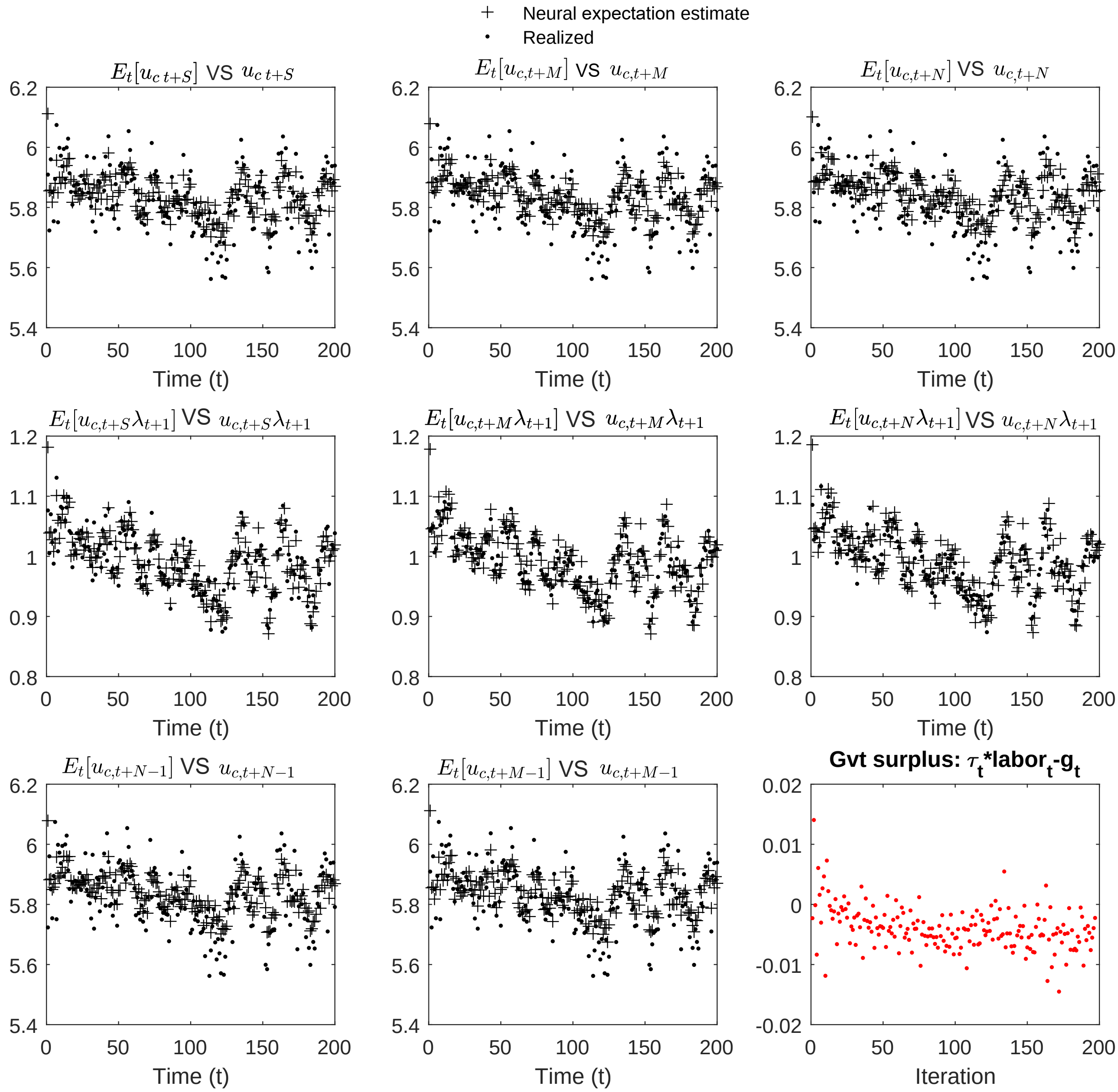

Figure 5: First 200 periods Neural network predictions of the expected value terms versus the realized sequences for the three bonds problem. Bounds are set at 100% the GDP. g follows an AR(1) process.

	ANN							
Projected term	$E_t[u_{c,t+S}]$	$E_t[u_{c,t+M}]$	$E_t[u_{c,t+N}]$	$E_t[u_{c,t+S}\lambda_{t+1}]$	$E_t[u_{c,t+M}\lambda_{t+1}]$	$E_t[u_{c,t+N}\lambda_{t+1}]$	$E_t[u_{c,t+N-1}]$	$E_t[u_{c,t+M-1}]$
Residual	0.048	0.069	0.081	0.013	0.015	0.018	0.069	0.048
$Residual_{\%}$	0.9%	1.2%	1.4%	0.2%	0.3%	0.3%	1.2%	0.9%
Time	1945s							

Table 5: $Residual = \frac{1}{T}\sum\left|Y_i - \hat{E}_i\right|$ and $Residual_{\%} = \frac{1}{T}\sum\left|\frac{Y_i-\hat{E}_i}{Y_i}\right|$

