Machine Learning Projection Methods for Macro-Finance Models

Alessandro T. Villa and Vytautas Valaitis [†]

August 6, 2019

Abstract

This paper develops a global simulation-based solution method to solve large states space macro-finance models using machine learning. We use an artificial neural network (ANN) to approximate the expectations in the optimality conditions in the spirit of the parameterized expectations algorithm (PEA). Because our method can process the entire information set at once, it is easily scalable to handle models with large state spaces that are highly collinear. We demonstrate these computational gains in two applications. First, we extend the optimal government debt problem studied by Faraglia et al. (Forthcoming) to ten maturities and we find that, when borrowing and lending constraints are tight, the optimal policy prescribes an active role for the medium-term maturities. Second, we reassess the solution of Kehoe and Perri (2002) for the international business cycle puzzles documented in Backus et al. (1992). We show that extending their two-country framework to three countries (US, Europe, China) can change the risk-sharing properties of the economy significantly.

Keywords: Machine Learning, Incomplete Markets, Projection Methods, Optimal Fiscal Policy, International Business Cycle.

JEL classification: C63, D52, E32, E37, E62, G12.

[*]We are thankful to Andrea Lanteri, Lukas Schmid and Matthias Kehrig for their encouragement. The paper received the Student Award of the Society of Computational Economics and benefited from comments by Albert Marcet, Serguei Maliar, Lilia Maliar, Swapnil Singh and seminar participants at Duke Macro Breakfast, 2018 Baltic Economic Conference, 2018 The Society for Computational Economics 24th International Conference and 2018 Econometric Society Summer European Meeting, 2019 Econometric Society African Meeting

[†]Villa: Duke University, alessandro.villa@ duke.edu, Valaitis: Duke University, vytautas.valaitis@ duke.edu.

1 Introduction

This paper introduces a new simulation-based method to solve DSGE models. In particular, we use an Artificial Neural Network (ANN) to approximate the expectation terms contained in the optimality conditions of a DSGE model, in the spirit of the Parameterized Expectations Algorithm (PEA) introduced by Den Haan and Marcet (1990). As noted in Mullainathan & Spiess (2017), applying machine learning to economics requires finding problems where prediction plays a crucial role. This suggests that the potential of machine learning in economics could extend beyond its econometrics applications, becoming a handy computational tool to solve *large state space* macro-finance models. In general, machine learning has already been used successfully as a computational tool for optimization and its applicability seems mostly relevant within the following two methods: (i) dynamic programming and (ii) projection methods. Our method belongs to this last class of algorithms and it leverages on the stochastic simulation mechanism in a similar fashion to the PEA. Simulation based methods allow to tackle large state space problem since they only visit the relevant part of the state space, but are likely to suffer from multicollinearity. The contribution of this paper is to show that an ANN, used in a similar fashion to the PEA, can endogenously extract the relevant information from a multi-collinear state space. Moreover, we show how our method overcomes the problem of dealing with a multi-colinear state space and non parametric non-linearities in the decision rules (e.g. caused by occasionally binding constraints) with two relevant economic applications.

The first application is a Ramsey taxation problem with incomplete markets. We globally solve the model with a full spectrum of ten maturities. We consider this application particularly challenging for several reasons. First, the state space explodes as a function of the length and the number of maturities. Second, this class of problems includes forward-looking constraints and the problem can be made recursive at the cost of adding even more state variables. Following Marcet & Marimon (2019), 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 increases the

state space and creates history dependence. Third, the state space includes lagged values of the same variable, which makes it highly multicollinear, requiring to extend the PEA with the so called Condensed PEA (Faraglia et al., Forthcoming). Lastly, it is known since Aiyagari et al. (2002) that the optimal level of government debt converges to a long-run limit, making it hard to solve such a problem by perturbing the system around a steady state level[1] [2]. Our method is not only faster but also more scalable than the Condensed PEA, making it an appealing choice when the model becomes increasingly complex: the length of the longest maturity and the number of debt instruments increase (with ten maturities this problem features 66 state variables). The basic intuition that a government should save with short-term bond and borrow with long-term bond, see (Angeletos, 2002), (Buerra and Nicolini, 2004) and (Faraglia et al., Forthcoming), still holds in our application with CRRA preferences and three maturities when the borrowing and lending constraints are loose. In contrast when the constraints are tight we find that the optimal policy prescribes an active role for the medium-term maturities. We find that in an environment when the government borrowing and lending constraints are tight, increasing the number of available maturities from two to ten reduces the volatility of distortionary labor taxes by 32%.

The second application is an international business cycle problem where markets are endogenously incomplete as in Kehoe & Perri (2002). Endogenous incompleteness arises from the presence of the enforcement constraints, which ensure that each country benefits by participating in the international market. The incomplete market setup in Kehoe & Perri (2002) can solve the puzzle documented in Backus et al. (1992). Despite incomplete risk-sharing between the two countries, welfare increases relative to autarky. Investigating what happens to welfare in a multi-country setup is hard using conventional computational methods. Our algorithm is particularly useful in this context, since an additional country not only enlarges the state space but can also trigger the enforcement constraints creating

[1](Bhandari et al., 2019) propose a method that allows to approximate a system around a current level of government debt.

[2](Lustig et al., 2008) on the other hand solve the optimal fiscal policy problem in incomplete markets with seven maturities up to 7 periods using a value function iteration on a sparse grid defined on the relevant subset of the state variables. In comparison to (Lustig et al., 2008) our method does not suffer from the curse of dimensionality inherent in the grid based methods

non-linear features in the decision rules that can be hardly approximated using a parametric approach. We exploit both of these features of our method to extend the model of Kehoe & Perri (2002) considering two and three countries, calibrated to US, EU and China. We find that adding a third country significantly complicates the risk sharing, resulting in a welfare loss for the US compared to the two-country case.

Literature Review Our method builds on the seminal work of Den Haan and Marcet (1990) that introduced the Parameterized Expectations Algorithm (PEA). PEA has been more recently extended (see Faraglia et al. (Forthcoming) and Faraglia et al. (2014)) to deal with multicollinearity (Condensed PEA) and over-identification (Forward-States PEA). The main contribution of our paper is to introduce an ANN-based Expectations Algorithm, allowing for machine learning to reduce the state space endogenously, resulting in greater scalability and capability to handle more complexity. In contrast, Condensed PEA achieves this result introducing an external loop that tests a subset of the state space as candidate to solve the model. Other papers that use machine learning for optimization and functional approximation include Scheidegger and Bilionis (2019), Azimovic et al. (2019), Fernández-Villaverde et al. (2019) and Duarte (2018)). Scheidegger and Bilionis (2019) use Gaussian process machine learning augmented with the active subspace method and parallelization to solve large state space problems. Azimovic et al. (2019) use deep learning to solve an overlapping generations model with convex adjustment costs and borrowing constraints. Fernández-Villaverde et al. (2019) extend the Krusell-Smith method to approximate, using an ANN, the non-linearities contained in the law of motion of the aggregate endogenous state variables. Duarte (2018) approximates the value function using an ANN and solves the Hamilton-Jacobi-Bellman equation in a similar fashion to neuro-dynamic programming.

Our method leverages on the stochastic simulation mechanism. Simulation-based methods typically allow to tackle problems with a larger state space, since they visit only its relevant part. The main challenge in this context, is represented by the problem of multicollinearity. The contribution of this paper is to show that an ANN, used in a similar fashion to the PEA, can extract the relevant information from a multi-collinear state space endogenously. In particular, when the state space is large and potentially contains lags of

the same state variables, the projection can be unstable. In the spirit of PEA, the literature tackles this problem introducing the so called Condensed PEA (Faraglia et al., Forthcoming), which requires an iterative procedure that looks for an orthogonal set of regressors and is conceptually similar to principal component extraction. The main contribution of this paper is to show that the ANN-based Expectations Algorithm can approximate these expectations digesting the entire information at once, allowing for machine learning to reduce the state space endogenously. PEA can potentially be used in combination with other standard econometric techniques that tackle the problem of multicollinearity, as in Judd et al. (2011). Similarly to our paper Judd et al. (2011) adopt the stochastic simulation approach and show how already established methods in econometrics can be used to alleviate the multicollinearity problem using a multi-country neoclassical growth model. Section 4.5 contains a detailed discussion about this method.

Our applications contribute to two strands of literature. In optimal fiscal policy, we refer to the literature on the optimal maturity structure Aiyagari et al. (2002), (Angeletos, 2002), (Buerra and Nicolini, 2004), (Faraglia et al., Forthcoming), (Bhandari et al., 2019) and (Bigio et al., 2019). Solving the Ramsey problem considered in this paper is particularly challenging. In fact, its state space explodes in function of the length of the maturities and the number of bonds. Moreover, this class of problem include forward-looking constraints and the commonly used recursive representation (Bellman equation) can not be adopted. Marcet & Marimon (2019) provide an alternative formulation to solve for the time-inconsistent optimal contract under full commitment: recursive Lagrangian or saddle-point functional equation. The solution involves adding even more state variables to the original problem. These additional state variables, necessary to recursify the problem, create history dependence. In this context, we consider an optimal debt management problem with three and ten maturities providing a methodology capable to deliver a relatively fast and scalable global solution and able to deal efficiently with the large and highly multi-collinear state space. Notable contributions to solving optimal policy problems with multiple maturities and aggregate risk have been made by (Bhandari et al., 2019) and (Bigio et al., 2019). (Bhandari et al., 2019) solve the problem by perturbating the system around a current level of government debt, whereas our methods delivers a global solution. (Bigio et al., 2019) proposes a framework

allowing to analyze an arbitrary number of maturities in a small open economy with liquidity frictions in a risky steady state. In this setting the issuance of a specific maturity depends on the spread between the international market price and the domestic one. Our paper complements the analysis by assuming a closed economy and allowing for a more general aggregate shock process. We find that the medium-term bonds are used actively when the lending and borrowing constraints are tight and bind in equilibrium. In international business cycles, we build on Backus et al. (1992) and Kehoe and Perri (2002) extending their framework to three countries. We calibrate the TFP processes using US, European and Chinese yearly data and we show that considering multiple countries is important since the addition of a new country can change the risk-sharing properties of the economy significantly. In particular, we find that with the addition of China the perfect risk-sharing contract cannot be implemented since the enforcement constraints start to bind. In this context, solving the model is particular challenging because of the larger state space and the non parametric nature of the non-linearities contained in the decision rules.

The paper is organized as follows. Section 2 illustrates the method in a baseline version of the model where government can issue only one type of non state-contingent bond. Section 3 extends the model to a generic N non-state contingent bonds with different maturities and Epstein-Zin preferences. Section 4 outlines a generic solution method for a generic N bonds model. We extend the results to the ten bonds case comparing our method with the standard approach. Section 5 presents an international business cycle model with Europe, US and China. Section 6 concludes.

2 Illustrative Model: one bond economy

2.1 Setting

The model we consider in this section is the one proposed in Aiyagari et al. (2002) and extended in Faraglia et al. (Forthcoming). It is a version of the stochastic neoclassical model with incomplete markets and a Ramsey planner[3]. The economy is populated by a

[3]In Appendix C we solve a neoclassical growth model for a pure illustration purpose of our methodology

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:

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

Where the superscript on b_t^N indicates that it is a N-periods maturity bond. In each period the aggregate endowment in the economy is 1 unit that can be used for consumption, leisure and government expenditure. This leads to the aggregate resource constraint $c_t + g_t = 1 - l_t$, where $1 - l_t$ 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, coincides with the household's stochastic discount factor. This gives the following budget constraint for the government:

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

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. Households can only buy and sell this new issuance of government debt[4]. 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 bounds for government debt, we can express the government's problem as:

$$\max_{\{c_t\}_{t=0}^{\infty}, \{b_t\}_{t=0}^{\infty}} \mathbb{E}_0 \sum_t \beta^t \left[u(c_t) + v(1 - c_t - g_t)\right]$$

Subject to a sequence of measurability constraints for every time period t[5]:

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

[4]In principle, households are able to trade government securities in the secondary market among themselves, which should lead to have longer lags of b_t in the household's budget constraint. However, since we assume a representative household, such trades do not happen in equilibrium and to ease the notation we do not include this option.

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

And borrowing limits[6]:

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

The optimality conditions are:

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

$$\mu_t = \mathbb{E}_t(u_{c,t+N})^{-1}\left[\mathbb{E}_t(u_{c,t+N}\mu_{t+1}) + \frac{\xi_{U,t}}{\beta^N} - \frac{\xi_{L,t}}{\beta^N}\right] \quad (2)$$

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

Where μ_t is the Lagrange multiplier on the time t 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:

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

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 12 in Appendix B. For now, we use a standard CRRA utility for both consumption and leisure.

2.2 Solution

In this application the state space contains $2N+1$ variables and is multicollinear because the high correlation between contemporaneous variables and the presence of many lags of the same state variables. Equation 2 reveals that the recursive Lagrangian multiplier follows a random walk and, therefore, creates a strong collinearity between the state variables. For this reason, the model is hardly solvable using the standard PEA algorithm. To deal with the multicollinearity problem, the expected value terms in the first order conditions need to

[6] $\frac{\bar{M}_N}{\beta^N} \geq b_t^N$ is the government saving constraint, which is equivalent to having a borrowing constraint for the household sector.

be approximated using some functions of a core set[7] 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: parameterize the expected value terms with some functions of the state variables, iterate on the system of equations (1)-(3) for a large time horizon T, perform regressions of equations (4)-(6) (given the model implied dynamics) and obtain the new coefficients for the approximating functions. Iterate till convergence: when the predicted dynamic of the system and the weights of the approximating function do not change anymore.

