Solving Dynamic Portfolio and Consumption Problems by Going Forward in Time

Yixuan Ma

Finance Department, Maastricht University, Maastricht, Netherlands, 6211 LM,
(yixuan.ma@maastrichtuniversity.nl)

Paulo Rodrigues

Finance Department, Maastricht University, Maastricht, Netherlands, 6211 LM,
(p.rodrigues@maastrichtuniversity.nl)

Peter Schotman

Finance Department, Maastricht University, Maastricht, Netherlands, 6211 LM,
(p.schotman@maastrichtuniversity.nl)

Preliminary and Incomplete

Abstract The standard approach to solving dynamic portfolio and consumption problems numerically uses backward induction, which complicates the solution if decisions at time t depend on past decisions. In contrast, our solution algorithm goes forward in time. We use the insight that the main task in solving dynamic optimization problems consists of finding policy functions that use the current value of state variables as inputs and give the optimal decisions as outputs. Instead of assuming a functional form for these policy functions, we use a neural network for the estimation of the functions.

Key Words Optimal Portfolio, Dynamic Optimization, Neural Networks, Loss Aversion, Consumption Choice, Lifecycle Model

JEL Classifications:

Solving Dynamic Portfolio and Consumption Problems by Going Forward in Time

Preliminary and Incomplete

Abstract The standard approach to solving dynamic portfolio and consumption problems numerically uses backward induction, which complicates the solution if decisions at time t depend on past decisions. In contrast, our solution algorithm goes forward in time. We use the insight that the main task in solving dynamic optimization problems consists of finding policy functions that use the current value of state variables as inputs and give the optimal decisions as outputs. Instead of assuming a functional form for these policy functions, we use a neural network for the estimation of the functions.

Key Words Optimal Portfolio, Dynamic Optimization, Neural Networks, Loss Aversion, Consumption Choice, Lifecycle Model

JEL Classifications:

1. INTRODUCTION

The principal method for solving intertemporal savings and portfolio selection problems is numerical dynamic programming. As an alternative we explore a neural network approach. Our neural net approach has advantages for problems where the backward dynamic programming recursion is numerically cumbersome. Examples in the life-cycle literature are problems with path dependent decisions, such as saving in a separate retirement account, buying or renting a house, preferences with habits or loss aversion, taxes, and timing of retirement. What these problems have in common is that the solution method requires endogenous state variables, which values are not known while moving backward in time. They need to be solved 'on a grid' of potential values, with the relevant point on the grid only known at the very end of the recursion. With one or two endogenous state variables this is still feasible, but in higher dimensions this becomes numerically awkward.

Arguments for exploring a forward solution method, as opposed to the backward recursion of dynamic programming, are not new. Much of our motivation coincides with an early paper by Chacko, Desai, Golts, and Novakovsky (2005). They approximate the optimal policies for consumption and portfolio weights by a low order polynomial in the state variables. They then simulate the policies using a large set of scenarios and determine the optimal parameters for the polynomial solutions. Our approach replaces the polynomial functions by a very flexible neural network model. The main advantage of the neural network is its flexibility. The cost of the full flexibility is a heavily overparameterised model that requires regularisation.

The neural net is computationally very efficient and can quickly learn a reasonably good policy, albeit not the true optimal one. Using standard activation functions on the output variables it can also easily handle natural constraints such as non-negative consumption and portfolio weights.

The basic building block of the neural net approach is a set of scenarios. In a life-cycle model these scenarios can be long, but are of finite length. Working with a set simulated scenarios for returns and macro-economic state variables is also the starting point of the most successful backward solution methods, such as for example Brandt, Goyal, Santa-Clara, and Stroud (2005). The difference, of course, is that they develop an approximate backward solution method where the cross-section of scenarios is used to approximate the expected utility at every decision node in the scenario tree.

Apart from Chacko et al. (2005) we are not aware of other papers that directly optimise the policy function in a life-cycle model. In infinite horizon models, Judd (1998) discusses general projection methods for policy function iterations. Chen, Cosimano, and Himonas (2014) provide a literature review with a similar focus on the infinite horizon problem.