4 Potential Other Applications

4.1 Models with Financial Intermediation and credit constraints

An area where machine learning could be promising is in solving models that feature kinks or high non-linearities and, at the same time, have large state space. Such models are usually solved with modifications of value function iteration, which might need many grid points to properly capture kinks in the value function. At the same time, these methods are prone to the curse of dimensionality. These features are shared by many models including financial intermediation and possibility of government default. Consider a model by Bocola (2016) as a recent example, which features risk neutral banks that intermediate between representative consumers and firms. The value function of a representative bank is:

$$v^b(n, \mathbf{S}) = \max_{a_b, a_k, b'} \mathbb{E}_s \Lambda(\mathbf{S'}, \mathbf{S}) \left\{ (1-\psi)n' + \psi v^b(n', \mathbf{S}) \right\}$$

where n is banks net worth and exit is stochastic with probability $1-\psi$. a_b, a_k and b' are respectively the holdings of government bonds, firm equity and borrowing from households. $\mathbf{S}$ is an aggregate state vector. Nonlinearities in the model come from the following constraints of the bank:

$$\sum_{j=\{B,K\}} Q_j(\mathbf{S}) a_j \leq n + \frac{b'}{\mathbf{R}(S)}$$

$$n' = \sum_{j=\{B,K\}} R_j(\mathbf{S'}, \mathbf{S}) Q_j(\mathbf{S}) a_j - b'$$

The first constraint says that bank assets cannot exceed bank liabilities. The second one describes the law of motion for the net worth, which depends on the payoff on banks assets at the net of households payoff. In this model government can default according to a reduced form default process d_t with time varying probability $p^d(\mathbf{S})$:

$$d' = \begin{cases} 1 & \text{if } \epsilon'_d - \Psi(\mathbf{S}; \theta_2) \\ 0 & \text{otherwise,} \end{cases}$$

So that:

$$p^d(\mathbf{S}) \equiv Prob(d' = 1|\mathbf{S}) = \frac{exp(\Psi(\mathbf{S}; \theta_2))}{1 + exp(\Psi(\mathbf{S}; \theta_2))}$$

Government default triggers decline in the bank's next period net worth and, depending on banks holdings of government debt, this can drastically reduce the bank's ability to finance firms. The model is written to study the effects of both default and its time varying probability - the default risk. For this purpose it needs to be solved using global methods. Further complications arise from the fact that the aggregate state vector **S** is seven dimensional. In a more recent work Bocola and Lorenzoni (2017) present an open economy model with financial intermediation. Government has a choice to issue bonds denominated either in home or in foreign currency and banks borrow in both kinds of bonds against their capital. Authors show that in the constrained region bank's demand for capital can become upward sloping and this could immediately move the economy into a new equilibrium.

Another possible application is a model by Bianchi (2016), used to study the effects of government interventions in the credit market. The main non linearity comes from the firms credit and equity constraints:

$$\frac{b_{t+1}}{R} + \theta F(z,k,h) \leq \kappa_t k_{t+1}$$
$$d \geq \bar{d}$$

where κ_t is a stochastic shock on the credit constraint. In the stochastic steady state these constraints bind with positive probability, which is increasing in the amount of debt held by the firms. These occasionally binding constraints create kinks in the policy function and lead to a different behavior of the economy depending on where it is in the state space. We expect that models with financial intermediation and occasionally binding constraints should become even richer in terms of their dynamics as well as in the number of state variables. For instance, while the seminal paper by Arellano (2008) had two state variables, recent models of government default have 7 (Bocola, 2016). We believe that machine learning techniques are well suited to capture such nonlinearities globally.

4.2 Asset pricing and learning

Klaus et al. (2017) incorporate price subjective belief into a standard asset pricing model. Agents behave optimally given their imperfect knowledge of the price dynamics. The authors show that this feature of the model leads to two desirable facts: the model can replicate the volatility of the stock prices observed in the data and the subjective capital gain expectations are positively correlated with price-dividend ratio and with expected returns.

The investment problem is the following:

$$\max_{\{C_t^i \geq 0, S_t^i\}_{t=0}^{\infty}} \mathbb{E}_0^{\mathcal{P}^i} \sum_{t=0}^{\infty} \delta^t u(C_t^i)$$

$$S_t^i P_t + C_t^i = S_{t-1}^i (P_t + D_t) + W_t$$

where C_i denotes consumption, u corresponds to the investor's utility, S_i is investor's stock holdings (bounded by an upper and a lower limits), and $P > 0$ is the (ex-dividend) price of the stock. $\mathcal{P}^i$ denotes the agents subjective probability measure, which may or may not satisfy the rational expectations hypothesis.

Exogenous processes for dividend D_t and wage W_t are given by:

$$\left(1 + \frac{W_t}{D_t}\right) = (1+\rho)^{1-p} \left(1 + \frac{W_{t-1}}{D_{t-1}}\right)^p \ln \epsilon_t^W$$

$$D_t = \beta^D D_{t-1} \epsilon_t^D$$

Agents perceive prices evolving as:

$$\ln P_{t+1} - \ln P_t = \ln \beta_{t+1} + \ln \epsilon_{t+1}$$

where ϵ and β denote a transitory and persistent ($\ln \beta_{t+1} = \ln \beta_t + \nu_t$) shocks respectively. The innovations ϵ_t and ν_t are assumed to be i.i.d. and normally distributed, the investors learn from observed prices through an optimal Bayesian filter.

The solution to this problem, agents stock demand, takes the form:

$$S_t^i = S^i \left(S_{t-1}^i, \frac{P_t}{D_t}, \frac{W_t}{D_t}, m_t^i\right)$$

where m_t^i is a sufficient statistic characterizing the subjective distributions about future relevant ratios. Finding this solution is computationally expensive, cause it requires to iterate on the FOC of the problem:

$$u'(C_t^i) = \delta \mathbb{E}_t^{\mathcal{P}^i} \left[u'(C_{t+1}^i) \frac{P_{t+1} + D_{t+1}}{P_t} \right]$$

under the subjectively perceived dividend, wage and price dynamics, where agents understand that their beliefs evolve according to their learning process. In the paper this is done using time iteration, see Aruoba et al. (2006) and Rendahl (2013), in combination with the method of endogenous grid points, see Carroll (2006). Expectations are approximated via Hermite Gaussian quadrature using three interpolation nodes for the exogenous innovations. The consumption/stockholding policy is approximated by piece-wise linear splines, which preserves the non-linearities arising in particular in the PD dimension of the state space. Maximum (relative) Euler errors achieved in the paper are on the order of 10^{-3}. The approximation method presented in this paper allows to perform precise approximation of the relevant policy functions, or expectation terms, even when they present a high degree of non linearity. We suggest our technique might lead to accurate solutions with only few neurons. Few neurons already provide a fully non parametric approach and allows to avoid the need for the spline piece-wise approximation[15] and, at the same time, a fast training phase.