More formally, the solution involves approximating the following expectations as a function of the state variables:

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

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

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

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

2.2.1 Condensed PEA

The solution method used in the 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. (2014) for more details):

1. Parameterize the 3 expectations in equations (4)-(6) as functions of a subset of state variables (called core set) and given an initial guess of the polynomials parameters[8]:
   - Set the bounds for the bond (see Maliar and Maliar (2003))
   - Simulate the model given the parameters

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

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

- Using the simulated dynamics, run a regression of each prediction term 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 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.

2.2.2 ANN and multicollinearity

Three salient features make machine learning appealing in this context: 1. Robustness to multicollinearity. 2. Non-parametric nature 3. Interest in making good predictions rather than estimating parameters. Motivated by these three reasons we choose to approximate the mapping between states and expectation terms (equations (4)-(6)) using an ANN. Refer to Appendix D for a general description of the structure of an ANN. In the following paragraph we illustrate, through the lenses of a simple example, the advantages provided by features 1 and 2.

Multicollinear state space and bias Parameterized expectations algorithm requires to approximate the expected value terms in order to make their forecasts during the stochastic simulation. One general problem is that the functional form (not just the parameters) between the state variables and the approximated terms is ex-ante unknown. A standard practice is to make this approximation using polynomials with a structure typically chosen through trial and error. When the policies are correctly specified, multicollinearity leads to consistent, yet noisy parameter estimates (which may prevent the algorithm from converg-

ing), but it is only an issue of parameter inference. However, if the policies are misspecified, multicollinearity can potentially lead to severely biased and less precise predictions as we show in the following example.

Imagine a model with an equilibrium solution characterized by the following policy function[9]:

$$y_i = 2x_{i1} + 3x_{i2} + x_{i2}^2 \tag{7}$$

The objective is to make accurate predictions for y_i using some mapping $f()$ between a multicollinear state space $\mathbf{x}_i$ and the policy y_i, such that the prediction for the data point 'i' is given by $\hat{y}_i = f(\mathbf{x}_i)$. The success of the prediction is usually evaluated with the mean squared prediction error (MSPE), which is the average prediction error for point 'i' over many training samples. The error can be decomposed into bias and variance terms.

$$MSPE_i = \mathbb{E}\left[(y_i - \hat{y}_i)^2\right] = \underbrace{[y_i - \mathbb{E}(\hat{y}_i)]^2}_{Bias_i^2} + \underbrace{\mathbb{E}\left[\hat{y}_i - \mathbb{E}(\hat{y}_i)\right]^2}_{Variance_i} \tag{8}$$

In the following experiment we first attempt to predict y_i using a linear polynomial, incorrectly assuming that the mapping between state variables and policy is linear $y_i = \beta_1 x_{i1} + \beta_2 x_{i2}$[10]. As a second attempt we approximate $f()$ with an ANN, which has a non-parametric nature. More specifically, we fix the validation set $\mathbf{y}^{validation}$ and estimate the regression coefficients and train the ANN with many training samples $\{\mathbf{y}^{train}, \mathbf{X}^{train}\}$ drawn from the same distribution and policy as per equation 7. We calculate the MSPE and its decomposition into bias and variance terms, according to equation 8. Figure 1 reports the average mean square prediction error and its decomposition for the entire validation set in function of the correlation between x_{i1} and x_{i2}. Note that the higher the correlation the higher the collinearity between x_{i1} and x_{i2}.

[9]The same idea applies for expected policy functions

[10]We check that the results are robust to many types of misspecification

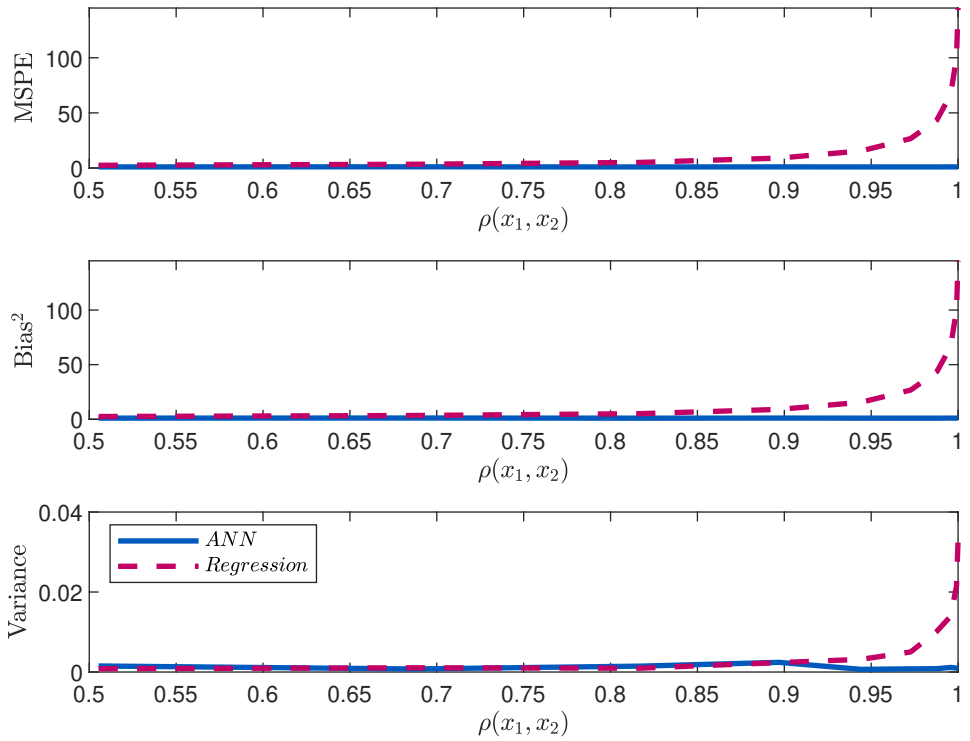

Figure 1: Mean square prediction error and its decomposition using polynomial regression and an ANN. Top Panel: $1/n \sum_{i=1}^{n} MSPE_i$, middle panel: $1/n \sum_{i=1}^{n} [y_i - \mathbb{E}(\hat{y}_i)]^2$, bottom panel: $1/n \sum_{i=1}^{n} \mathbb{E}[\hat{y}_i - \mathbb{E}(\hat{y}_i)]^2$, solid blue line - artificial neural network, red dashed line - polynomial regression

The higher the correlation between the state variables, the higher the inaccuracy of the regression model. Moreover, the decomposition of the MSPE (equation 8) suggests that most of the prediction error comes from the bias-square term (middle panel). The non-parametric nature of the ANN does not suffer from this misspecification problem and the backpropagation method used to train them is less sensitive to multicollinearity. This experiment suggests that the combination of these properties makes ANNs extremely useful when solving economic models characterized by policy functions (or expected policy functions) that are ex-ante unknown, contain significant non-linearities and whose domain presents multicollinear states. Note that another option would be to specify a rich polynomial structure with many higher order and cross terms. One problem with such approach is that higher order terms of the same variable are extremely multicollinear.

Tuning of the ANN parameters The calibration of an ANN typically requires to tune many parameters and, in this sense, it is often tougher to calibrate than to choose a polynomial structure. In table 1 we report the tuning parameters we use in our applications.

Parameter	Value
Hidden layers	1
Neurons	10
Activation function	Hyperbolic tangent sigmoid
Training algorithm	Levenberg-Marquardt backpropagation
Blending Factor (μ)	0.01
μ Decrease factor	0.01
μ Increase factor	10
Max num. epochs	1000

Table 1: ANN structure and parameters. See Appendix D for a highlight description of an ANN and its mostly commonly used terms

We choose to use only one hidden 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 our applications. Increasing the number of layers and neurons increases the capacity of the network to learn but, at the same time, can potentially lead to over-fitting and slow down computation in the learning phase. The number of neurons is calibrated in our simulations to avoid over-fitting. In particular, given the parameters in Table 12 (Appendix B), the number of neurons is 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 critical part of this procedure is represented by the fitting out-of-sample. After a certain number, it often stops improving and starts to diverge. We choose the number of neurons such that the fitting out-of-sample is minimized (see figure 2).

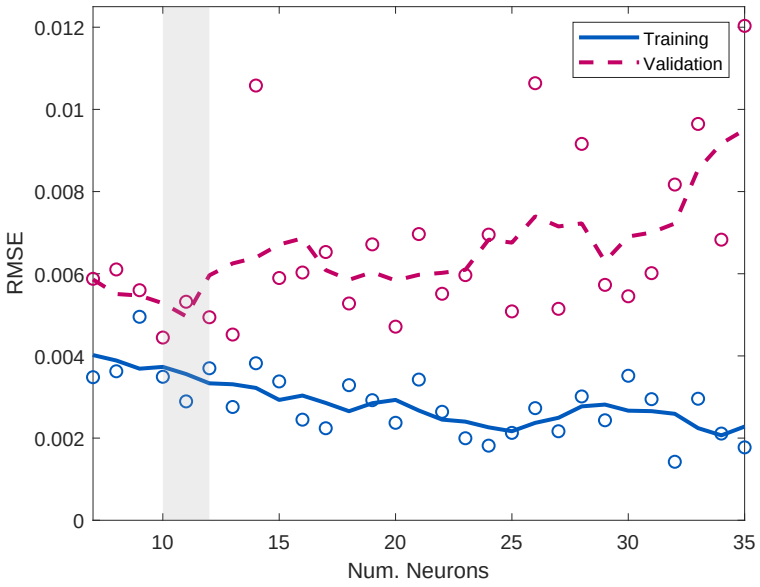

Figure 2: This graph shows the Root MSE in-sample (Training) and out-of-sample (Validation). The higher the number of neurons (x-axis) the lower the RMSE in-sample. The graph shows that there is a point after which the RMSE out-of-sample tends to diverge. We calibrated the number of neurons to minimize the RMSE out-of-sample (hence, to maximize the prediction power). This graph has been obtained solving the three bonds model of section 4.3 with parameters as specified in Table 12.

We train the ANN with a Levenberg-Marquardt backpropagation (LMB) algorithm. The LMB algorithm has been designed specifically to work with loss functions that take the forms of sum of square errors (largely used in this paper) and it is particularly suitable for function fitting problems. We decide to use this training method because it unifies the advantages of the gradient descent method (the simplest training algorithm that requires only information from the gradient) and the Newton method (computationally more expensive because it uses information from the Hessian but more precise and faster near an error minimum). When the blending factor μ is large LMB becomes a gradient descent method, when μ is 0 LMB becomes a Newton's method. The parameter μ is initially set to 0.01 and can be automatically increased and decreased by the algorithm using the factors specified in table 1.

The last parameter in the table is the maximum number of epochs. An epoch is one complete presentation of the training set to the ANN. The maximum number of epochs

is essentially set to be large enough so that the training algorithm never reaches it. The algorithm uses, in fact, an early stopping method. This works in a similar fashion to the method we explain above to calibrate the number of neurons. Within each single training phase, the entire data set is divided into training set (in-sample) and validation set (out-of-sample). As shown in figure 3, the training phase stops when the prediction error on the validation set starts to diverge.

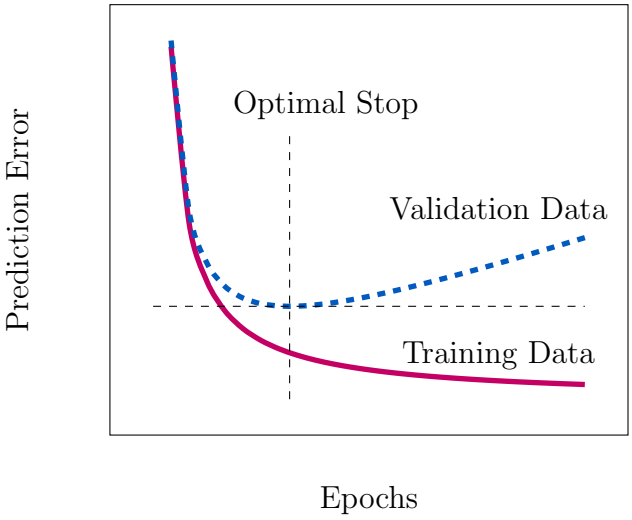

Figure 3: Graphical representation of the early stopping algorithm. Note that, although the concept is very similar to the one used in figure 2 to calibrate the number of neurons, this idea is here applied in a different context. The early stopping algorithm chooses the number of epochs in each training phase. The number of neurons is chosen exogenously and fixed throughout convergence of the algorithm.

Solving with an ANN The following points summarize how the solution algorithm works when the expectations in equations (4)-(6) are approximated with an ANN.

- Start with an initial guess for the weights of the ANN (initializing sequences are generated as in section 2.2.1)
- 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 $\mathbb{E}_t(u_{c,t+N}\mu_{t+1})$.
- Combine predictions and optimality conditions to generate an implied model dynamics

- Use the implied model dynamics 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.

2.2.3 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 an AR(1) process), same and parameters values as reported in Table 12 in Appendix B and the same starting point [11].