The current paper explores the performance of the numerical optimization routine. We start with a few simple models. These examples serve to explain the method. Since the solution in the simple examples is also available in closed form, we can compare the solutions. The main part of the paper delves into a more challenging problem for which there does not exist an easy alternative solution.

Much of recent research has stressed that individuals derive utility from comparing their current payoffs relative to some benchmark such as a habit. We analyse a model with loss aversion, where the benchmark is a function of current wealth. These models have been investigated for long-term investors by Berkelaar, Kouwenberg, and Post (2004) and Van Bilsen, Laeven, and Nijman (2020). Both papers derive closed-form solutions for long-term investors, but these solutions involve highly leveraged positions in stocks. Imposing leverage and short-sell constraints requires a numerical solution, which is somewhat cumbersome, because both actual wealth and the benchmark change over time, which requires a two-dimensional grid of state variables. This is the type of models where the neural net approach will be most beneficial.

2. NEURAL NETWORK POLICY FUNCTION

To explain the method we start with a standard multi-period investment problem. At time 0, the start of the planning period, the investor's indirect utility function is

$$J(W_0, z_0) = \max_X \mathrm{E}_0 \left[U(W_T) \right], \tag{1}$$

where $X = (x_0, \ldots, x_{T-1})$ is a sequence of M-vectors of portfolio weights. The portfolio generates a return $x_t' R_{t+1}$ with R_{t+1} the M-vector of gross returns on different assets. Arguments in the value function $J(W_0, z_0)$ are initial wealth W_0 and a K-vector of state variables z_0. Elements in z include the returns R_t plus predictors of future returns. The final element in z is the remaining time until the end of the planning horizon, $\tau_t = T - t$. The preferences $U(W)$ are a well-behaved utility function in final wealth ($U'(W) > 0$, $U''(W) < 0$). In later sections we extend the model to include more complicated preferences and intermediate consumption. Portfolio problem (1) is fairly standard. In continuous time, assuming CRRA utility, this is the Merton model. In our discrete time setup it can be solved by other methods as well, for example the backward dynamic programming algorithm in Brandt et al. (2005) or the log-linear approximation in Jurek and Viceira (2011a). We use these solutions to benchmark the performance of our NN algorithm.

The goal is to find a decision rule

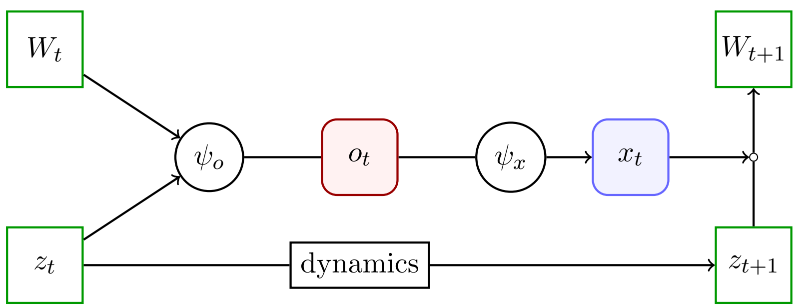
$$x_t = h(W_t, z_t) \tag{2}$$

that solves the maximisation problem (1) subject to the wealth constraint

$$\frac{W_{t+1}}{W_t} = x_t' R_{t+1} \tag{3}$$

The optimal decision rule is a function of the current values of the state variables. For computational feasibility, we wish to have the same function $h(W, z)$ for all periods, instead of requiring separate functions $h_t(z_t, W_t)$ for each period. Two design features are important to obtain time-invariant solution functions. First, we account for the horizon dependence in the decision rule by including τ_t in the inputs.

Second, we generate scenarios for the state variables that randomize the initial condition z_0. In practice, we simulate one very long sequence of $N(T+1)$ state variables (including the returns), and cut this long time series into N non-overlapping scenarios of length $T+1$.[1] Assuming that the states are generated by a stationary dynamic process, the procedure results in initial conditions that can be regarded draws from the unconditional distribution of z. An important assumption underlying (2) is that the decisions only depend on the current state variable and not on the entire history $\{z_0, \ldots, z_t\}$. This is true for the life-cycle model with time-separable utility, but need not be true in other models. Due to the time-invariant solution functions we achieve a huge dimension reduction that will make it feasible to train models with a moderately long horizon using a limited number of scenario paths. Although the solution function is time-invariant, decisions will vary over time according to the evolution of the state variable z_t, among them the time until maturity. In each scenario we always set wealth at the same initial value W_0.