Efficient and precise algorithms, in this context, are relevant since, as suggested by the authors, this setup is relatively simple and further complications of this model might be required to explore interesting questions. For example, as mentioned in the paper, in this basic setup price fluctuations do not adversely impact welfare. Authors suggest extensions such as heterogeneous investors and endogenous output which might allow to study the effects of stock price fluctuations on welfare and, therefore, questions relative to policy intervention in asset pricing.

5 Conclusion

In this paper we show how an artificial neural network can be applied efficiently to solve macroeconomics and asset pricing models. We present an interesting application in optimal fiscal policy where, due to market incompleteness, the state space increases rapidly in the number of financial instruments involved. The application of a neural network in this context not only results in a much faster solution but also proves to be more scalable, allowing to

[15] which still follows a parametric approach

study optimal policy with multiple maturities more easily. We also present an extension of the model equipped with Epstein-Zin preferences that allows to better match bond price data. In particular, we believe that an important extension is to introduce inflation and jointly calibrate the model with both nominal and real bonds data. Single-layer neural networks with a small number of neurons (as the one adopted in this paper) present efficient learning performance and, at the same time, demonstrate themselves to be capable to handle a large set of multicollinear variables. Moreover, their non-parametric nature does not require commitment to a prespecified functional form, which is particularly relevant in problems where the non linear features of the policy functions are crucial: asset pricing where investors have subjective beliefs and models where government can default. Although there is no theory of asymptotic behavior for this methodology, the method arrives at the same solution as the standard Condensed PEA technique, but in a lesser time and comparable precision.

References

S. Rao Aiyagari, Albert Marcet, Thomas J. Sargent, and Juha Sappala. Optimal taxation without state-contingent debt. *Journal of Political Economy, 110(6): 1220-1254*, 2002.

George-Marios Angeletos. Fiscal policy with noncontingent debt and the optimal maturity structure. *Quarterly Journal of Economics, 117 (3): 1105-1131*, 2002.

Mikhail Golosov Anmol Bhandari, David Evans and Thomas J. Sargent. The optimal maturity of government debt. *Working Paper*, 2017.

Cristina Arellano. Default risk and income fluctuations in emerging economies. *American Economic Review, 98:3, 690–712*, 2008.

Javier Bianchi. Efficient bailouts? *American Economic Review, 106(12): 3607–3659*, 2016.

Luigi Bocola. The pass - through of sovereign risk. *Journal of Political Economy vol. 124, no. 4*, 2016.

Luigi Bocola and Guido Lorenzoni. A model of financial crises in open economies. *Working Paper*, 2017.

Francisco Buerra and Juan Pablo Nicolini. Optimal maturity of government debt without state contingent bonds. *Journal of Monetary Economics, Volume 51, Issue 3, 531-554*, 2004.

Victor Duarte. Sectoral reallocation and endogenous risk-aversion: Solving macro-finance models with machine learning. *Working Paper*, 2018.

E. Faraglia, A. Marcet, R. Oikonomou, and A. Scott. Government debt management: the long and the short of it. *CEPR Working Paper: P 10281*, 2014a.

E. Faraglia, A. Marcet, R. Oikonomou, and A. Scott. Optimal fiscal policy problems under complete and incomplete financial markets: A numerical toolkit. *Working Paper*, 2014b.

Anastasios G. Karantounias. Optimal fiscal policy with recursive preferences. *Working Paper*, 2017.

Adam Klaus, Albert Marcet, and Johannes Beutel. Stock price booms and expected capital gains. *American Economic Review, Vol.107, No 8, 2352-2408*, 2017.

Lilia Maliar and Serguei Maliar. Parameterized expectations algorithm and the moving bounds. *Journal of Business and Economic Statistics 21/1, 88-92*, 2003.

Albert Marcet and Ramon Marimon. Recursive contracts. *Barcelona GSE Working Paper: 552*, 2011.

S. Mullainathan and J. Spiess. Machine learning: An applied econometric approach. *Journal of Economics Perspectives*, 2017.