	ANN		C. PEA	
Projected term	*Residual*	$Residual_{\%}$	*Residual*	$Residual_{\%}$
$\mathbb{E}_t(u_{c,t+N}\mu_{t+1})$	0.02	2.00%	0.01	1.25%
$\mathbb{E}_t(u_{c,t+N})$	0.08	1.41%	0.07	1.16%
$\mathbb{E}_t(u_{c,t+N-1})$	0.08	1.42%	0.07	1.18%
Time	69s		695s	

Table 2: Table compares prediction residuals in a model with CRRA preferences using ANN-based Expectations algorithm and the Condensed PEA. Bond bounds are exogenously set at 100% the GDP. Both methods converge to comparable solutions: $Residual = \frac{1}{T}\sum\left|Y_i - \hat{E}_i\right|$ and $Residual_{\%} = \frac{1}{T}\sum\left|\frac{Y_i - \hat{E}_i}{Y_i}\right|$, where Y_i is the realized sequence and $\hat{E}_i$ - predicted sequence, simulation length T=1000.

As shown in Table 3, the ANN approach reaches a solution that features similar residuals

[11] We initialize polynomial coefficients and network weights with the simulated model when both bonds are 0 and keep increasing the borrowing limits during the simulation

on all three expectations [12]. Note that at convergence these residuals do not necessarily have to converge to zero, because they represent the average residual term between expected values and realized values. These residuals do need to stabilize: the difference between iterations of the residuals converge to zero. The key message from this table is that, given a comparable solution, the ANN approach takes around 90% 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 69s, 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 58s.

2.3 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 4, 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[13] 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).

[12] These residuals are calculated from the last time iteration before the algorithm converges

[13] 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).

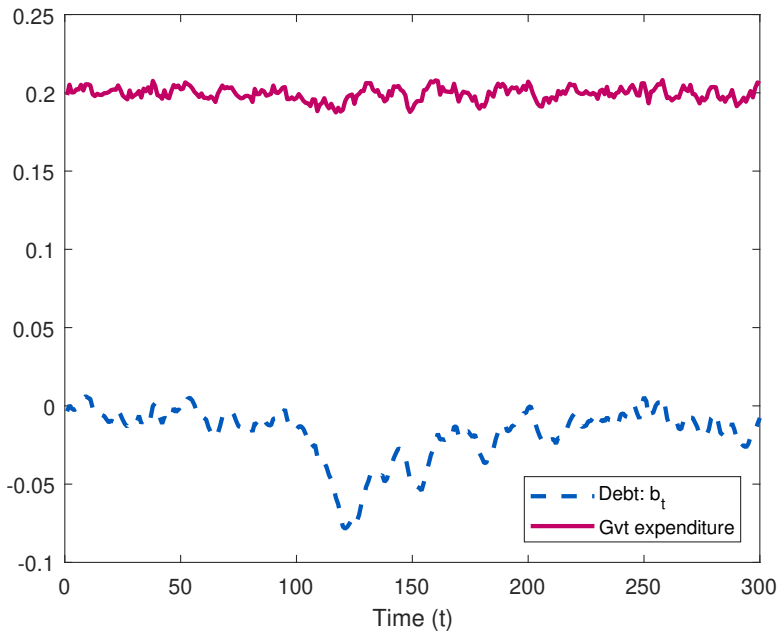

Figure 4: Solution to the one bond model using neural network. g_t follows AR(1) process. Solid line - exogenous sequences of government expenditure, dashed line - simulated optimal level of debt

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 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 due to its non-parametric nature, which can handle jumps and non-linearities induced by the interaction of borrowing/lending limits and variations in the exogenous process.

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

3.1 Adding Epstein-Zin preferences

Matching the relevant moments in the bonds data can be difficult with the model just considered. Taking model to the data requires to introduce different types of debt instruments, having a maturity structure and adding Epstein-Zin preferences. In this section we present an approach to solve an optimal government debt management problem with Epstein-Zin preferences and a generic number of debt instruments with different maturities.

Consider a representative household with 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 A for more details).

3.1.1 Sequential formulation of the Ramsey problem

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, 2018). First order condition 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 evaluate welfare at each corresponding solution.

When $\rho = \gamma$ Epstein-Zin preferences collapse into the CRRA case and It is easy to verify that the above optimality conditions collapse to the following set of equations:

$$c_t : u_{c,t} - v_{l,t} + \mu_t \big[u_{c,t} - v_{l,t} + u_{cc,t} c + v_{ll,t}(c_t + g_t)\big] + \sum_{i=1}^{N} (\mu_{t-i} - \mu_{t-i+1}) b_{t-i}^i u_{cc,t} = 0$$

$$b_{t+1}^i : \mu_t = \left[\mathbb{E}_t u_{c,t+i}\right]^{-1} \left[\mathbb{E}_t \mu_{t+1} u_{c,t+i} + \frac{\xi_{U,t}^i}{\beta^i} - \frac{\xi_{L,t}^i}{\beta^i}\right] \forall i$$

$$\mu_t : \sum_{i=1}^{N} b_t^i \mathbb{E}_t \beta^{i-1} \frac{u_{c,t+i-1}}{u_{c,t}} = s_t + \sum_{i=1}^{N} b_{t+1}^i \mathbb{E}_t \beta^i \frac{u_{c,t+i}}{u_{c,t}}$$

In the main body of the paper we describe computational strategy and numerical results using the CRRA preferences. Details on the implementation and results using Epstein-Zin preferences can be found in appendix A.

4 ANN-based expectation algorithm

We describe in detail the computational strategy when $\rho = \gamma$. A more general description of the algorithm can be found in Appendix A.

At every instant t the information set is $\mathcal{I}_t = \{g_t, \{\{b^i_{t-k}\}_{k=0}^{N-1}\}_{i=1}^{N}, \{\mu_{t-k}\}_{k=1}^{N}\}$. Consider projections of $\mathbb{E}_t u_{c,t+i}$, $\mathbb{E}_t \mu_{t+i} u_{c,t+i}$ 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[14] terms to project and, if one bond matures in 1 period, they reduce to five[15]. In particular, if the long maturity is $N > 1$ then the terms to approximate are the following:

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

The solution procedure is summarized by the following algorithm.
Given starting values $\mu_{t-1} = 0$ and initial weights for $\mathcal{ANN}$ [16], simulate a sequence of $\{c_t\}$, $\{\mu_t\}$ and $\{b^i_t\}$ as follows:

1. Impose the Maliar moving bounds (see Maliar & Maliar 2003) 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 functions are used instead of the ξ terms to avoid out of bound solutions, see Faraglia et al. (2014) for more details[17]. Use forward-states on the following i equations:

$$\forall i: \quad \mu_t = \mathcal{ANN}^i_1(\mathcal{I}_t)^{-1} \left[\mathcal{ANN}^i_2(\mathcal{I}_t) + \frac{\xi^i_{U,t}}{\beta^i} - \frac{\xi^i_{L,t}}{\beta^i} \right]$$

   Note that μ_t is now over identified. We tackle this problem by using the Forward-States approach as described in Faraglia et al. (2014). 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[18].

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

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

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

[17] We also find that including ξ terms explicitly in the training set improves prediction accuracy.

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

The equations to solve are:

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

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

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

   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 [19].

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 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[20].

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

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

4.1 Two bonds

In table 3 we compare computation times required to solve the two bonds problem with maturities of 1 and 10, when the expectations are parameterized 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, parametrization and initial conditions [21], the only difference being the way it approximates the expectation. Comparison of the two solutions is presented in table 3.

	ANN		C. PEA	
Projected term	*Residual*	$Residual_{\%}$	*Residual*	$Residual_{\%}$
$\mathbb{E}_t(u_{c,t+N}\mu_{t+1})$	0.019	0.32%	0.005	0.45%
$\mathbb{E}_t(u_{c,t+N})$	0.026	0.45%	0.021	0.36%
$\mathbb{E}_t(u_{c,t+N-1})$	0.0063	0.59%	0.021	0.36%
$\mathbb{E}_t(u_{c,t+S})$	0.026	0.44%	0.023	0.40%
$\mathbb{E}_t(u_{c,t+S}\mu_{t+1})$	0.0062	0.58%	0.006	0.60%
Time	20min		450min	

Table 3: Table compares prediction residuals in a model with CRRA preferences using ANN-based Expectations algorithm and the Condensed PEA. Bond bounds are exogenously set at 0.5. $Residual = \frac{1}{T}\sum \left|Y_i - \hat{E}_i\right|$ and $Residual_{\%} = \frac{1}{T}\sum \left|\frac{Y_i - \hat{E}_i}{Y_i}\right|$, where Y_i is the realized sequence and $\hat{E}_i$ - predicted sequence, simulation length T=1000

Compared to the one bond case, in this scenario there are even more possible parameter combinations the Condensed PEA needs to explore. In summary, both methods converged to qualitatively the same solutions with similar precision, but Condensed PEA took on average 450 minutes, whereas ANN took 20 minutes.

For illustration purposes we solve the model with CRRA preferences [22] using the same process for government expenditure and in one bond model. Figure 5 shows the simulated

[21] Initial conditions are the same as in the 1 bond model

[22] Solution with Epstein-Zin preferences can be found in appendix A.

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.

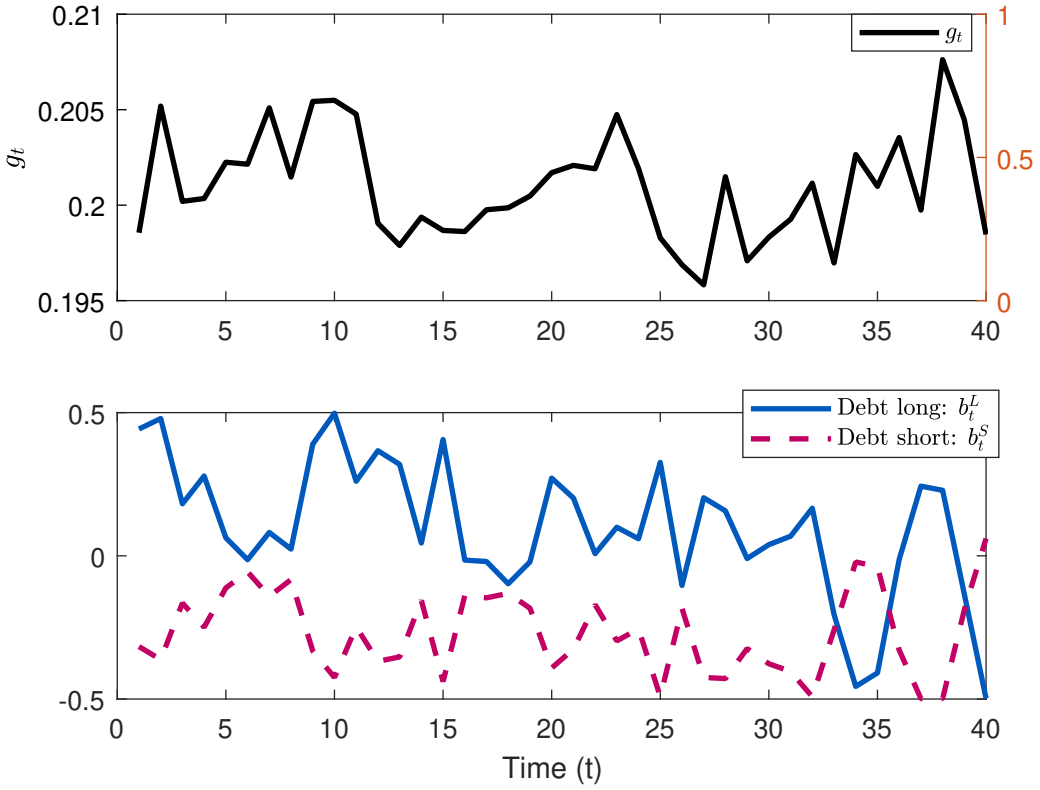

Figure 5: Simulated series with 2 bonds and CRRA preferences. Top panel: exogenous sequence of government expenditure. Bottom panel: optimal sequence of government debt. Solid line - long bond, dashed line - short bond. Bond bounds are exogenously set at 100% of GDP

4.2 Scalability

The capability of the ANN to digest the entire information set at once makes our methodology scalable. To demonstrate the scalability of our approach, we used our methodology to solve for the 3 and the 10 bonds optimal portfolios. Figure 6 shows solution times for the 1, 2, 3 and 10 bonds model with CRRA preferences using ANN and for 1 an 2 bond models using condensed PEA technique. Overall the ANN based method is not only faster, while being as

precise, but more scalable making it an appealing methodology to study more complicated portfolio problems. This is because ANN is able to absorb the whole vector of state variables at once, whereas condensed PEA needs to keep extracting the relevant information from the state space step by step, which adds an additional loop to the algorithm. Hence, the larger and the more collinear the state space, the more iterations Condensed PEA requires to extract the relevant information from the state space. Therefore, the solution time increases at a faster rate. Going from 2 to 3 bonds, solution time using ANN increases even though training time almost does not change. Most of this time increase happens because the system of first order conditions includes one more equation, which takes longer to solve for. Such speed gains allows us to investigate the optimal portfolio composition with 3 and 10 bonds as shown in the following two sections.