The graph shows the structure of the recursive neural network containing inputs (W_t, z_t), a hidden layer with neurons in the vector o_t, and a decision layer for portfolio weights. The activation function ψ_o transforms the inputs into the hidden layer outputs. The decision layer uses activation function ψ_x. State variables evolve exogenously; wealth endogenously, depending on the decisions for x plus the realised returns that are part of z_{t+1}.

Figure 1: Network architecture life-cycle model

We train a neural network that accepts as inputs starting wealth W_0, a sequence of states z_t $(t = 0, \ldots, T-1)$, and a sequence of return vectors R_t $(t = 1, \ldots, T)$. The network returns the portfolio weights $x_t = h(W_t, z_t)$. We set up two layers that differ by the activation function and number of nodes. Figure 1 provides a graphical representation of the network architecture. The first layer is a hidden layer containing a flexible number of L neurons in the vector o_t. For the second layer, the decision layer, we fix the number of neurons to be equal to the number of assets. The output from this layer are the portfolio weights. The final layer provides the intertemporal connection for the evolution of wealth using the portfolio weights together with the returns. The neural network recursively calculates final wealth W_T. The architecture emphasises the difference between the state variables W_t and z_t. W_t is determined endogenously as a result of the decision

[1] A scenario has length $T+1$ dated $t = 0, \ldots, T$.

variable x_t, while z_t is evolves exogenously.

In equations the network recursively processes the inputs as

$$o_t = \psi_o(a_o + A_{oz} z_t + A_{oW} W_t) \tag{4}$$

$$x_t = \psi_x(a_x + A_{xo} o_t) \tag{5}$$

with ψ_o and ψ_x activation functions. The vectors a_o, a_x and A_{oW}, and the matrices A_{oz} and A_{xo}, are parameters that are constant across all t, such that the output just varies with the input values. As activation functions we use 'ReLu' for ψ_o and 'softmax' for ψ_x. The 'softmax' function ensures that all portfolio weights are between zero and one and sum up to one.

Wealth at maturity is the final result that goes into the utility function, $U(W_T)$, and defines the loss on which the network will be trained. With N scenario paths for the state variables we obtain N values $W_{T,j}$ of final wealth. The empirical loss function is the average loss over the N simulated samples using the policy function $x = h(W, z)$,

$$\text{LOSS} = -\frac{1}{N} \sum_{j=1}^{N} U(W_{T,j}) \tag{6}$$

After training the network we use the following output as our policy function

$$h(z, W) = (\psi_x \circ \psi_o)(W, z) \tag{7}$$

Along a specific scenario path j we then have the portfolio weights $x_{t,j} = h(z_{t,j}, W_{t,j})$ for all periods $t < T$. Following wealth along the optimal paths we obtain the value function as the negative of (6).

In various cases we have theoretical results about the structure of the solution. For example, with CRRA utility wealth does not enter the portfolio solution. In addition, when returns have time-invariant mean and covariance matrix, the solution does not depend on the investment horizon. For other utility functions, for example SAHARA (Chen, Pelsser, and Vellekoop 2011), the optimal portfolio depends on wealth. In the large literature on strategic asset allocation (Campbell and Viceira 2002) predictability of returns has a large effect on the optimal allocation to equity investments.

The solution is written as the unconditional function $x_t = h(z_t, W_t)$, so we can obtain optimal portfolio for any value of the state variables. Accuracy of the solution obviously depends on whether the chose point $(\tilde{z}, \tilde{W})$ is located relative to the training data the network has seen.

When the solution depends on wealth, numerical backward dynamic programming solutions become more cumbersome, as they must be solved 'on a grid'. At each time step the solution must be computed for a range of values for wealth. More challenging from an optimisation perspective are preferences that involve more than just wealth. One example is Loss Aversion. In Berkelaar et al. (2004) the utility depends on final wealth relative to a benchmark $\bar{W}_T$, which itself may change over time depending on past returns. We then have two endogenous state variables and would require a bivariate grid in a backward solution. It is here that the neural net method should have its largest computational benefits.