Appendix

Appendix A

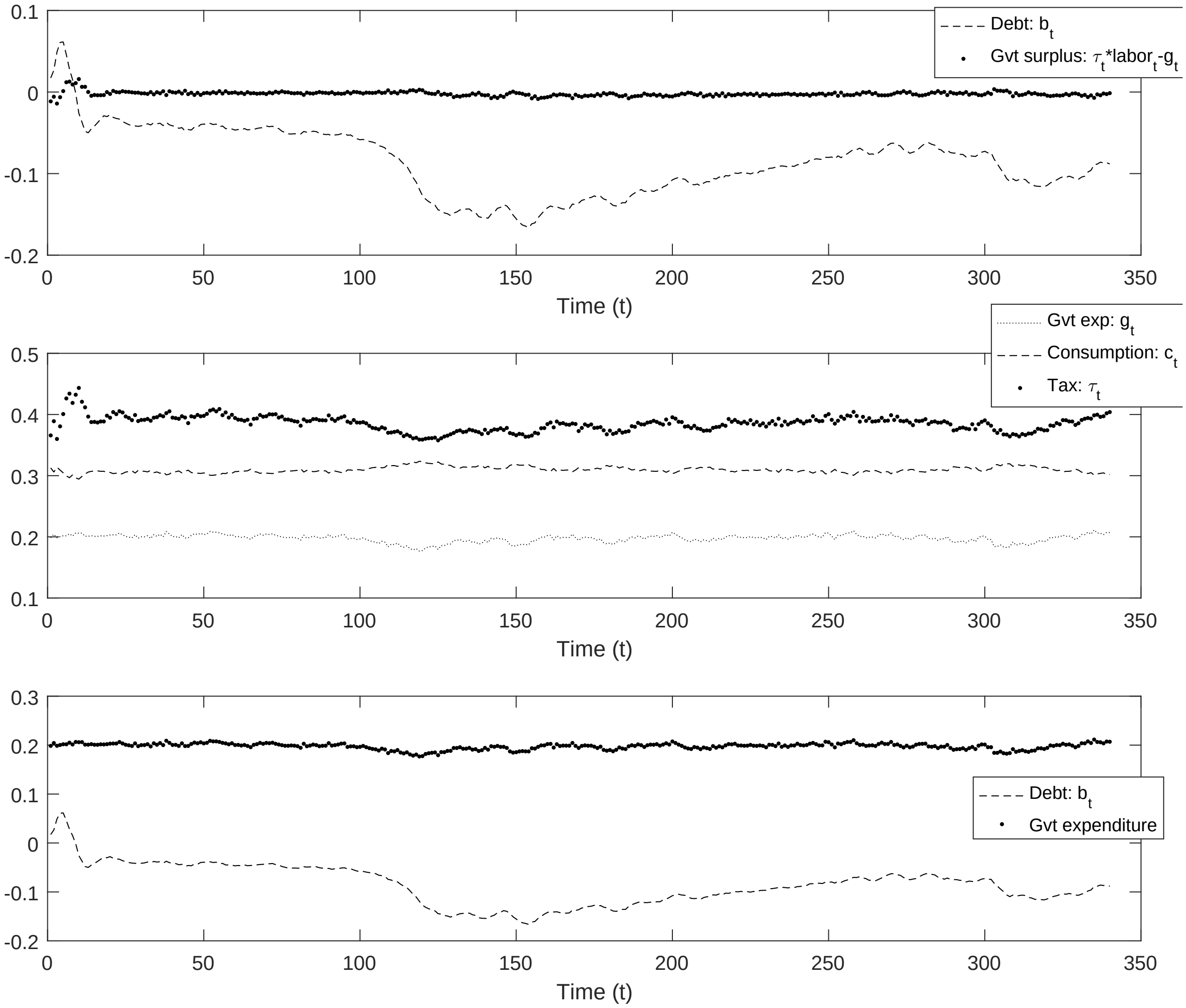

Figure 6: Solution to the 1 bond case with maturity $N = 10$, g is an exogenous AR(1) process