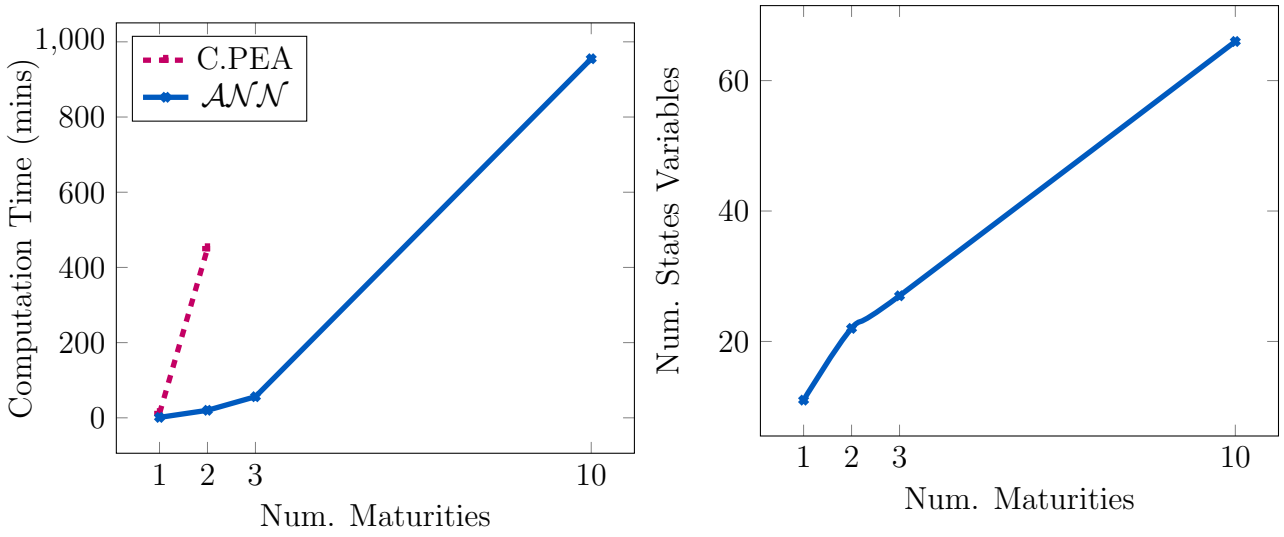

Figure 6: Left panel: solution time with Condensed PEA and an ANN based expectations algorithm as a function of government debt instruments. Right panel: number of state variables as a function of government debt instruments

4.3 Three bonds

In this section we extend the model with CRRA preferences adding a medium-term b_t^M bond of 5-year maturity.

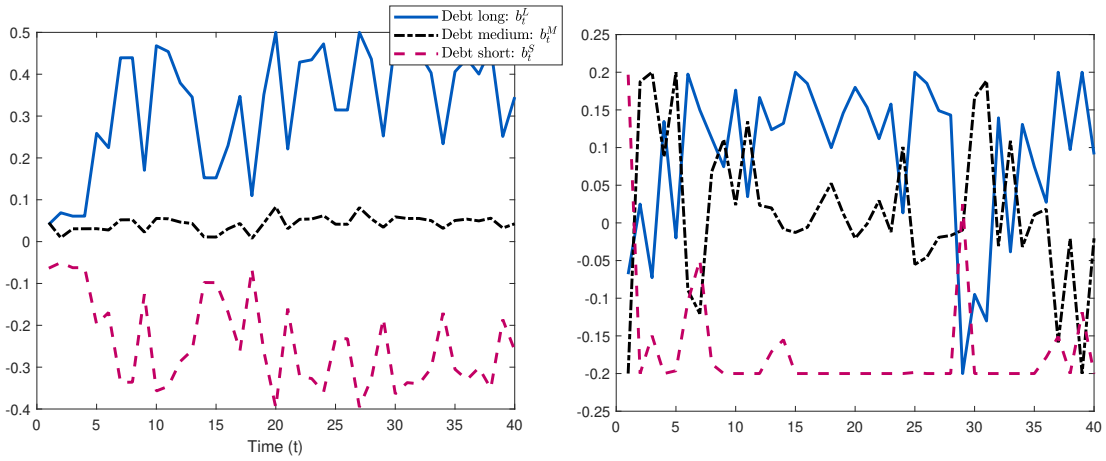

Figure 7: Simulated series with 3 bonds and CRRA preferences. Left: Solution when bond bounds are set at 0.5. Right: Solution when bond bounds are set at 0.2. Solid line - long bond (N=10), dashed line - short bond (S=1), dot-dashed line - medium term bond (M=5).

The solution reported in figure 7 shows that the intuition from the 2 bonds case still holds: it is optimal to exploit variations in the debt instruments with the most distant maturities, while the medium-term bond remains largely unused when the borrowing constraints are loose, as shown in the top panel of figure 7. As known since Angeletos (2002) and Buerra and Nicolini (2004), the price of the long bond is more responsive to shocks than the price of the short bond. This means that when the government goes short on the long bond and long on the short the value of government liability decreases in states characterized by a high g_t. In this sense, it is optimal to use debt instruments whose prices are the least correlated, which in this case are the long and short maturities. Likewise, the medium maturity bond remains largely unused. In contrast bottom panel of figure 7 shows that when the model is solved with the borrowing and lending constraints at ± 0.2, such constraints bind frequently the medium-term bond is used actively. Tables 13 and 14 in appendix B show the projection residuals from the solutions with both loose and tight borrowing constraints. In both cases residuals are small and of the similar size, illustrating the fact that the ANN is capable of

making accurate predictions in an environment where the occasionally binding constraints bind on the equilibrium path and are economically important.

4.4 Ten bonds

The advantages provided by the ANN-based expectation algorithm allowed us to solve the optimal portfolio problem containing a full spectrum of maturities between 1 and 10 with the borrowing and lending constraints set at 0.15 [23]. Figure 8 shows the optimal amount of debt, for each maturity, calculated as the average of a time simulation of 1000 periods.

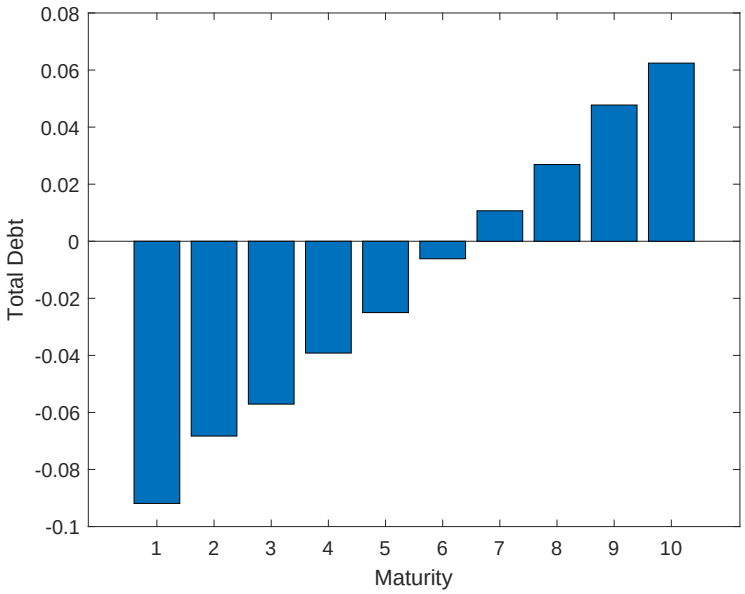

Figure 8: The graph reports, for each maturity, the average (over time) outstanding debt.

As clear from the graph, when constraints are tight all maturities play an important role in determining the optimal portfolio. The government decides optimally to lend using the 6 shortest maturities and to borrow using the remaining 4 longer maturities. Figure 9 shows a brief part of the bond dynamics given the exogenous shocks for government expenditure. The three shortest and longest maturities tend to be more volatile when the other ones respond less to government expenditure. In more formal terms, the shortest and longest maturities are used to hedge against the higher frequency movements in government expenditure and

[23] We used the same ANN hyperparameters as in the previous sections

mid-term maturities (e.g. 4, 5, 6 and 7) are used to hedge against the lower frequency movements in government expenditure.

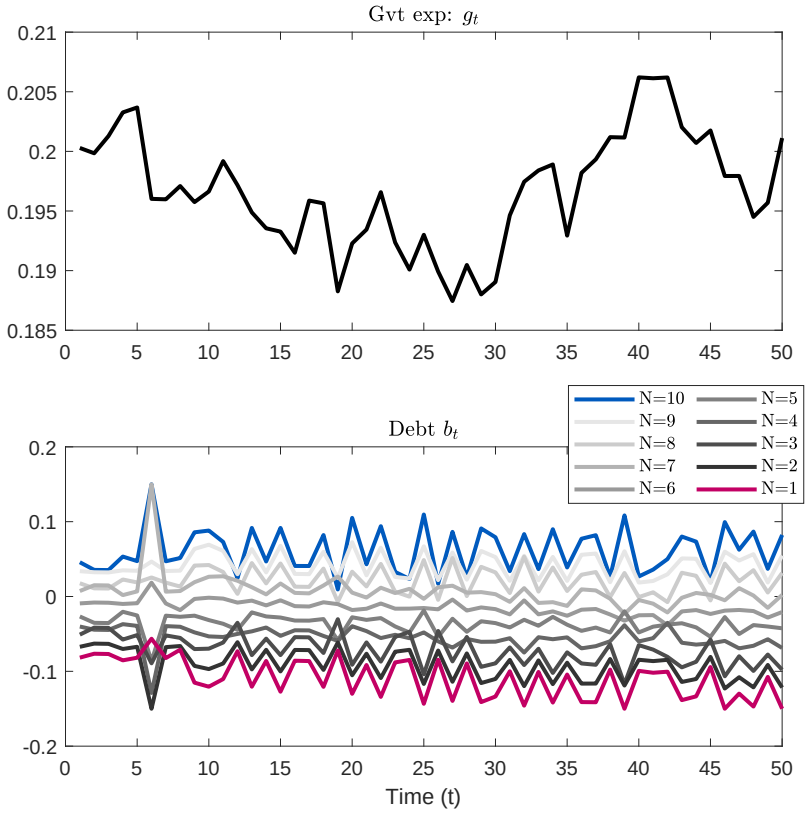

Figure 9: Simulated series with 10 bonds and CRRA preferences. Top panel: exogenous sequence of government expenditure. Bottom panel: optimal sequence of government debt. Bond bounds are both 0.15

A larger choice of debt instruments gives the Ramsey planner more tools to hedge against spending shocks and makes it easier to achieve the complete markets benchmark, characterized by constant labor taxes. In table 4 we explore how much the Ramsey planner is able to move towards the complete markets when the number of debt instruments increases from 1 to 10 when borrowing and lending constraints are set at 0.15. The first column of table 4 reports the standard deviation of taxes in the four different scenarios.

Clearly, the more maturities are available to the government the smaller is $\sigma(\tau)$ and the closer the solution is to the complete market case. We can see that the 10 bonds portfolio implies more stable labor taxes, reducing $\sigma(\tau)$ by almost 14% compared to the 3 bonds model

one and by around 32% compared to the case when only the long-term bond is available. The remaining three columns of the table report the correlation between $S = 1$, $M = 5$ and $L = 10$ maturities. The availability of additional maturities makes the long and the short bonds considerably less negatively correlated, signaling that less extreme positions would be needed to complete the markets as the number of maturities increases.

Num. maturities	$\sigma(\tau)$	$\rho(b^S, b^M)$	$\rho(b^S, b^L)$	$\rho(b^M, b^L)$
1	0.0091	-	-	-
2	0.0075	-	-0.74	-
3	0.0072	-0.49	-0.66	0.014
10	0.0062	0.46	-0.34	0.31

Table 4: The first column reports the standard deviation of taxes and the other columns show the correlations (ρ) between the short (S equivalents to 1 period), medium (M equivalents to 5 periods) and long (L equivalents to 10 periods) term bonds in a model with two, three and ten debt instruments and CRRA preferences.

4.5 Relation to alternative methods

Judd et al. (2011) propose a related method called generalized stochastic simulation algorithm (GSSA) to deliver high accuracy predictions as well as to resolve the multicollinearity problem. Judd et al. (2011) resolve this issue using standard econometric techniques, such as single value decomposition, principal components or ridge regression. At the same time, a high accuracy is achieved by approximating the policy function and integrating it using Gauss-Hermite quadrature instead of approximating the whole expectation term in the optimality conditions. In the Ramsey problem application it is particularly challenging to approximate the policy functions directly, because the expectations are over N periods ahead. In the context of the model presented in section 2 of this paper, the evaluation of $\mathbb{E}u'(c_{t+1})$ requires to know K_{t+2}, which is a function of K_{t+1}, which itself is a function of K_t and z_t, $K_{t+1} = \phi(K_t, z_t)$. That is, $c_{t+1} = z_{t+1}\phi(K_t, z_t)^\alpha + (1-\delta)\phi(K_t, z_t) - \phi(\phi(K_t, z_t), z_{t+1})$. When the expectation is N periods ahead, evaluation of $\mathbb{E}u'(c_{t+N})$ using GSSA approach would

require to iterate on the approximated policy function and the budget constraint N times to have the value for K_{t+N+1}, which would result in an imprecise evaluation of the expectation and lower stability of the algorithm compared to an application where the forecast is one period ahead.