A second extension is intermediate consumption. With time-separable utility the problem can be written as

$$J(W_0, z_0, y_0) = \max_{\{C_t, x_t\}} \mathrm{E}_0 \left[\sum_{t=0}^{T-1} \delta^t U(C_t) + B(W_T) \right], \tag{8}$$

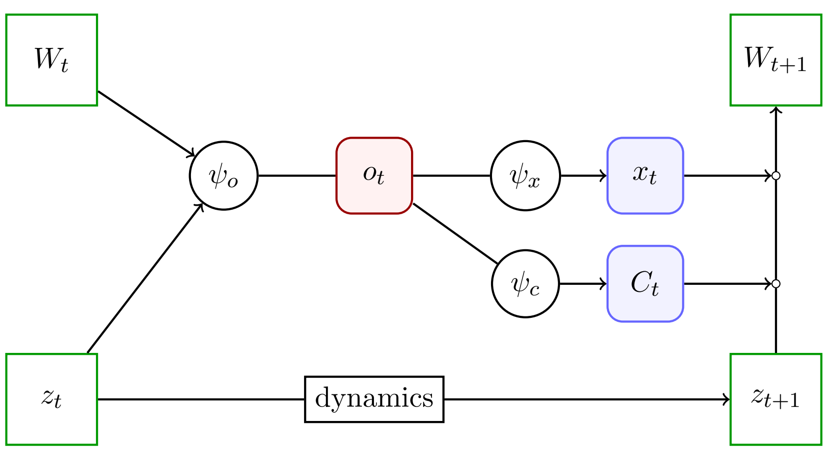
where C_t is consumption, and δ a time preference parameter. The value function has labor income y as an additional argument. For an individual, labour income is partly a function of the endogenous labour supply decision. In the current version of our algorithms we don't take this into account. We similarly leave labour income exogonously given as a function of age and individual characteristics. The goal is to find portfolio and consumption functions

$$\begin{pmatrix} x_t \\ C_t \end{pmatrix} = \begin{pmatrix} h^C(W_t, z_t) \\ h^x(W_t, z_t) \end{pmatrix} = h(W_t, z_t) \tag{9}$$

that solve the maximisation problem subject to the budget constraint

$$W_{t+1} = (W_t - C_t + y_t) x_t' R_{t+1} \tag{10}$$

Figure 2 shows the extended architecture. It has an additional layer for the consumption decision with an activation function ψ_C, which we take as 'tanh' generating a positive consumption / wealth ratio in every state. Given the consumption-wealth ratio c_t, consumption follows as $C_t = c_t W_t$.



The graph shows the structure of the recursive neural network containing inputs (W_t, z_t), a hidden layer with neurons in the vector o_t, and decision layers for consumption and portfolio weights.

Figure 2: recursive NN with consumption

Apart from the additional layer we proceed as before. The network iterates through all periods until maturity and collects all C_t ($t = 0, \ldots, T-1$) together with W_T along all scenario paths. The loss function is the negative of the realised utility,

$$\text{LOSS} = -\frac{1}{N}\sum_{j=1}^{N}\left(\sum_{t=0}^{T-1} U(C_{t,j}) + B(W_{T,j})\right) \tag{11}$$

which the algorithm uses to fit the network parameters.

Summarizing, the neural network uses four key steps:

1. Simulate the economic model to generate a large number of scenarios.
2. A sufficiently flexible layer setup that uses the information generated in the simulations as inputs and returns values of our decision variables.
3. A layer that has no trainable parameters, but takes the decisions and plugs them into the constraints to ensure that the strategies are feasible.
4. The computation of the criterion function in the loss function used.

3. BENCHMARK MODELS: CRRA UTILITY

In this section, we apply our solution method to cases where an approximate closed-form solution exists. For these exercises, we assume that the investor has a CRRA utility function with a risk-aversion parameter of 10 and a subjective discount factor of 0.96. We simulate quarterly returns for 5 years 100,000 times for the training data. The validation dataset contains 30,000 sequences of quarterly returns for 5 years. Finally, we test the solution on 100,000 sequences that again simulate quarterly returns over 5 years.