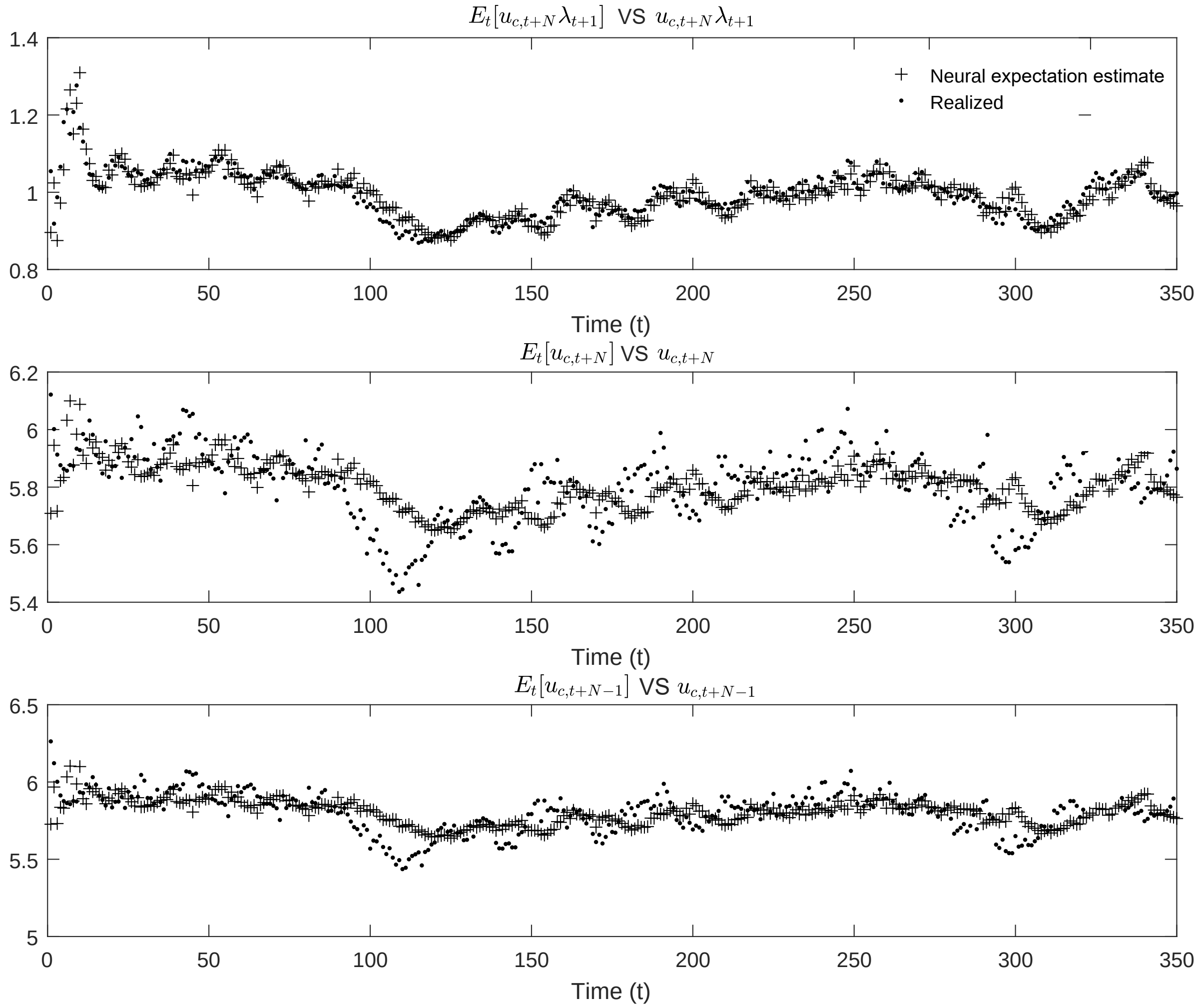

Figure 7: Neural network predictions of the expected value terms (VS realized sequences) to the 1 bond case with maturity $N = 10$, g is an exogenous AR(1) process

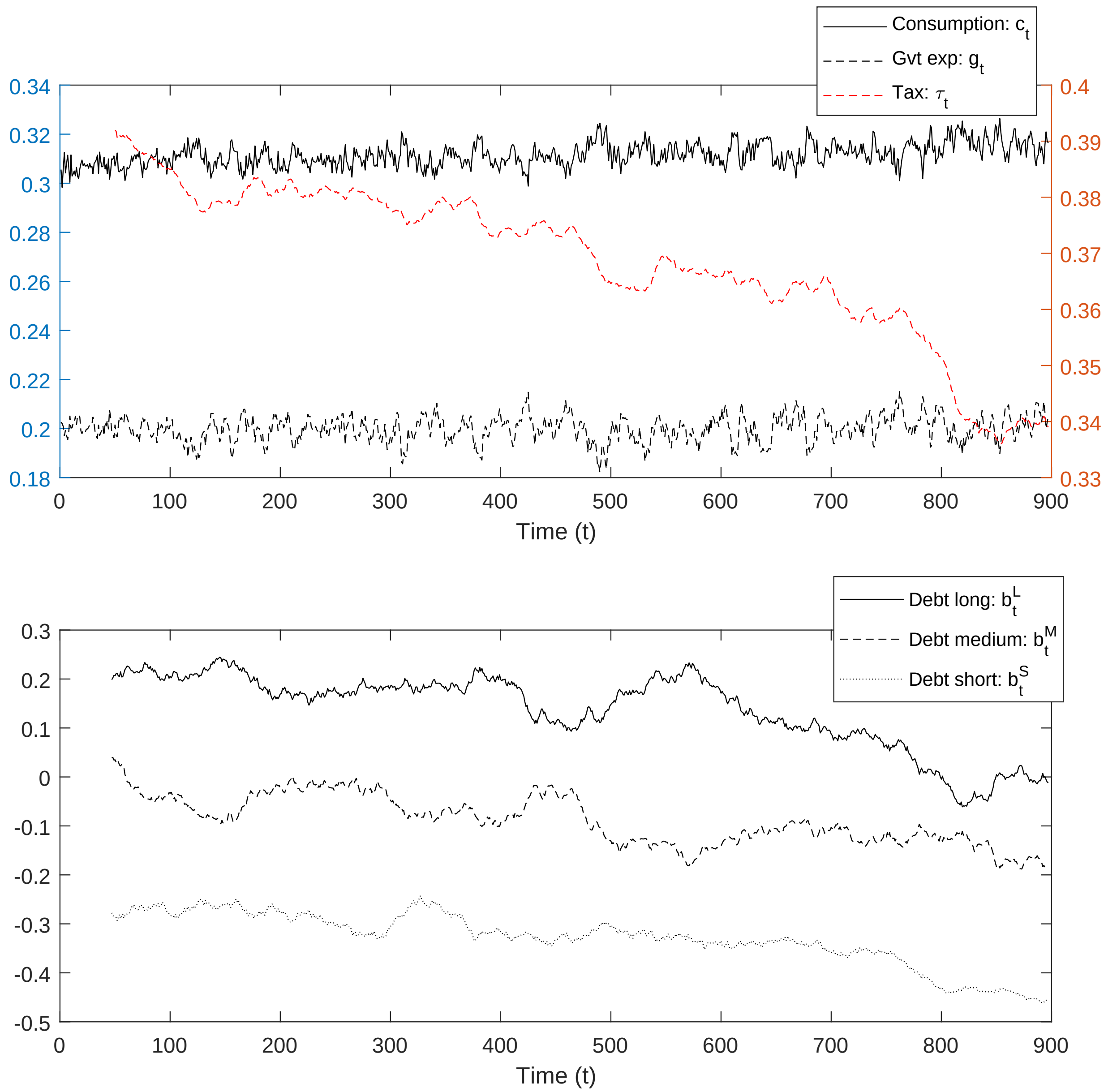

Figure 8: Solution to the 3 bonds case. g is an exogenous process generated as an AR(1). Taxes and bonds holdings are highly volatile, to make the graphs readable a 50-periods simple moving average has been applied. In the long-run taxes decrease as more government accumulates assets.