In this section we attempt to solve the problem in sections 3.2.3 and 5.1 with the standard PEA, but using ridge regression instead of the ANN. Ridge regression imposes a penalty on the size of the coefficients providing stability but causing them to be downward biased. In order to choose the penalty parameters it is required to add an additional loop and to solve the model with many different combinations of them. However, in this application such a choice is non-trivial, since the regression is performed on simulated data, which is not fixed and not exogenous to the choice of the penalty: simulated data depends on the coefficients and penalties obtained in the previous iteration. Additionally, the optimal penalty is different at every step of the fixed point iteration as the Maliar bounds open. Two most common criteria to pick the penalty are ridge trace and cross-validation. We adopt the latter approach and, more specifically, we change the penalty dynamically at each step of the fixed point iteration in the regression stage. We select the penalty parameters, which minimize the mean squared prediction error between predicted and simulated sequences [24]. Equation 9 illustrates the penalty selection procedure.

$$\begin{aligned} &\min_{\kappa} ||Y - X\hat{\boldsymbol{\beta}}||_2^2 \quad \text{s. t.} \\ &\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} ||Y - X\boldsymbol{\beta}||_2^2 + \kappa||\boldsymbol{\beta}||_2^2 \end{aligned} \tag{9}$$

4.5.1 Results

From our numerical experiments we discover that PEA with ridge regression converges only under specific conditions. Note that throughout the paper we have been using Maliar bounds and debt limit $\bar{M}, \underline{M}$ was set to ± 0.5, approximately 100% of the GDP. This effectively introduces an occasionally binding constraint, which makes the multicollinearity problem even

[24] similarly, Judd et al. (2011) choose the smallest penalty that ensures numerical stability of the fixed point iteration, which also provides a high accuracy solution.

more severe if debt sequence tends to stay in the constrained region. This introduces additional instability if the simulated debt sequence tends to spend a different amount of time at the constraint across the consecutive fixed point iterations. Besides, the algorithm requires using Maliar bounds, which potentially cause even more instability since the borrowing constraint changes as the bounds are opening. We find this to be crucially important. For an illustration we consider the one N period bond model of section 2. We run the algorithm setting the $\bar{M}, \underline{M}$ at ± 0.5 and at ± 0.9. At every iteration we compute the maximum difference between the ridge regression coefficients at the current and the previous iteration. When the borrowing constraint is set to 0.9, the regression coefficients stabilize when the Maliar bounds are wide enough and the borrowing constraint stops binding. In contrast, when the $(\bar{M}, \underline{M})$ are set at ± 0.5, the constraint binds, requiring to use large penalty parameters, which prevents the algorithm from converging. Table 5 illustrates this point. The first row shows the average change in ridge coefficients across consecutive iterations for the last 100 iterations. The second row shows the percentage of time that debt visits the constraint in the last iteration[25]. When the bound is at 0.5, it binds around 50% of the time and ridge coefficients never stabilize. Figure 15 in appendix B shows how ridge coefficients change during the last 40 iterations: when the bound is at 0.9, they stabilize once the constraint stops binding, which does not happen in $\bar{M}, \underline{M} = \pm 0.5$ case. As a result this model does not stabilize and does not provide accurate approximations (see Table 6).

	$\bar{M}, \underline{M} = 0.5$	$\bar{M}, \underline{M} = 0.9$
$\mathbb{E}(\Delta\beta)$	2.8852	0.0470
% constraint binds	0.4900	0

Table 5: Solution using ridge regression. First row: average change in β over the last 100 iterations, Second row: % of time debt constraint binds

[25]when $\bar{M}, \underline{M}$ was set to ± 0.5 the algorithm did not converge. For comparison purpose we ran the same number of iterations that were needed for $\bar{M}, \underline{M} \pm 0.9$ version to converge.

	$\mathbb{E}_t(u_{c,t+N}\mu_{t+1})$	$\mathbb{E}_t(u_{c,t+N})$	$\mathbb{E}_t(u_{c,t+N-1})$
$\bar{M}, \underline{M} = 0.5$	0.1371	0.2431	0.2321
$\bar{M}, \underline{M} = 0.9$	0.0105	0.0640	0.0632

Table 6: Prediction accuracy using ridge regression. First row - bounds on government bonds set at 0.5, second row - bounds set at 0.9. $\frac{1}{T}\sum \left| Y_{t+N} - \hat{E}_t(Y_{t+N}) \right|$

In the model with 2 bonds the problem described above becomes even more severe for two reasons. First, the long and short bonds are highly negatively correlated. Secondly, the optimal debt policy is very volatile and tends to hit the borrowing constraint frequently. As a result the algorithm breaks even before the Maliar bounds open[26]. Alternatively, we could have tried using singular value decomposition or a principal components analysis, but we chose not to for two reasons. First, ridge regression is supposed to work better than SVD when multicollinearity is severe, which is the case of our model. Second, we did not use principal component analysis since the condensed PEA technique effectively does the same extraction of orthogonal components, just iteratively.

5 International Business Cycles with Endogenously Incomplete Markets

To illustrate our method's ability to deal with occasionally binding constraints we solve the international business cycle model with endogenously incomplete markets as in (Kehoe and Perri, 2002). In Kehoe and Perri (2002) endogenous incompleteness arises from participation constraints added onto the standard multi-country business cycle model of Backus et al. (1992). This friction comes from the assumption that countries are allowed to renege on their debts and leave the risk sharing contract at the cost of staying in the autarky for ever. In order to keep countries in the risk sharing contract, the social planner needs to make

[26]In the 1 bond code ridge penalty was a scalar. When solving the 2 bonds model we allowed every coefficient to have different penalty, but this did not result in an improvement.

transfers between countries such that all countries prefer to stay in the contract at every point in time. (Kehoe and Perri, 2002) show that this friction helps to resolve the international business cycle puzzles stated in Backus et al. (1992), namely the fact that complete markets models produce a too high correlation of consumption across countries relative to output and that it misses the sign of correlation between investment and employment across countries. Endogenous market incompleteness complicates the model solution significantly. The friction introduces an occasionally binding constraint for every country, which can potentially bind at any point in the state space. This randomness poses a complicated prediction problem because it introduces kinks into policy functions at ex-ante unknown points. Additionally the problem is made recursive using recursive contracts (Marcet and Marimon, 2019), where changes to the recursive Lagrangian multipliers are permanent, implying permanent changes to other endogenous variables. This renders local methods not feasible. A common practice is to solve such models using value function or policy function iteration as done in Kehoe and Perri (2002). However, these methods become impossible to use when the number of state variables increases beyond 5 or 6; for instance the two-country version has 5 state variables. We extend the model to 3 countries featuring 8 state variables and solve it using our method. In contrast to the fiscal policy application, here we approximate the integrands instead of the expectation terms in the first order conditions. The latter is usually smooth even if the policy function has kinks or non-linearities, which is why the regular polynomial PEA approach works in most cases. In this particular model the policy functions are extremely non-smooth and accurate solution can only be achieved by approximating the integrands, which is possible due to the non parametric nature of the neural network. More precisely, approximating the integrands allows us to evaluate the enforcement constraints more precisely thanks to the fact that the effect of the choice of the future endogenous states is taken into consideration.

5.1 Model

A world economy consists of three countries populated by identical infinitely lived consumers deriving utility from consumption and disutility from labor and each country posses the same production technology, using capital and effective labor as inputs. The social planner's objective is to maximize aggregate expected utility subject to the pooled resource constraint

(equation 11) and enforcement constraints (equation 12) that need to hold for every country. Absent (equation 12) the model is the standard complete markets multi-country economy.

$$\max_{c_i,l_i}\left[\sum_{i=1}^{3}\sum_{t=0}^{\infty}\sum_{s^t}\beta^t\pi(s^t)U(c_i,l_i)\right] \tag{10}$$

$$\sum_{i=1}^{3}\left[c_i(s^t)+k_i(s^t)\right]=\sum_{i=1}^{3}\left[F(k_i(s^t),A_i(s^t)l_i(s^t))+(1-\delta)k_i(s^t)\right] \tag{11}$$

$$\sum_{r=t}^{\infty}\sum_{s^r}\beta^{r-t}\pi(s^r|s^t)U(c_i(s^r),l_i(s^r))\geq V_i(k_i(s^{t-1},s^t))\quad\forall i \tag{12}$$

The enforcement constraint requires that the expected utility from staying in the contract needs to be higher that the value of leaving it and remaining in autarky indefinitely $V_i(k_i(s^{t-1},s^t))$.

$$V_i(k_i(s^{t-1},s^t))=\max_{c_i,k_i,l_i}\sum_{r=t}^{\infty}\sum_{s^t}\beta^{r-t}\pi(s^r|s^t)U(c_i(s^r),l_i(s^r))\quad\text{s.t.}$$

$$c_i(s^r)+k_i(s^r)\leq F(k_i(s^{r-1},A_i(s^r)l_i(s_r)))+(1-\delta)k_i(s^{r-1})$$

Optimality conditions imply the equalization of marginal utilities of consumption across countries as long as neither country wants to leave the risk sharing contract. But when a country j finds it optimal to leave, the social planner makes a transfer to country j so that equation 12 holds. In that case a positive ν_j [27] induces a permanent distortion in the ratio of marginal utilities.

$$\frac{U_{ic}(s^t)}{U_{jc}(s^t)}=\frac{1-\nu_i(s^t)}{1-\nu_j(s^t)}z_{ij}(s^{t-1})\qquad z_{ij}(s^t)=\frac{1-\nu_i(s^t)}{1-\nu_j(s^t)}z_{ij}(s^{t-1}) \tag{13}$$

The other two equations are the standard inter and intra temporal optimality conditions. Besides distorting the marginal utility ratio across countries, binding enforcement constraint also enters the RHS of equation 13 introducing a kink in the individual country saving rules.

[27] $\nu(s_t)=\frac{\mu_i(s^t)}{M_i(s^t)}$ where $\mu_i(s^t$ is the multiplier on the participation constraint and $M_i(s^t)=M_i(s^{t-1}+\mu_i(s_t))$. This change of variable is done to reduce the state space.

$$U_{ic}(s^t) = \beta \sum \pi(s^{t+1}|s^t) \left[\frac{U_{ic}(s^{t+1})}{1-\nu_i(s^{t+1})}(F_{ik}(s^{t+1}) + 1 - \delta) - \frac{\nu_i(s^{t+1})}{1-\nu_i(s^{t+1})} V_{ik}(s^{t+1}) \right] \quad (14)$$

$$\frac{U_{il}(s^t)}{U_{ic}(s^t)} = F_{il}(s^t) \quad (15)$$

The model is solved by a slight modification of the algorithm used in the fiscal policy application. The first difference is that instead of approximating the whole expectation term in the first order conditions we use an ANN to approximate the integrands in the expectations on the LHS of (12) and RHS of (15). The stochastic simulation is performed to obtain the model endogenous variables $\{\{c_i(s^t), l_i(s^t), k_i(s^t)\}_{i=1}^3$ and $\{v_i(s^t)\}_{i=1}^3$ from equations (11)-(15) at every point in time. We discretize the exogenous processes using the multi-variate Tauchen's method using 6 states and use the implied probability transition matrix to perform the integration. The second difference arises from the enforcement constraints. Since the binding pattern is ex-ante unknown, in every time step of stochastic simulation we solve for the endogenous variables assuming that equation (12) holds with strict inequality for every country ($\{v_i(s^t)\}_{i=1}^2$)=0) and check if this is indeed the case. If not, there are 6 cases: equation (12) binds for country 1,2 or 3, equation (12) binds for countries 1 and 2, 2 and 3 or 1 and 3. In equilibrium the enforcement constraint cannot bind for all countries at the same time. If the enforcement constraint binds for country i, the system is resolved by imposing that equation (12) holds with equality for country i and is slack otherwise ($\{v(s^t)\}_{-i} = 0$). The ANN is trained using the data generated from the stochastic simulation. The relevant state variables used in the prediction problem are $\{\{k_{it}, A_{it}\}_{i=1}^3\}_{t=1}^T$ and $\{\{z_{it}\}_{i=1}^2\}_{t=1}^T$. We use two separate artificial neural networks in the prediction problems in equations (12) and (14). The algorithm stops when the error between predicted and simulated series converges. The ANN used in this section is the same as table 1 with the only exception that the hidden layer is composed by 12 neurons.

5.2 Quantitative Implementation

In this section we present the numerical solution of the model with two and three countries. Functional forms for preferences and technology follow Backus et al. (1992) and Kehoe and

Perri (2002): $U(c,l) = (c^{\gamma}l^{1-\gamma})^{1-\sigma}/(1-\sigma)$ and $F(k,Al) = k^{\alpha}(Al)^{1-\alpha}$. The key difference from those papers is the specification of the exogenous productivity process. In this application we consider a case with three countries: US, China and a weighted aggregate of five major European economies - Germany, France, UK, Italy and Spain [28]. We assume that each country's productivity follows an AR(1) process and estimate parameters ρ and σ using a yearly TFP data [29]. Correlation of shocks is then calculated using estimated residuals $\hat{\epsilon}_t$.

$$log(A_t) = \rho \log(A_{t-1}) + \epsilon_t, \qquad \epsilon_t \sim N(0,\sigma)$$

Table 7 summarizes the remaining parameter values. β is set to 0.985 implying a yearly interest rate of 1.5%. Depreciation rate is set to 10%.

Parameter	Value
Preferences	$\beta = 0.985$, $\gamma = 2$, $\sigma = 2$
Technology	$\delta = 0.1$, $\alpha = 0.36$
Exogenous Process:	$a_{US} = 0.4798$, $a_{CHN} = 0.7127$, $a_{EUR} = 0.3990$
	$\sigma(\epsilon_{US}) = 0.0079$,$\sigma(\epsilon_{CHN}) = 0.0241$, $\sigma(\epsilon_{EUR}) = 0.0069$
	$\rho(\epsilon_{US}, \epsilon_{CHN}) = 0.1016$, $\rho(\epsilon_{US}, \epsilon_{EUR}) = 0.2178$, $\rho(\epsilon_{CHN}, \epsilon_{EUR}) =$ 0.0300

Table 7: Parameter Values used in 2 and 3 country models

5.2.1 Results

Next we compare the model implied moments in the two and three country models. In the first case we consider US and Europe only. Table 8 compares the model implied moments with data counterparts calculated using yearly growth rates. In contrast to Kehoe and Perri (2002), the enforcement constraints do not resolve the international business cycle puzzle: i. consumption is still more correlated across countries than output and labor and ii. investment are negatively correlated, which happens because the enforcement constraints

[28] An aggregate European TFP is constructed by weighting individual TFP's by country's share in the aggregate GDP

[29] Yearly data is used for all countries due to limited availability of Chinese data at a higher frequency. Estimation is done after applying HP filter on a logarithm of original data

almost never bind in equilibrium[30].