3.1 Identically and Independently Distributed Returns

We assume that log returns follow the following distribution

$$r_{t+\Delta t} = \ln(S_{t+\Delta t}) - \ln(S_t) \sim \mathrm{N}\left((\mu - \tfrac{1}{2}\sigma^2)\Delta t, \sigma^2 \Delta t\right) \tag{12}$$

with $\mu = 0.1$ and $\sigma = 0.2$. The log risk-free rate is constant at $r_f = 0.05$. The investor has a horizon of 5 years and rebalances every quarter ($\Delta t = 1/4$). With these assumptions, the optimal portfolio weight on the stock is approximately (Campbell 2018)

$$x_t = \frac{\mu - r_f}{\gamma\sigma^2} = 0.1003 \tag{13}$$

The solution is approximate since a portfolio return of a lognormal stock return and a constant risk-free rate is not exactly lognormally distributed.

The first test for the NN methodology is if it will be able to find this very simple constant solution if we feed it with a more general potential solution function $h(z, W)$ with z including irrelevant state variables: the part return r_t, the length of the planning horizon τ_t, as well as the current wealth, none of which should matter in the solution function.

3.1.1 Maximizing Utility of Terminal Wealth

This problem constitutes a low bar for the procedure to clear. As with any numerical procedure based on simulations, the results will contain some variation. For this exercise, we normalize the value of the starting wealth to 1.

We use the policy function found by training the neural network and compare the outcomes to the approximate optimal solution when using the 100,000 sequences of the test dataset. The results are given in Figure 3. The resulting distribution of the realizations of the terminal wealth of each simulated path can be found on the right side of the upper row of Figure 3. We plot the optimal solution with light red bars and the results using the NN method with light blue bars, which results in a light purple bar for the overlapping cases. We see that both distributions are basically on top of each other, implying that there is no economically significant difference between them. The difference in certainty equivalent return using the average realized utility is a fraction of a basis point.

We show the resulting weights on the risky assets in the left panel of the upper row of Figure 3. We can see that there is some variation around the optimal solution for the NN methodology, i.e., the estimated exposures to the irrelevant state variables will not equal exactly zero. However, we see that the average portfolio weight is very close to the optimal solution, and the differences are mostly much smaller than one percentage point. Finally, we show a violin plot of the weights for each quarter in the lower part of Figure 3. We see that the average weight is very close to the optimal solution, which is plotted as the red line. Again, the variation stems from the fact that the weights in the functions are not trained to be exactly zero.

Summarizing, we see that the method performs well in this simple exercise. Note that we did not use any prior knowledge to steer the method to the optimal solution, but included several redundant state variables that increase the amount of learning that needs to take place. One obvious way to use prior knowledge would be to restrict the number of states used in the policy function or not use any states at all. In this case, the only free parameters would be the bias term, resulting in a mechanically constant portfolio policy.