Appendix B

Model with Epstein Zin preferences

Household Preferences:

$$V_t = [(1-\beta)U(c_t, l_t)^{1-\rho} + \beta(\mathbb{E}_t V_{t+1}^{1-\gamma})^{\frac{1-\rho}{1-\gamma}}]^{\frac{1}{1-\rho}}$$

where $l_t = 1 - h_t$. Budget constraint (BC):

$$c_t + q_t b_{t+1} = b_t + (1-\tau_t)h_t$$

Defining $W_t = b_t$ and $R_{t+1} = W_{t+1}/q_t b_{t+1} = 1/q_t$:

$$c_t + q_t W_{t+1} = W_t + (1-\tau_t)h_t$$
$$\implies W_{t+1} = R_{t+1}(W_t - c_t + (1-\tau_t)h_t)$$

The HH problem can be rewritten as:

$$V_t(W_t) = \max_{c_t,h_t}[(1-\beta)U(c_t, 1-h_t)^{1-\rho} + \beta(\mathbb{E}_t V_{t+1}(W_{t+1})^{1-\gamma})^{\frac{1-\rho}{1-\gamma}}]^{\frac{1}{1-\rho}}$$
$$W_{t+1} = R_{t+1}(W_t - c_t + (1-\tau_t)h_t)$$

CE is $\mathcal{R}_t(V_{t+1}) = (\mathbb{E}_t V_{t+1}^{1-\gamma})^{\frac{1}{1-\gamma}}$.

Optimal consumption (FOC_c):

$$V_t^{\rho}\left((1-\beta)(1-\rho)U_t^{-\rho}U_{c,t} - \beta(1-\rho)(\mathbb{E}_t V_{t+1}^{1-\gamma})^{\frac{\gamma-\rho}{1-\gamma}}(\mathbb{E}_t V_{t+1}^{-\gamma}R_{t+1}V_{W,t+1}\right) = 0$$
$$\implies (1-\beta)U_t^{-\rho}U_{c,t} = \beta\mathcal{R}_t^{\gamma-\rho}\mathbb{E}_t V_{t+1}^{-\gamma}R_{t+1}V_{W,t+1}$$

Optimal labor supply (FOC_h):

$$V_t^{\rho}\left(-(1-\beta)(1-\rho)U_t^{-\rho}U_{l,t} + \beta(1-\rho)(\mathbb{E}_t V_{t+1}^{1-\gamma})^{\frac{\gamma-\rho}{1-\gamma}}(\mathbb{E}_t V_{t+1}^{-\gamma}(1-\tau_t)R_{t+1}V_{W,t+1}\right) = 0$$
$$\implies (1-\beta)U_t^{-\rho}U_{l,t} = (1-\tau_t)\beta\mathcal{R}_t^{\gamma-\rho}\mathbb{E}_t V_{t+1}^{-\gamma}R_{t+1}V_{W,t+1}$$

Envelope condition:

$$V_{W,t} = V_t^{\rho}\beta\mathcal{R}_t^{\gamma-\rho}\mathbb{E}_t V_{t+1}^{-\gamma}R_{t+1}V_{W,t+1}$$

Combine FOC_c with FOC_h to get:

$$\frac{U_{l,t}}{U_{c,t}} = 1 - \tau_t$$

Combine FOC_c with the envelope condition to get:

$$V_{W,t} = V_t^{\rho}(1-\beta)U_t^{-\rho}U_{c,t} \implies V_{W,t+1} = V_{t+1}^{\rho}(1-\beta)U_{t+1}^{-\rho}U_{c,t+1}$$

Plugging back into FOC_c:

$$(1-\beta)U_t^{-\rho}U_{c,t} = \beta\mathcal{R}_t^{\gamma-\rho}\mathbb{E}_t V_{t+1}^{-\gamma}R_{t+1}V_{t+1}^{\rho}(1-\beta)U_{t+1}^{-\rho}U_{c,t+1}$$

Rearranging and simplifying leads to the following inter-temporal Euler equation:

$$1 = \beta\mathbb{E}_t\mathcal{M}_t(V_{t+1})\left(\frac{U_{t+1}}{U_t}\right)^{-\rho}\frac{U_{c,t+1}}{U_{c,t}}R_{t+1}$$

where $\mathcal{M}_t(V_{t+1}) = \left(\frac{V_{t+1}}{\mathcal{R}_t(V_{t+1})}\right)^{\rho-\gamma}$.

We can find the the bond's price p_t as the expected value of the SDF:

$$q_t = \beta\mathbb{E}_t\mathcal{M}_t(V_{t+1})\left(\frac{U_{t+1}}{U_t}\right)^{-\rho}\frac{U_{c,t+1}}{U_{c,t}}$$

FOC_{V_t}

$$\beta^{t-i}\sum_{i=1}^{N}\pi(g^{t-i}|g^0)\beta^i\mu_{t-i}\pi(g^t|g^{t-i})b_{t-i+1}^i U_{c,t}U_t^{-\rho}\frac{\partial\mathcal{M}_{t-i}(V_t)}{\partial V_t(g^t)} -$$
$$\beta^{t-i+1}\sum_{i=1}^{N}\pi(g^{t-i+1}|g^0)\beta^{i-1}\mu_{t-i+1}\pi(g^t|g^{t-i+1})b_{t-i+1}^i U_{c,t}U_t^{-\rho}\frac{\partial\mathcal{M}_{t-i+1}(V_t)}{\partial V_t(g^t)} -$$
$$\lambda_t^V\beta^t\pi(g^t|g^0) + \beta^{t-1}\pi(g_{t-1}|g^0)\lambda_{t-1}^V\beta V_{t-1}^{\rho}\mathcal{R}_{t-1}(V_t)^{-\rho}\mathcal{M}_{t-1}(V_t)^{\frac{-\gamma}{\rho-\gamma}}\pi(g_t|g^{t-1}) = 0$$

1. $\partial V_t/\partial c_{t+j}$**.**

If $j < 0$:

$$\partial V_t/\partial c_{t+j} = 0$$

If $j = 0$:

$$\frac{\partial V_t}{\partial c_t} = (1-\beta)V_t^{\rho}U_t^{-\rho}\frac{\partial U_t}{\partial c_t}$$

