Approximating High-Dimensional Economic State-Spaces with Deep Neural Embeddings

Emmet Hall-Hoffarth

November 5, 2021

Abstract

Placeholder

1 Introduction

Ever since the seminal work of Krusell and Smith (1998) there has been an explosion of interest in stochastic macroeconomic models that represent the behaviour of heterogeneous agents. However, there is a fundamental challenge associated with solving these models known as the *curse of dimensionality* (Dreyfus & Bellman, 1962), that computational solutions to optimization problems quickly become intractable as the state-space increases in size. This problem arises because the optimal decision of any rational, forward-looking agent in a non-trivial heterogeneous agent economy depends, at least in principle, on the state(s) of every other agent. In their paper Krusell and Smith (1998) overcome this by showing in their setting agents are sufficiently rational (they make sufficiently accurate predictions of future states) if their perceived state-space consists only of their own asset holdings and the mean aggregate asset holdings in the economy. However, this property is unlikely to hold as the literature begins to consider increasingly complex models (Krusell & Smith, 2006), and therefore, new techniques will be needed.

More recently, papers by Duarte (2018) and Azinovic et al. (2019) have applied machine learning techniques in order to assuage some of these concerns. In particular, in the case of the former, the (continuous time) value function is parameterized as a neural network. Since (deep) neural networks are particularly well suited to high-dimensional and highly non-linear problems, they provide an excellent functional approximation of the value function that can then be estimated via stochastic gradient descent. Furthermore, since this is still a full parameterization, it avoids some of the limitations involved with projection-based methods. This line of reasoning is suggestive of the potential for other machine learning techniques for solving macroeconomic models with high-dimensional state-spaces. In particular, this paper will consider the introduction of (neural) embeddings similar to those found in Bengio et al. (2003).

In machine-learning embeddings are generally used for word or image processing, however, more broadly they are useful for tackling high-dimensional, yet sparse, input data. They do so by mapping the high-dimensional input data into a (much) lower dimensional and dense *latent space*. One popular implementation, as in Bengio et al. (2003), is to linearly map into the latent space with an *embedding matrix*, and then optionally feed the output through a neural network to handle potential non-linearities in the output signal. Thus, embedding models are well suited for solving heterogeneous agent stochastic macroeconomic models which have both high-dimensional and sparse state-spaces, and potentially strongly non-linear responses.

This paper is organized as follows: Section 2 will cover contextual information on both heterogeneous agent models as well as, (deep) neural embeddings. Section 3 will introduce embedding models as a solution method to economic models in a general setting. Section 4 will introduce a particular economic model which will be estimated using this technique. The final two sections will present the results of this application and give some remarks.

2 Literature Review

2.1 Machine Learning

2.1.1 Embeddings

2.1.2 Neural Networks

2.1.3 SGD and Cross-Validation

2.2 Heterogenous Agent Models

3 Method

The goal is to approximate the (Bellman) value function

$$v(X) = \max_{C} u(C) + \beta v(X')s.t. \tag{1}$$

$$X' = h(C, X) \tag{2}$$

where $C \in \mathbb{R}^p$ is a set of controls and $X \in \mathbb{R}^n$ is a set of state variables in a finite, yet high dimensional setting where exact solution methods are infeasible. In order to do so we will reduce the dimension of X using an embedding matrix and make a (flexible) parametric assumption about $v(.)$. The former is achieved by introducing a parameter matrix $A \in \mathbb{R}^{m \times n}$ such that $\widetilde{X} \equiv AX$ where $m < n$. The latter can be achieved in a number of ways, as this is a common component of simulation based solution methods. For example, the approach of Den Haan (1996) is to use an n-th order polynomial approximation. Instead, here we will follow the approach of Duarte (2018) and approximate the value function with

a dense feed-forward neural network. In particular, we approximate $v(.)$ as $\hat{v}(.)$ where

$$v(X) \approx \hat{v}(X;\theta) = f(g_k(g_{k-1}(...(g_0(AX))))) \tag{3}$$
$$f(x) = w_k x + b_k \tag{4}$$
$$g_i(x) = \max\{0, w_i x + b_i\} \tag{5}$$
$$w_i \in \mathbb{R}^m \tag{6}$$
$$b_i \in \mathbb{R} \tag{7}$$