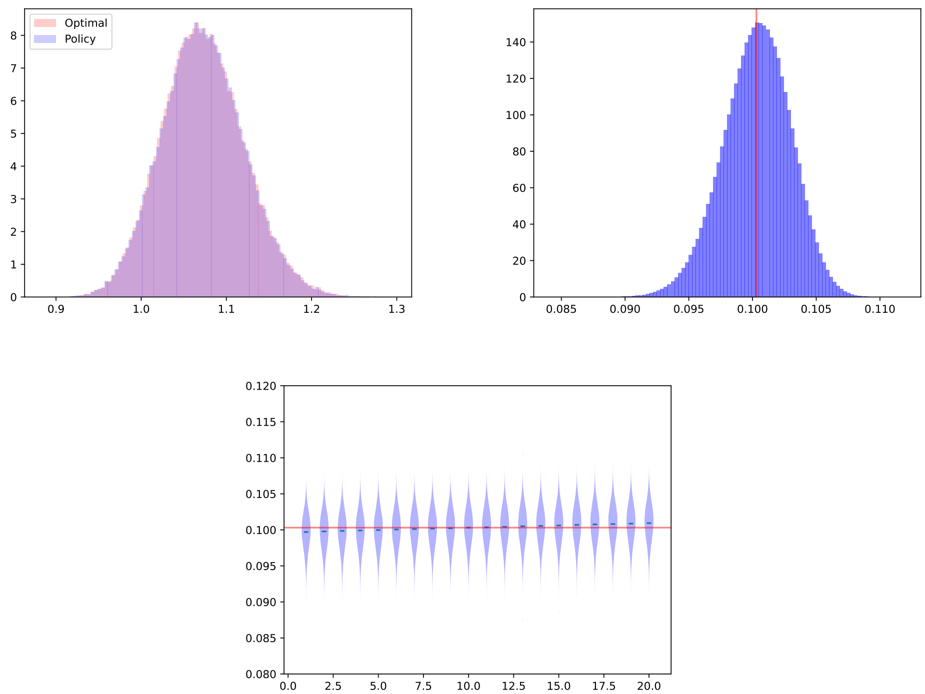


Figure 3: This figure shows the performance of the NN methodology in the case the investor maximizes the utility of terminal wealth and the economy follows an i.i.d. process. The upper row shows the distribution of the terminal wealth for each sequence of the test data when using the optimal solution in light red and when using the policy function in light blue. For the case in which both distributions overlap, the color turns light purple. The right side of the upper row shows the distribution of the weight on the risky asset for the policy function across all time points and sequences of the test data in light blue, the optimal solution is plotted as a red line. The lower row of the figure shows a violin plot for the portfolio weights at each point in time as well as the optimal solution as a red line.

3.1.2 Solving the Consumption and Portfolio Problem

In the case of i.i.d. returns and CRRA utility, we can determine the optimal portfolio weight as[2]

$$E\left(R_{P,t}^{-\gamma} R_t\right) = R_f E\left(R_{P,t}^{-\gamma}\right) \tag{14}$$

with $R_{P,t}$ denoting the portfolio return, which is a function of the weight of the risky asset. We can use the unconditional expectation since the returns are i.i.d. Moreover, the optimal weight is constant. The optimal consumption-wealth ratio at each t is given by

$$c_t = \frac{a_1 c_{t+1}}{(1 + a_1 c_{t+1})} \tag{15}$$

with $c_T = 1$ and $a_1 = \left[\delta E\left(R_{P,t}^{-\gamma}\right)\right]^{-1/\gamma}$. This results in a consumption stream for each t of

$$C_t = c_t W_{t-1} \tag{16}$$

Our policy function is found by using a neural network containing a "relu" activation function and 10 nodes for the hidden layer. The decision layer for the portfolio weights uses a "softmax" and the consumption wealth ratio a "sigmoid" activation function, respectively. As all consumption has to be financed by the portfolio return and the initial wealth level, we normalize the initial wealth to 10 in this exercise.

Figure 4 shows the results for the NN methodology in comparison to the optimal solution. The upper left graph shows the histogram for the realized utility values across all 100,000 test sequences. We see that there is a large overlap between both solutions. In terms of expected utility, the NN methodology returns a value of -971.49 vs. an optimal value of -950.48. To give economic meaning to the difference in expected utility, we compute the certainty equivalent as a constant consumption stream that would result in a utility value equal to the expected utility. The result is a value of 0.508 for the optimum and a value of 0.507 for the NN methodology, which is over 99.7% of the optimum.

The resulting portfolio policies can be seen in the upper right as well as in the lower left graph shown in Figure 4. The graph in the upper right corner depicts the histogram for the portfolio weights on the risky asset over all time points across all sequences in the test data in light blue and the optimal weight as a red line. We see that the NN methodology puts, on average, about 30 basis points more weight into the risky asset than optimal. In addition, we see that the histogram resembles multiple small distributions, which is due to the effect of time to maturity, which can be seen in the lower left-hand graph of the figure. The graph shows the violin plots of the portfolio weights for each quarter across all test sequences as well as the optimal solution as a red line.

[2] See, e.g., Pennacchi (2008).