If $j=1$:

$$\begin{aligned}
\frac{\partial V_t}{\partial c_{t+1}(g^{t+1})} &= V_t^{\rho}\beta\mathcal{R}_t(V_{t+1})^{\gamma-\rho}\pi(g_{t+1}|g^t)V_{t+1}^{-\gamma}\frac{\partial V_{t+1}}{\partial c_{t+1}(g^{t+1})} \\
&= V_t^{\rho}\beta\mathcal{R}_t(V_{t+1})^{\gamma-\rho}\pi(g_{t+1}|g^t)V_{t+1}^{-\gamma}\left((1-\beta)V_{t+1}^{\rho}U_{t+1}^{-\rho}\frac{\partial U_{t+1}}{\partial c_{t+1}(g^{t+1})}\right) \\
&= V_t^{\rho}\beta(1-\beta)\mathcal{M}_t(V_{t+1})\pi(g_{t+1}|g^t)U_{t+1}^{-\rho}\frac{\partial U_{t+1}}{\partial c_{t+1}(g^{t+1})}
\end{aligned}$$

If $j=2$:

$$\begin{aligned}
\frac{\partial V_t}{\partial c_{t+2}(g^{t+2})} &= V_t^{\rho}\beta\mathcal{R}_t(V_{t+1})^{\gamma-\rho}\pi(g_{t+1}|g^t)V_{t+1}^{-\gamma}\frac{\partial V_{t+1}}{\partial c_{t+2}} \\
&= V_t^{\rho}\beta\mathcal{R}_t(V_{t+1})^{\gamma-\rho}\pi(g_{t+1}|g^t)V_{t+1}^{-\gamma}\left(V_{t+1}^{\rho}\beta(1-\beta)\mathcal{R}_{t+1}(V_{t+2})^{\gamma-\rho}\pi(g_{t+2}|g^{t+1})V_{t+2}^{\rho-\gamma}U_{t+2}^{-\rho}\frac{\partial U_{t+2}}{\partial c_{t+2}}\right) \\
&= V_t^{\rho}\beta^2(1-\beta)\prod_{k=1}^{2}\mathcal{M}_{t+k-1}(V_{t+k})\prod_{k=1}^{2}\pi(g_{t+k}|g^{t+k-1})U_{t+2}^{-\rho}\frac{\partial U_{t+2}}{\partial c_{t+2}(g^{t+2})}
\end{aligned}$$

For a generic $j\geq 0$:

$$\frac{\partial V_t}{\partial c_{t+j}(g^{t+j})} = V_t^{\rho}\beta^j(1-\beta)\mathcal{X}_{t,t+j}\pi(g^{t+j}|g^t)U_{t+j}^{-\rho}\frac{\partial U_{t+j}}{\partial c_{t+j}(g^{t+j})}$$

2. $\frac{\partial \mathcal{M}_{t-1}(V_t)}{\partial V_t}$

$$\begin{aligned}
\frac{\partial \mathcal{M}_{t-1}(V_t)}{\partial V_t} =&(\rho-\gamma)\frac{\mathcal{M}_{t-1}(V_t)^{\frac{\rho-\gamma-1}{\rho-\gamma}}}{\mathcal{R}_{t-1}(V_t)^2}\left[\mathcal{R}_{t-1}(V_t) - V_t\underbrace{\mathcal{M}_{t-1}(V_t)^{\frac{-\gamma}{\rho-\gamma}}\pi(g_t|g^{t-1})}_{\frac{\partial\mathcal{R}_{t-1}(V_t)}{\partial V_t}}\right] \\
&= (\rho-\gamma)\frac{\mathcal{M}_{t-1}(V_t)^{\frac{\rho-\gamma-1}{\rho-\gamma}}}{\left(\frac{V_t}{\mathcal{M}_{t-1}(V_t)^{\frac{1}{\rho-\gamma}}}\right)^2}\left[\mathcal{R}_{t-1}(V_t) - V_t\mathcal{M}_{t-1}(V_t)^{\frac{-\gamma}{\rho-\gamma}}\pi(g_t|g^{t-1})\right] \\
&= (\rho-\gamma)\frac{\mathcal{M}_{t-1}(V_t)^{\frac{\rho-\gamma+1}{\rho-\gamma}}}{V_t^2}\left[\mathcal{M}_{t-1}(V_t)^{\frac{-1}{\rho-\gamma}}V_t - V_t\mathcal{M}_{t-1}(V_t)^{\frac{-\gamma}{\rho-\gamma}}\pi(g_t|g^{t-1})\right] = \\
&= (\rho-\gamma)\frac{\mathcal{M}_{t-1}(V_t)^{\frac{\rho-\gamma+1}{\rho-\gamma}}}{V_t}\left[\mathcal{M}_{t-1}(V_t)^{\frac{-1}{\rho-\gamma}} - \mathcal{M}_{t-1}(V_t)^{\frac{-\gamma}{\rho-\gamma}}\pi(g_t|g^{t-1})\right] = \\
&= (\rho-\gamma)\frac{\mathcal{M}_{t-1}(V_t)}{V_t}\left[1 - \mathcal{M}_{t-1}(V_t)^{\frac{1-\gamma}{\rho-\gamma}}\pi(g_t|g^{t-1})\right]
\end{aligned}$$