where $\theta \in \Theta \equiv \mathbb{R}^{(n+k)m+k}$ refers to the set of all parameters A, w_i, and b_i. As a result of this parameterization there are a total of $(n+k)m+k$ parameters; $m \times n$ from the (unrestricted) matrix A and $k(m+1)$ from the parameters w_i and b_i of the neural network. An advantage of this embedding-based approach in comparison to Duarte (2018) is already clear — since m is generally chosen to be small relative to n, the neural network here has many fewer parameters for a given level of expressiveness (depth) k. Nonetheless, overall there is still a large number of parameters here, so as in the majority of the machine-learning literature, they will be estimated using Stochastic Gradient Descent (SGD)

In particular, estimation proceeds as follows. Since the Bellman equation is a contraction mapping, its left-hand and right-hand sides are equal at the solution, so any reasonable approximation should also possess this property. Therefore, define the (approximate) Bellman error as:

$$\epsilon^2 = (\hat{v}(X;\theta) - (\max_{C,X'} u(C) + \beta \hat{v}(X';\theta)))^2 \tag{8}$$

The goal is to minimize this error, and ideally reduce it to zero. Let C^*, X'^* be the optimal control in the intratemporal optimization problem, and resulting future state respectively. Then, since we have fully parameterized $\hat{v}$, the Jaccoibian $J_{\epsilon^2,\theta}$ is defined. Indeed, by application of chain rule it is exactly:

$$J_{\epsilon^2,\theta} = -2(\hat{v}_\theta(X;\theta) - \beta \hat{v}_\theta(X'^*;\theta)) \tag{9}$$

Where

$$\begin{aligned} \hat{v}_\theta(Z;\theta) = & f'(g_k(g_{k-1}(\dots(g_0(AZ))))) \times \\ & f(g_k'(g_{k-1}(\dots(g_0(AZ))))) \times \\ & \dots \\ & f(g_k(g_{k-1}(\dots(g_0'(AZ))))) \times \\ & f(g_k(g_{k-1}(\dots(g_0(AZ)))))Z \end{aligned} \tag{10}$$

For $Z \in X, X'^*$

This object, cumbersome as it may seem, can be calculated quickly and efficiently with modern machine learning software using backpropogation. Given that this gradient can be calculated, we can estimate the optimal parameters

by SGD. Given some initial guess θ_0, we iteratively update successive guesses of θ_t according to:

$$\theta_{t+1} = \theta_t - \delta \frac{1}{n} \sum_{i=1}^{n} J_{\epsilon_i^2,\theta} \tag{11}$$

where δ is the *learning rate*, and $i \in 1, ..., n$ are a random subset of the indicies of X, until the gradient $J_{\epsilon_i^2,\theta}$ is below some acceptance threshold. This is the simplest form of SGD, however, in practice more sophisticated algorithms have been developed which take into account, among other things, the previous gradients, higher order derivatives, and the number of iterations undertaken so far, in an effort to optimally adjust the learning rate for each iteration. The application in this paper will make use of one such algorithm, known as the Adam optimizer (Kingma & Ba, 2014).

4 Application

5 Results

6 Conclusion

References

Azinovic, M., Gaegauf, L., & Scheidegger, S. (2019). Deep equilibrium nets. *Available at SSRN 3393482*.

Bengio, Y., Ducharme, R., Vincent, P., & Janvin, C. (2003). A neural probabilistic language model. *The journal of machine learning research*, *3*, 1137–1155.

Den Haan, W. J. (1996). Heterogeneity, aggregate uncertainty, and the short-term interest rate. *Journal of Business & Economic Statistics*, *14*(4), 399–411.

Dreyfus, S. E., & Bellman, R. (1962). *Applied dynamic programming*. Princeton University Press.

Duarte, V. (2018). Machine learning for continuous-time economics. *Available at SSRN 3012602*.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krusell, P., & Smith, A. A. (2006). Quantitative macroeconomic models with heterogeneous agents. *Econometric Society Monographs*, *41*, 298.

Krusell, P., & Smith, A. A., Jr. (1998). Income and wealth heterogeneity in the macroeconomy. *Journal of political Economy*, *106*(5), 867–896.