	$\rho(US, EU)$	
	Model	Data
c_t	0.89	0.15
l_t	−0.35	0.37
i_t	−0.86	0.22
y_t	0.06	0.32

Table 8: Cross country correlations between US and Europe. Left - 2 country model, Right - data. Data: growth rates of yearly series, source: St. Louis FRED database

Consider now the three-country model with China. Looking at table (9), the model gets closer to the data when China is added to the model. We can observe three effects: i. it reverses the sign of cross country correlations of investment and labor between US and Europe, ii. cross-country correlation of output rises and iii. cross-country correlation of consumption goes down. The inclusion of China into the international market changes the equilibrium extent of risk sharing, because the properties of its exogenous process cause the enforcement constraints to bind more frequently for every country. The intuition is the following. China's productivity is less correlated with other countries meaning that China is likely to get a positive shock when the other two countries are in a slack. At the same time its shocks are more persistent, which makes the option of leaving the contract more tempting in states of high productivity therefore, to incentivize China to stay, the social planner needs to make larger transfers. This tends to happen when US and Europe are in a slack and would prefer to leave the contract instead of letting the social planner transfer their resources to China. As a consequence, the addition of a third country makes the market incomplete. The impulse response functions below illustrate this point.

[30]We find that this binding pattern is sensitive to the choice of β. When it increases, continuation value in the enforcement constraint becomes increasingly volatile causing the constraint to bind more frequently. The difference to (Kehoe and Perri, 2002) arises because we use yearly calibration with lower β.

	Model			Data		
	$\rho(US, CHN)$	$\rho(US, EU)$	$\rho(EU, CHN)$	$\rho(US, CHN)$	$\rho(US, EU)$	$\rho(EU, CHN)$
c_t	0.47	0.65	0.41	-0.20	0.15	0.01
l_t	-0.27	0.07	-0.34	0.31	0.37	-0.40
i_t	-0.47	0.45	-0.92	-0.27	0.22	-0.53
y_t	0.05	0.27	-0.31	-0.12	0.32	0.21

Table 9: Cross country correlations in the three-country model and in the data. Data: growth rates of yearly series, source: St. Louis FRED database

5.3 Response to a Productivity Shock

We illustrate the quantitative findings by analyzing impulse responses to a productivity shock in one country. Figures (10) - (11) plot the impulse response functions to a two standard deviations productivity shock in the Europe in a two-country model. Right panel of figure (10) shows that in response to the shock, enforcement constraints do not bind for any country.

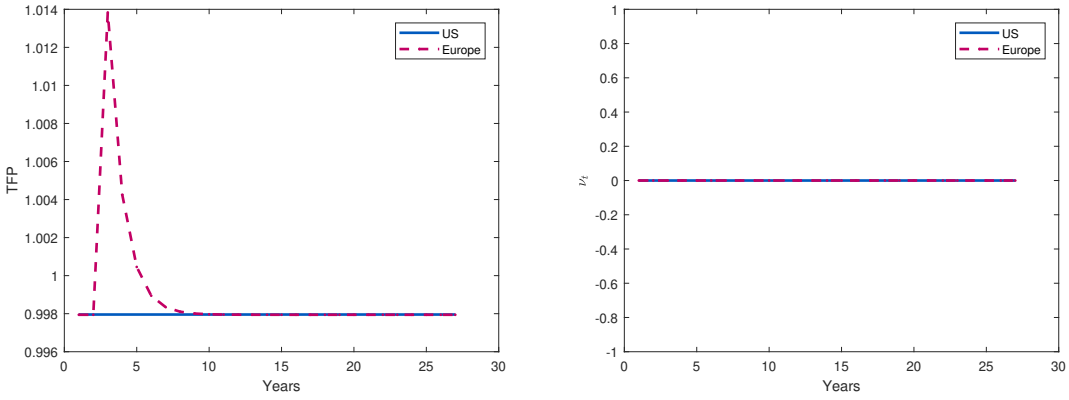

Figure 10: Impulse responses in a two country model. Left: exogenous TFP sequence, right: multiplier on the enforcement constraint. Solid line: US, dashed line: Europe

This yields a behavior for consumption and capital analogous to the complete markets model. Left panel of figure (11) shows that the social planner moves goods to US in order to equate the ratio of marginal utilities. Consumption increases in both countries. At the

same time, the social planner prefers to allocate capital in Europe (right panel of figure 11) as it becomes temporary more productive.

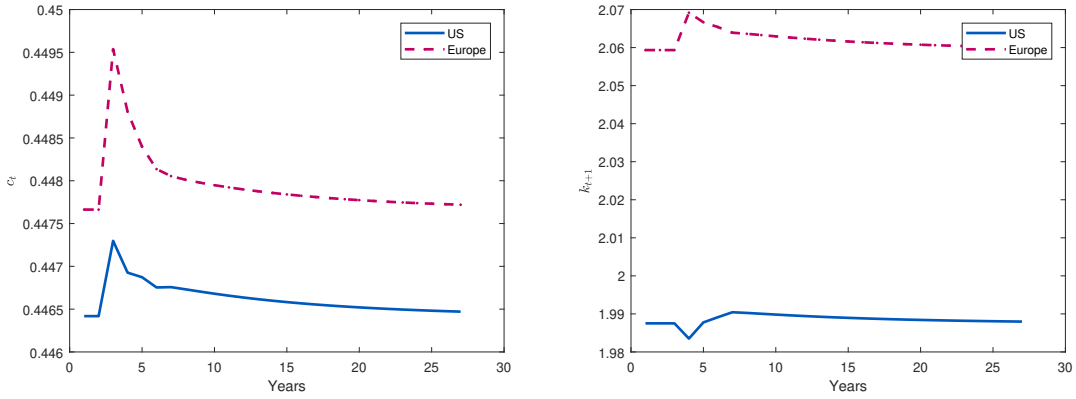

Figure 11: Impulse responses in a two-country model. Left: consumption, right: Capital. Solid line: US, dashed line: Europe

Figures (12) - (13) plot impulse responses to a two standard deviations TFP shock to China in a three country specification. In this case China has incentives to leave the risk sharing contract and, as a consequence, the enforcement constraints bind for all countries (right panel of figure 12). In order to keep China in the contract, the social planner needs to make a consumption transfer from the other two countries, causing consumption in Europe to fall (left panel of figure13) which is the opposite of what happens in the two-country case. This triggers a feedback mechanism. Since Europe needs to pay to keep China in the contract, it finds it optimal to leave and the social planner needs to give Europe a permanent increase in consumption. This causes a consumption drop in the US, which now finds it optimal to leave the contract instead of financing the other two countries. Mathematically, this corresponds to an increase in the recursive Lagrange multiplier in the right panel of figure 12 and lower consumption in period 6 in the left panel of figure (13). Economy stabilizes and the enforcement constraints stop binding as technology reverts back to the initial level.

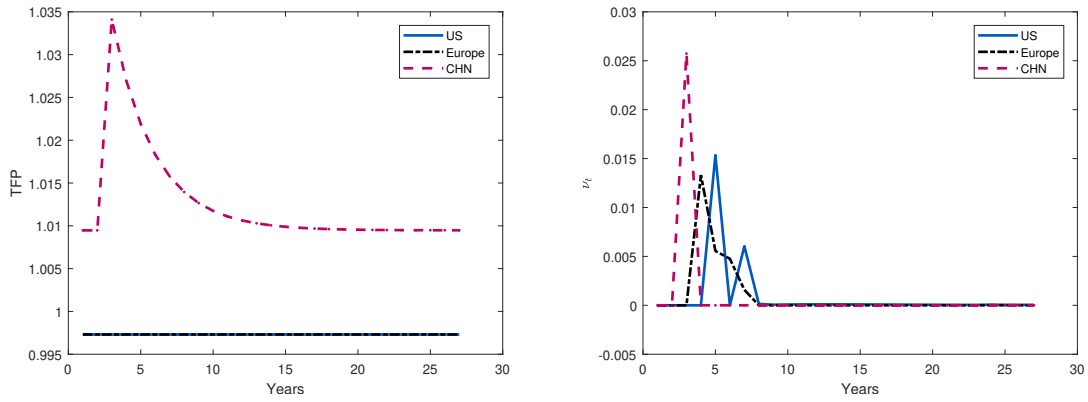

Figure 12: Impulse responses in a three-country model. Left: exogenous TFP sequence, right: multiplier on the enforcement constraint. Solid line: US, dashed line: China, dot-dashed line: Europe

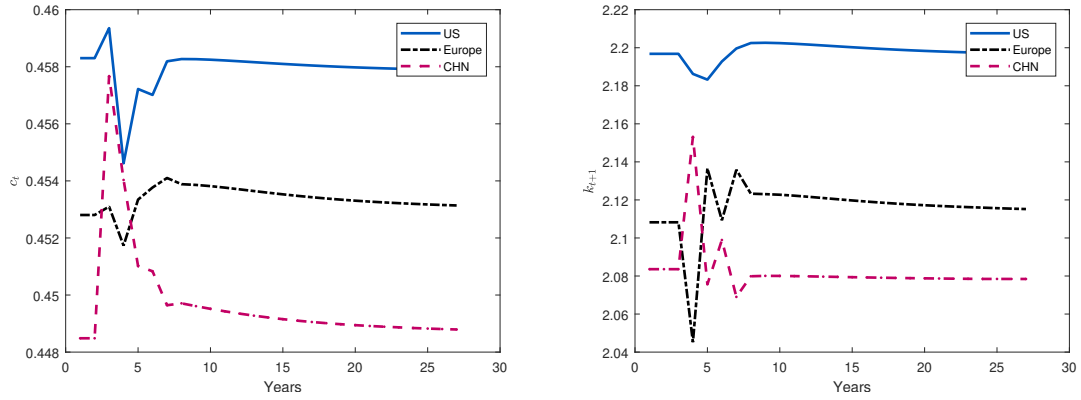

Figure 13: Impulse responses in a three-country model. Left: consumption, right: capital. Solid line: US, dashed line: China, dot-dashed line: Europe

Table (10) shows welfare comparison between two and three country specifications. Since adding a third country makes markets incomplete, welfare effects are ambiguous. The inclusion of China increases the expected utility for Europe but it is lower in the US relative to the two-country case. The second row reports the percentage of time that the enforcement constraint binds for each country. In equilibrium it never binds in the two-country model and roughly a third of time when China is added to the risk sharing contract.

		US	EUR	CHN
2 Countries	$\mathbb{E}[U(c,l)]$	-1.7450	-1.7504	
	Binding Frequency	0	0	
3 Countries	$\mathbb{E}[U(c,l)]$	-1.7455	-1.7468	-1.7457
	Binding Frequency	0.3567	0.3753	0.3167

Table 10: Welfare Comparison, 2 and 3 country models. First row: Expected utility, second row: percent of time the enforcement constraint binds for a particular country

The use of an ANN in this application enables to approximate highly nonlinear integrands, which allows to achieve high accuracy solutions using stochastic simulation. Such grid free methods combined with an artificial neural network make it possible to solve multi-country models with economically interesting non-linearities that would be challenging with grid-based methods or polynomial-based PEA. In this application we show that there could be crucial differences in model behavior depending on what types of countries are included into risk sharing contract: the properties of their exogenous TFP processes, as well as their size, can have important positive and normative implications.

6 Conclusion

In this paper we show how an artificial neural network can be applied to solve macroeconomics and asset pricing models globally. In order to highlight the salient features of our method we present two applications: i. a Ramsey taxation problem with market incompleteness and multiple maturities and ii. an international business cycle application with Europe, US and China with an endogenously incomplete international market.

The Ramsey taxation problem is particularly challenging since the state space is large and highly multi-collinear. We show that the ANN-based Expectations Algorithm is not only faster, but also more scalable because of its ability to digest the entire information set at once. Moreover, we extend the model to include a full spectrum of 10 maturities, from 1 period to 10 periods, and we show that all maturities are used significantly in composing the optimal portfolio.

The international business cycle model poses other challenges, such as the need to approximate the policy functions directly instead of their expectations. This is needed due to the high sensitivity of the enforcement constraints to the future endogenous state variables. In this context, the ANN-based Expectations Algorithm becomes handy to approximate the kinks contained in the policy functions in proximity of the constrained region. Our methodology enables us to include China in the risk sharing contract between US and Europe. We find that adding a third country significantly complicates the risk sharing, resulting in a welfare loss for the US compared to the two-country case.

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.

Marlon Azimovic, Luca Gaegauf, and Simon Scheidegger. Deep equilibrium nets. *Working Paper*, 2019.

David K. Backus, Patrick J. Kehoe, and Finn E. Kydland. International real business cycles. *Journal of Political Economy*, 100(4):745–775, 1992.

Anmol Bhandari, David Evans, Mikhail Golosov, and Thomas J. Sargent. The optimal maturity of government debt. *Working Paper*, 2019.