Appendix C

Solution algorithm for the model with Epstein - Zin preferences:
At every instant t the information set is $\mathcal{I}_t = \{g_t, \{\{b^i_{t-k}\}_{k=0}^{N-1}\}_{i=1}^N, \{\mu_{t-k}\}_{k=1}^N, \{\lambda^V_{t-k}\}_{k=1}^N\}$. Consider projections of $\mathcal{R}_{t-i}(V_t)$, $\mathbb{E}_t\mathcal{M}_t(V_{t+i})U_{t+i}^{-\rho}U_{c,t+i}$, $\mathbb{E}_t\mu_{t+i}\mathcal{M}_{t+1}(V_{t+i})U_{t+i}^{-\rho}U_{c,t+i}$ and $\mathbb{E}_t\mathcal{M}_t(V_{t+i-1})U_{t+i-1}^{-\rho}U_{c,t+i-1}$ onto $\mathcal{I}_t$. We model these relationships using one single-layer artificial neural network $\mathcal{ANN}(\mathcal{I}_t)$. For example, with two bonds[16] we would have $4N+1$ inputs and 8 outputs. In particular, use the following notations for each output:

$$\mathcal{ANN}_1^i = \mathcal{R}_{t-i}(V_t) \quad \text{for } i = \{1, N-1, N\}$$
$$\mathcal{ANN}_2^i = \mathbb{E}_t\mathcal{M}_t(V_{t+i})U_{t+i}^{-\rho}U_{c,t+i} \quad \text{for } i = \{1, N\}$$
$$\mathcal{ANN}_3^i = \mathbb{E}_t\mu_{t+i}\mathcal{M}_{t+1}(V_{t+i})U_{t+i}^{-\rho}U_{c,t+i} \quad \text{for } i = \{1, N\}$$
$$\mathcal{ANN}_4^i = \mathbb{E}_t\mathcal{M}_t(V_{t+i-1})U_{t+i-1}^{-\rho}U_{c,t+i-1} \quad \text{for } i = \{N\}$$

Given starting values $\mu_{t-1} = \lambda^V_{-1} = 0$ and initial weights for $\mathcal{ANN}$, simulate a sequence of $\{c_t\}$, $\{\lambda^V_t\}$, $\{\mu_t\}$ as follow:

1. Use forward-states on the following i equations:

$$\forall i: \quad \mu_t = \frac{\mathcal{ANN}_3^i(\mathcal{I}_t)}{\mathcal{ANN}_2^i(\mathcal{I}_t)}$$

2. Find λ^V_t, c_t and $\{b^i_{t+1}\}$ that solve the following system of equations:

$$\textbf{i.} \quad \lambda^V_t = \sum_{i=1}^N \left(\mu_{t-i}\frac{\partial\mathcal{M}_{t-i}(V_t)}{\partial V_t(g^t)} - \mu_{t-i+1}\frac{\partial\mathcal{M}_{t-i+1}(V_t)}{\partial V_t(g^t)}\right) b^i_{t-i+1}U_{c,t}U_t^{-\rho} + \lambda^V_{t-1}\left(\frac{V_{t-1}}{V_t}\right)^{\rho}\left(\frac{V_t}{\mathcal{ANN}_1^1}\right)^{\rho-\gamma}$$

$$\textbf{ii.} \quad V_0^{\rho}(1-\beta)\mathcal{X}_{0,t}U_t^{-\rho}\frac{\partial U_t}{\partial c_t(g^t)} + \mu_t\left(\frac{\partial U_t^{-\rho}U_{c,t}}{\partial c_t(g^t)}s_t + \frac{\partial s_t}{\partial c_t}U_t^{-\rho}U_{c,t}\right) +$$
$$\frac{\partial U_t^{-\rho}U_{c,t}}{\partial c_t(g^t)}\sum_{i=1}^N\left(\mu_{t-i}\left(\frac{V_t}{\mathcal{ANN}_1^i}\right)^{\rho-\gamma} - \mu_{t-i+1}\left(\frac{V_t}{\mathcal{ANN}_1^{i-1}}\right)^{\rho-\gamma}\right)b^i_{t-i+1} + \lambda^V_t V_t^{-\rho}(1-\beta)U_t^{-\rho}\frac{\partial U_t}{c_t(g^t)} = 0$$

$$\textbf{iii.} \quad \sum_{i=1}^N \beta^{i-1}b^i_t\mathcal{ANN}_4^i = s_tU_c^{-\rho}U_{c,t} + \sum_{i=1}^N\beta^i b^i_{t+1}\mathcal{ANN}_2^i$$

[16] One with maturity 1 and the other with maturity N

Where:

$$\frac{\partial \mathcal{M}_{t-i}(V_t)}{\partial V_t} = (\rho-\gamma)\frac{\left(\frac{V_t}{\mathcal{ANN}_1^i}\right)^{\rho-\gamma}}{V_t}\left[1-\left(\frac{V_t}{\mathcal{ANN}_1^i}\right)^{1-\gamma} f_{g_t}(g_t|g_{t-i})\right]$$
$$V_t = [(1-\beta)U(c_t, 1-c_t-g_t)^{1-\rho} + \beta\mathcal{ANN}_1^1(\mathcal{I}_{t+1})^{1-\rho}]^{\frac{1}{1-\rho}}$$

and

$$\frac{\partial U_t}{\partial c_t} = U_{c,t} - U_{l,t}$$

$f_{g_t}(g_t|g_{t-1})$ is the conditional probability density of the exogenous g process.

3. Use the simulated sequence to train the $\mathcal{ANN}$ and re-start from point 1 till convergence of the predicted sequence over the realized one.

Appendix D

Parameter	Value
Total endowment in a period	$A = 1$
Discount factor	$\beta = 0.96$
RRA	$\gamma = 1.5$
1/EIS	$\rho = 1.6$
Leisure utility parameter	$\eta = 1.8$
AR(1) parameter in g_t	$\phi_1 = 0.8$
constant in AR(1) process of in g_t	$c = 0.04$
Variance of the disturbances to g_t	$\sigma_\epsilon^2 = 0.00001$
Borrowing limits	$\bar{M}^N = 0.5$,$\bar{M}^S = 0.5$ $\underline{M}^N = 0.5$, $\underline{M}^S = 0.5$

Table 6: Parameter Values used in both models one and two bond models