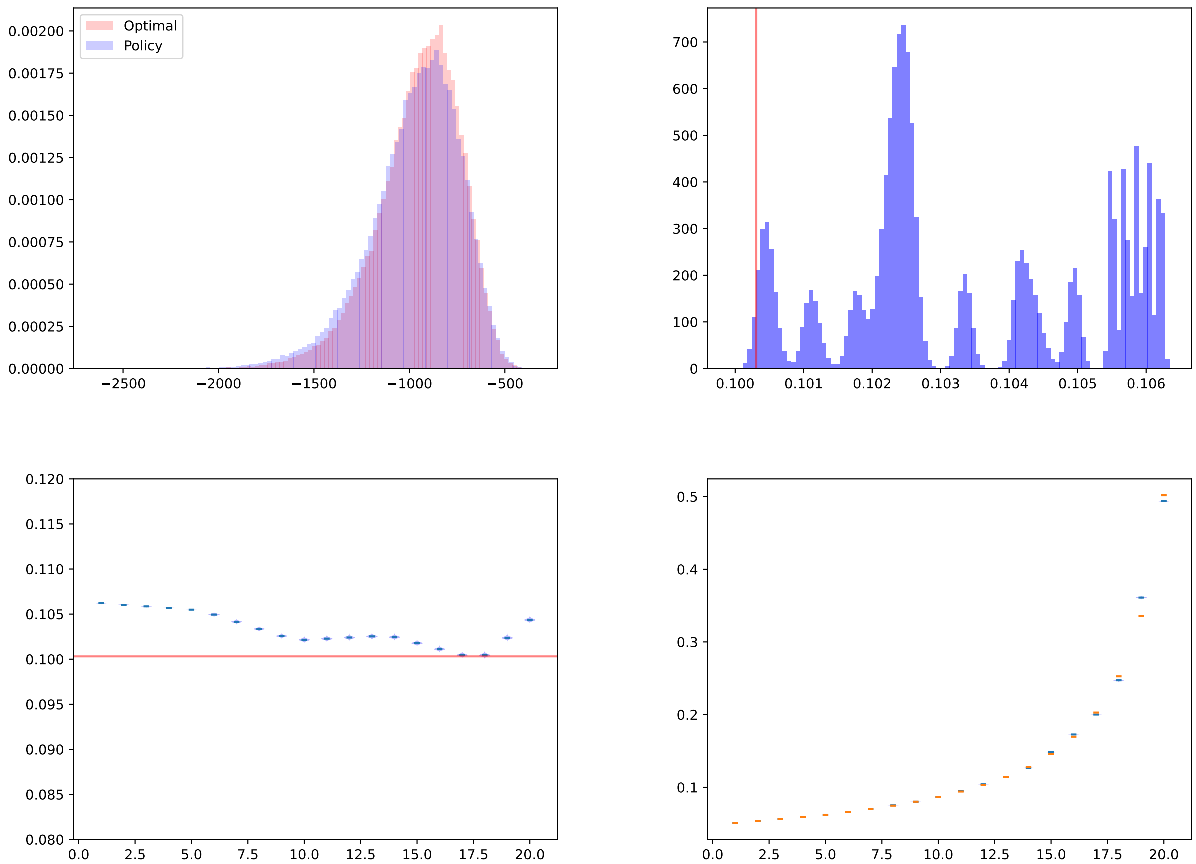


Figure 4: This figure shows the performance of the NN methodology in the case the investor maximizes the utility of terminal wealth and the economy follows an i.i.d. process. The upper row shows the distribution of the realized utility results for each sequence of the test data when using the optimal solution in light red and when using the policy function in light blue. For the case in which both distributions overlap, the color turns light purple. The right side of the upper row shows the distribution of the weight on the risky asset for the policy function across all time points and sequences of the test data in light blue, the optimal solution is plotted as a red line. The left-hand side of the lower row of the figure shows the violin plots for the portfolio weights at each point in time as well as the optimal solution as a red line. On the right-hand side of the lower row, we see the violin plots of the consumption-wealth ratio for each period in blue together with the optimal value depicted in red.