Saki Bigio, Galo Nuño, and Juan Passadore. A framework for debt-maturity management. *Working Paper*, 2019.

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

Wouter den Haan and Albert Marcet. Solving the stochastic growth model by parameterizing expectations. *Journal of Business and Economic Statistics*, 8(1):31–34, 1990.

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. Optimal fiscal policy problems under complete and incomplete financial markets: A numerical toolkit. *Working Paper*, 2014.

E. Faraglia, A. Marcet, R. Oikonomou, and A. Scott. Government debt management: the long and the short of it. *Review of Economic Studies*, Forthcoming.

Jesús Fernández-Villaverde, Samuel Hurtado, and Galo Nuño. Financial frictions and the wealth distribution. *Working Paper*, 2019.

Kenneth Judd, Lilia Maliar, and Serguei Maliar. Numerically stable and accurate stochastic simulation approaches for solving dynamic economic models. *Quantitative Economics, 2, 173-210*, 2:173–210, 2011.

Anastasios G. Karantounias. Optimal fiscal policy with recursive preferences. *Review of Economic Studies*, 85:2283–2317, 2018.

Patrick J. Kehoe and Fabrizio Perri. International business cycles with endogenous incomplete markets. *Econometrica*, 70:907–928, 2002.

Hanno Lustig, Christopher Sleet, and Sevin Yeltekin. Fiscal hedging with nominal assets. *Journal of Monetary Economics*, 55:710–727, 2008.

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. *Econometrica, forthcoming*, 2019.

S. Mullainathan and J. Spiess. Machine learning: An applied econometric approach. *Journal of Economics Perspectives*, 31(2):87–106, 2017.

Simon Scheidegger and Ilias Bilionis. Machine learning for high-dimensional dynamic stochastic economies. *Journal of Computational Science*, 33:68–82, 2019.

Appendix

Appendix A - Optimal Fiscal policy with Epstein - Zin preferences

Household Problem 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 $R_{t+1} = W_{t+1}/q_t b_{t+1} = 1/q_t$:

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

The HH problem can be rewritten as:

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

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

Optimal consumption (FOC_c):

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

Optimal labor supply (FOC_h):

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

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:

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

Ramsey Problem:

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

Subject to:

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

Optimality conditions:

- c_t

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 the end of the Appendix B we explain how to find an expression for $\frac{\partial V_0}{\partial c_t(g^t)}$ [31], with which is possible to find the following optimality condition:

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

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[32].

- b_{t+1}^i

The first order condition with respect to b_{t+1}^i, yields the following inter-temporal expression for the promise keeping Lagrange multiplier μ:

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

- V_t

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

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

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

which after rearranging yields the following recursion for λ_t^V [33]:

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

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

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 $j = 2$:

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

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

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

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

Algorithm to solve with Epstein - Zin preferences

At every instant t the information set is $\mathcal{I}_t = \{g_t, \{\{b_{t-k}^i\}_{k=0}^{N-1}\}_{i=1}^N, \{\mu_{t-k}\}_{k=1}^N, \{\lambda_{t-k}^V\}_{k=1}^N\}$. 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[34] we would have $4N+1$ inputs and 8 outputs. In particular, use the following notations for each output:

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

[34] One with maturity 1 and the other with maturity 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:

$$\forall i: \quad \mu_t = \frac{\mathcal{ANN}^i_3(\mathcal{I}_t)}{\mathcal{ANN}^i_2(\mathcal{I}_t)}$$

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

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

ii. $$V_0^{\rho}(1-\beta)\mathcal{X}_{0,t} U_t^{-\rho} \frac{\partial U_t}{\partial c_t(g^t)} + \mu_t \left(\frac{\partial U_t^{-\rho} U_{c,t}}{\partial c_t(g^t)} s_t + \frac{\partial s_t}{\partial c_t} U_t^{-\rho} U_{c,t} \right) +$$
$$\frac{\partial U_t^{-\rho} U_{c,t}}{\partial c_t(g^t)} \sum_{i=1}^{N} \left(\mu_{t-i} \left(\frac{V_t}{\mathcal{ANN}^i_1} \right)^{\rho-\gamma} - \mu_{t-i+1} \left(\frac{V_t}{\mathcal{ANN}^{i-1}_1} \right)^{\rho-\gamma} \right) b^i_{t-i+1} + \lambda^V_t V_t^{-\rho} (1-\beta) U_t^{-\rho} \frac{\partial U_t}{c_t(g^t)} = 0$$

iii. $$\sum_{i=1}^{N} \beta^{i-1} b^i_t \mathcal{ANN}^i_4 = s_t U_c^{-\rho} U_{c,t} + \sum_{i=1}^{N} \beta^i b^i_{t+1} \mathcal{ANN}^i_2$$

Where:

$$\frac{\partial \mathcal{M}_{t-i}(V_t)}{\partial V_t} = (\rho - \gamma) \frac{\left(\frac{V_t}{\mathcal{ANN}^i_1} \right)^{\rho-\gamma}}{V_t} \left[1 - \left(\frac{V_t}{\mathcal{ANN}^i_1} \right)^{1-\gamma} f_{g_t}(g_t | g_{t-i}) \right]$$
$$V_t = [(1-\beta) U(c_t, 1-c_t-g_t)^{1-\rho} + \beta \mathcal{ANN}^1_1(\mathcal{I}_{t+1})^{1-\rho}]^{\frac{1}{1-\rho}}$$

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.

Solution with 2 bonds and EZ preferences

Bhandari et al. (2019) convincingly demonstrate that the optimal debt portfolio is no longer that volatile when the model matches asset pricing moments. In this section we change preferences to Epstein-Zin and add a shock that is orthogonal to the government expenditure process, similar to Bhandari et al. (2019). In particular we add an endowment shock z_t, such that $l_t = z_t - h_t$ [35]. Figure 14 shows that in this setting the optimal portfolio does not hold any short position, the allocation shares are around equal among different maturities and little portfolio re-balancing happens in response to government shocks. The intuition is that the presence of a shock orthogonal to government expenditure makes it risky to hold a highly leveraged position and this risk is magnified by Epstein-Zin preferences. Bhandari et al. (2019) solve a similar model using a perturbation method around current level of government debt. Our results using a global solution method are consistent with their intuition.

[35] z_t is independent of g_t and follows an AR1 process $z_t = \mu_z \rho_z z_{t-1} + \epsilon_t^z$ with $\mu_z = 0.9$, $\rho_z = 0.1$, $\epsilon \sim N(0, 0.003)$

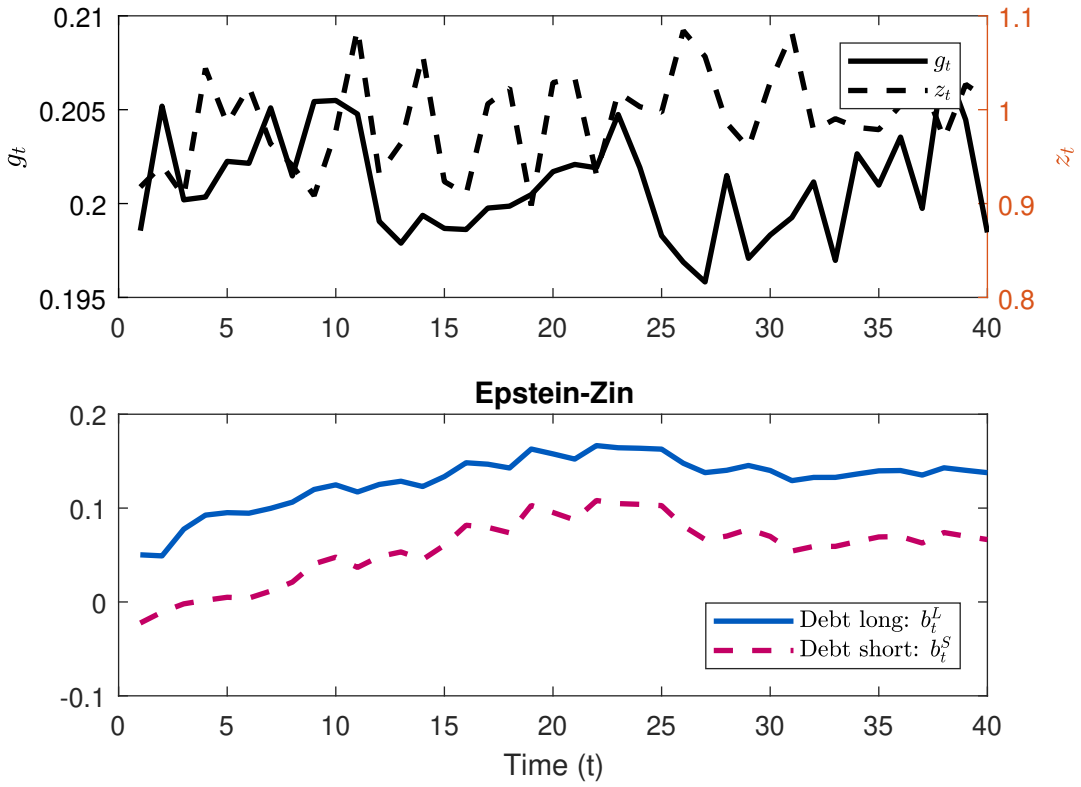

Figure 14: Simulated series with 2 bonds and Epstein-Zin preferences. Top panel: solid line - exogenous sequence of government expenditure, dashed line - exogenous sequence of TFP. Bottom panel: optimal sequence of government debt. Solid line - long bond, dashed line - short bond

There might be multiple interesting ways to match the data, for example Bhandari et al. (2019) 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.

Parameter	Value
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 11: Parameter Values used in the model with Epstein-Zin preferences

Appendix B: Optimal Fiscal policy with CRRA preferences

Parameter	Value
Discount factor	$\beta = 0.96$
RRA	$\gamma = 1.5$
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 12: Parameter Values used in both models one and two bond models

	ANN							
Projected term	$E_t[u_{c,t+S}]$	$E_t[u_{c,t+M}]$	$E_t[u_{c,t+N}]$	$E_t[u_{c,t+S}\mu_{t+1}]$	$E_t[u_{c,t+M}\mu_{t+1}]$	$E_t[u_{c,t+N}\mu_{t+1}]$	$E_t[u_{c,t+N-1}]$	$E_t[u_{c,t+M-1}]$
Residual	0.013	0.019	0.019	0.003	0.004	0.004	0.0196	0.0177
$Residual_{\%}$	0.22 %	0.33 %	0.34 %	0.37 %	0.43 %	0.42 %	0.35 %	0.31 %
Time	56.18min							

Table 13: ANN prediction errors in the 3 bonds model with CRRA preferences. $Residual = \frac{1}{T}\sum\left|Y_i - \hat{E}_i\right|$ and $Residual_{\%} = \frac{1}{T}\sum\left|\frac{Y_i - \hat{E}_i}{Y_i}\right|$, where Y_i is the realized sequence and $\hat{E}_i$ - predicted sequence, simulation length T=1000, bond bounds at 0.5

	ANN							
Projected term	$E_t[u_{c,t+S}]$	$E_t[u_{c,t+M}]$	$E_t[u_{c,t+N}]$	$E_t[u_{c,t+S}\mu_{t+1}]$	$E_t[u_{c,t+M}\mu_{t+1}]$	$E_t[u_{c,t+N}\mu_{t+1}]$	$E_t[u_{c,t+N-1}]$	$E_t[u_{c,t+M-1}]$
Residual	0.0164	0.0218	0.0223	0.0062	0.0062	0.0059	0.0222	0.0213
$Residual_{\%}$	0.29 %	0.38 %	0.39 %	0.70 %	0.70 %	0.67 %	0.39 %	0.37 %

Table 14: ANN prediction errors in the 3 bonds model with CRRA preferences. $Residual = \frac{1}{T}\sum\left|Y_i - \hat{E}_i\right|$ and $Residual_{\%} = \frac{1}{T}\sum\left|\frac{Y_i - \hat{E}_i}{Y_i}\right|$, where Y_i is the realized sequence and $\hat{E}_i$ - predicted sequence, simulation length T=1000, bond bounds at 0.2

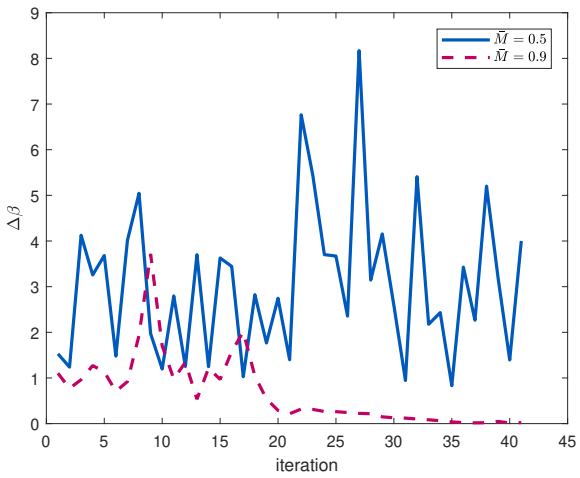

Figure 15: Maximum difference of $\beta(\eta)$ in last 120 iterations. Solid line: bound set at 0.5, dashed line: bound set at 0.9

Appendix C: Method applied to the Stochastic Neoclassical Growth Model

For illustrative purposes we solve a simple neoclassical growth model using our ANN-based Expectations Algorithm and show that the solution coincides with the one calculated through a standard projection method. The demand side of the model is populated by a representative household that maximizes expected lifetime utility:

$$\max_{c} \quad \mathbb{E} \sum_{t=0}^{\infty} \beta^t \ln(c_t)$$

Subject to a budget constraint:

$$k_t(1 + r_t - \delta) + w_t N_t = c_t + k_{t+1}$$

Where k is capital, r is the rental rate of capital, δ is the depreciation rate, w is wage, N is labor and c is consumption. A representative firm produces output using a Cobb-Douglas technology and rents capital and labor from households:

$$\max_{N_t, K_t} \quad z_t K_t^{\alpha} N_t^{1-\alpha} - r_t K_t - w_t N_t$$

where z_t follows an AR(1) process. Taking the optimality conditions and imposing market clearing gives the Euler equation and the resource constraint:

$$\frac{1}{c_t} = \beta \mathbb{E}\left[\frac{1}{c_{t+1}} z_{t+1} K_{t+1}^{\alpha-1} + 1 - \delta\right]$$
$$c_t + K_{t+1} - (1-\delta)K_t = z_t K_t^{\alpha}$$

We solve the model using the Euler equation with three different methods. First we use projection of the consumption policy function approximated as a third degree Chebyshev polynomial and solve for the coefficients that minimize the Euler equation residuals on a grid for capital and technology. Second and third, we use PEA and the ANN-based Expectations Algorithm to approximate the expectation term in the Euler equation and solve using stochastic simulation. The pseudo-code of the algorithm used in this section can be found in Appendix A. The following steps provides a highlight description of the algorithm:

1. Approximate the expectation term contained in the Euler Equation with ANN, as function of k_t and z_t. Given an initial guess of the ANN weights:

   - *Stochastic simulation phase:* simulate the model in time solving, for every t, for c_t and K_{t+1} given the ANN and the current state $\{K_z, z_t\}$.
   - *Training phase:*
     - *Forward phase:* Feed the ANN with the simulated path for K_t and the exogenous process z_t to generate a sequence of prediction for the expectation term.
     - *Backward phase:* Use the error between the sequence of the predicted expectation term and $\frac{1}{c_{t+1}} z_{t+1} K_{t+1}^{\alpha-1} + 1 - \delta$ to update the weights of the ANN.

2. Iterate and stop when the prediction matches the simulated data.

More details about the ANN we used can be found in table **??** in Appendix. An in-depth explanation of what an ANN is can be found in section 3.2.2. In this example we use a simple single layer ANN with 5 neurons and use Levenberg-Marquardt as a training algorithm [36]. Figure 16 shows the simulated solution for consumption and capital for each method used. Given the same initial guess, all three methods converge to the same solution. It is known that stochastic simulation only converges to a rational expectations equilibrium given a right guess. This exercise confirms that if stochastic simulation converges using a polynomial, it also does with an ANN.

[36] We use the following model parameters: $\beta = 0.95$, $\alpha = 0.36$, $\delta = 0.1$, z_t follows a log AR(1) process: $log(z_t) = \rho log(z_t) + \epsilon_t$, where $\rho = 0.8$ and $\sigma^{\epsilon} = 0.25$, simulation length $T = 500$

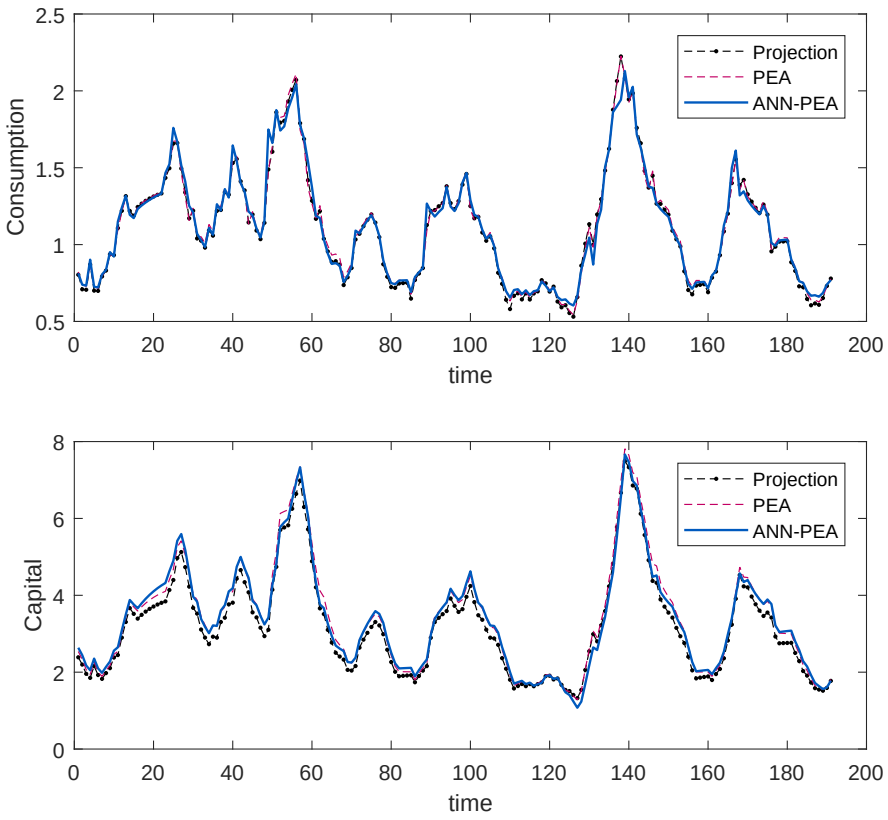

Figure 16: Top panel: simulated consumption path under three solution methods. Bottom panel: simulation capital path under three solution methods. Solid line - ANN based PEA, dashed - PEA with polynomials, dash-dot line - projection

Figure 17 shows the convergence of the ANN-based Expectations Algorithm. The left panel reports the prediction mean-squared errors on the expectation term. Note that, in the spirit of the PEA, we are comparing the ANN predictions (which are for the expectation term) with the actual realization (the argument of the expectation), therefore the errors do not necessarily converge to 0 but just stabilize. The right panel shows the norm of the difference of the ANN weights between consecutive iterations [37]. Both these measures show that the ANN converges in few iterations and that it does not oscillate along the convergence path.

[37]More information on ANN weights are in section 3.2.2.

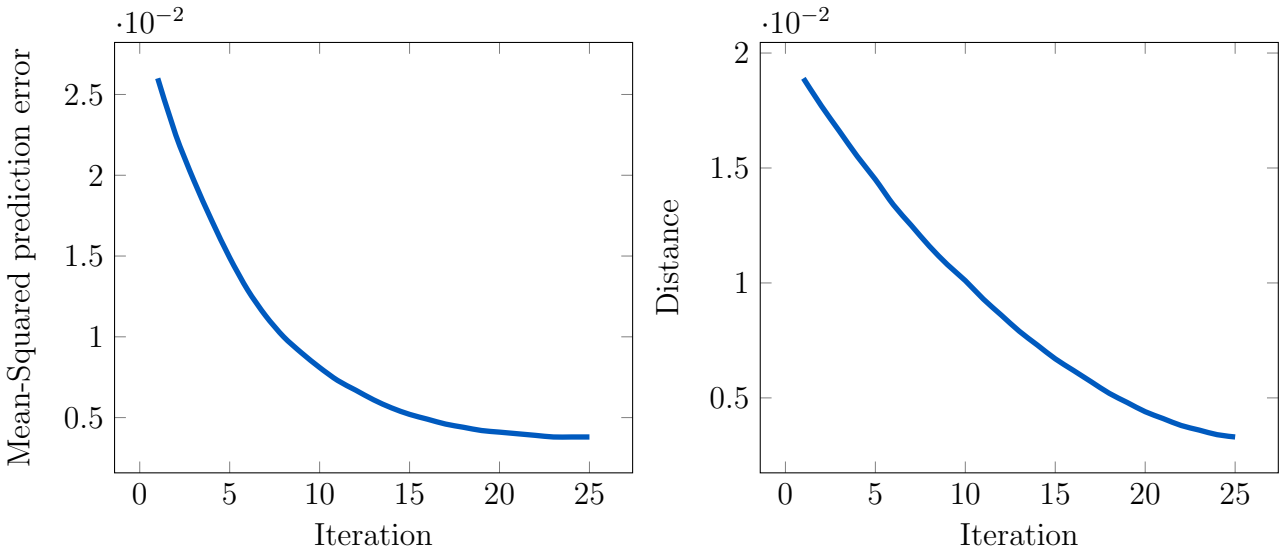

Figure 17: Network convergence, left: mean-squares ANN prediction error, right: distance between parameters of the ANN before and after training measured as the Euclidian norm

Pseudo-code of the illustrative example

Initial guess for weights of the $\mathcal{ANN}$
repeat
  //Stochastic simulation phase
  for $t = 1$ to T (large) **do**
    $\mathbb{E}_t^* = PredictExpectations(K_t, z_t, \mathcal{ANN})$
    $c_t = (\beta \mathbb{E}_t^*)^{-1}$
    $K_{t+1} = z_t K_t^{\alpha} + (1 - \delta) K_t - c_t$
  end for
  Initialize $\mathcal{ANN}$ weights
  //Training phase (equivalent to the regression in PEA)
  repeat
    for $t = 1$ to T **do**
      //Forward Pass
      $\mathbb{E}_t^* = PredictExpectations(K_t, z_t, \mathcal{ANN})$
      $Err_t = \frac{1}{c_{t+1}} z_{t+1} K_{t+1}^{\alpha - 1} + 1 - \delta - \mathbb{E}_t^*$
      //Backward Pass
      Calculate $\frac{\partial Err_t}{\partial w_{ij}}$
      $w_{ij} = w_{ij} - \alpha \frac{\partial Err_t}{\partial w_{ij}}$
    end for
  until Validation set error stops improving
until $\mathcal{ANN}$ weights keep changing

Appendix D - highlight description of ANN

In general, an ANN is composed by three types of layers: input, hidden and output. In our applications, the input layer takes as input the state variables of the model and normalizes them. The output layer outputs the expectation terms contained in the model optimality condition. The hidden layer contains an exogenous number of neurons. Intuitively, it represents an intermediate transformation of the state space.

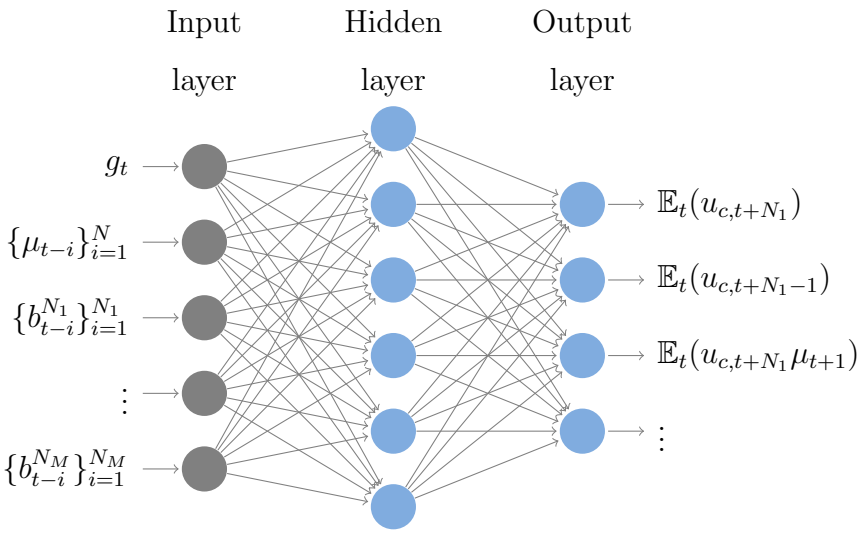

Figure 18: Structure of a one-layer ANN. Each circle in the picture represents an artificial neuron, the arrows point the direction of information flow in the prediction process. Neurons in the hidden layer perform the non linear activation of inputs, which are combined linearly in the output layer

The structure of a one-layer ANN is reported in figure 18, which represents the interconnection of artificial neurons. Each node in the picture represents an artificial neuron and its structure is reported in figure 19. Each neuron receives inputs and processes them.

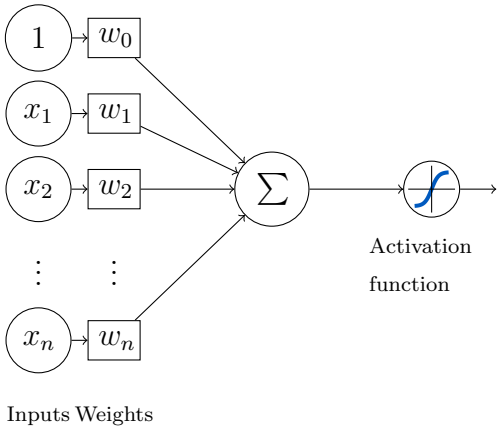

Figure 19: Structure of an artificial neuron: first inputs are combined linearly using input weights, then the linear combination is passed through a non linear activation function

Each input $(1, x_1, ..., x_n)$ (outputs of a previous layer of artificial neurons as shown in figure 19) is multiplied by a specific weight$(w_0, ..., w_n)$. The results of these multiplications are added and, if the sum exceeds a certain threshold, the neuron activates by activating its output according to an activation function:

$$\text{Neuron Output} = \frac{2}{1 + \exp\left(-2\sum_{i=0}^{n} w_i x_i\right)} - 1$$

The weight quantifies the importance of the input. 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.

Training and validation phase In general, the entire available data set is divided into two groups: training set and validation set. The training set is used for the training phase, when the ANN weights are adjusted in order to match the data in the training set. 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 training 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.