Table 1: This table shows the average values for the portfolio weights and the consumption levels for the optimal solution and the values found by using the policy function of the NN methodology

Period	Optimal		Policy	
	Weight	Consume	Weight	Consume
Period 0	0.100	0.510	0.106	0.509
Period 1	0.100	0.509	0.106	0.510
Period 2	0.100	0.507	0.106	0.509
Period 3	0.100	0.505	0.106	0.506
Period 4	0.100	0.503	0.105	0.502
Period 5	0.100	0.501	0.105	0.501
Period 6	0.100	0.500	0.104	0.502
Period 7	0.100	0.498	0.103	0.500
Period 8	0.100	0.496	0.103	0.496
Period 9	0.100	0.494	0.102	0.494
Period 10	0.100	0.493	0.102	0.497
Period 11	0.100	0.491	0.102	0.494
Period 12	0.100	0.489	0.103	0.487
Period 13	0.100	0.488	0.102	0.482
Period 14	0.100	0.486	0.102	0.494
Period 15	0.100	0.484	0.101	0.491
Period 16	0.100	0.482	0.100	0.473
Period 17	0.100	0.481	0.100	0.469
Period 18	0.100	0.479	0.102	0.517
Period 19	0.100	0.477	0.104	0.453
Period 20	-	0.476	-	0.467

We see that the policy function shows little variation across the state variables that are not the time-to-maturity. This can be seen since the violin plots are very tight around the mean at every point in time, but the variation in portfolio weights comes mainly from the value of the mean at each time point.

Finally, the graph on the lower right-hand of the figure shows the violin plots for the consumption-wealth ratio that results from the policy function of the NN methodology in blue as well as the optimal value, shown in red, for each quarter. We see that in the first 18 quarters, there is almost no difference between the policy function values and the optimal ratio. Only in the last two quarters can we see a small difference between the values. Again, the dependence on the state variables other than time-to-maturity is small, as the box plots are very tight. The variation comes almost exclusively from the time-to-maturity, which is optimal.

To give a numerical overview of the results, we present the average values of the portfolio weights and consumption levels in Table 1. The means for the portfolio weight are all less than 0.7 percentage points away from the optimal solution. More interestingly, we can see that the consumption levels are very close to the optimal consumption levels.

In summary, we see that the NN methodology performs well in this slightly more complicated setup than shown in Section 3.1.1. All differences are economically insignificant.

3.2 Predictable Returns

For this exercise, we follow the setup in Section 3.2 of Brandt et al. (2005) and assume the following dynamics for quarterly stock returns

$$r^e_{t+1} = 0.227 + 0.06dp_t + \varepsilon^r_{t+1} \tag{17}$$

$$dp_{t+1} = -0.155 + 0.958dp_t + \varepsilon^d_{t+1} \tag{18}$$

with r^e_t denoting the excess log stock return, dp_t denoting the log dividend yield, and ε^r_t and ε^d_t denoting residuals from a bivariate normal distribution with variances 0.006 and 0.0049, respectively, and a covariance of -0.0051. In addition, we assume that the maturity is 5 years, i.e., we have a 20-period problem with each period comprising one quarter.

3.3 Maximizing Utility of Terminal Wealth

For this setup Jurek and Viceira (2011b) developed an approximate solution. The policy function is given

$$x^{(t)}_{T-t} = A^{(t)}_0 + A^{(t)}_1 z_{T-t} \tag{19}$$

with z_{T-t} denoting the vector containing the state variables, in our case the excess return and the log dividend-price ratio. The parameters $A^{(t)}_0$ and $A^{(t)}_1$ are functions of the model parameters, as shown in the appendix to the paper by Jurek and Viceira (2011b). Note that the parameters are time-dependent and are found recursively by using backward induction. Since the portfolio policy does not restrict the weight on the risky asset to be between zero and one, we impose this restriction in our simulation exercise by setting the output to one (zero) in case the resulting weight using equation (19) is above one (below zero).

We use 10 nodes for the hidden layer together with a "relu" activation function. The decision layer for the portfolio policy uses a "softmax" layer. Again, we simulate 100,000 sequences of 20 quarters each to test the policy function fitted by the neural network, i.e., the network has no prior exposure to this data.

The main result can be found in Figure 5, which shows the distribution of the terminal wealth across the testing sequences on the left-hand side of the upper row. We see that the results are very similar. In terms of numerical outcomes, we can state that the difference in certainty equivalent returns between the two portfolio policies is 2.1 basis points per year.

To get more insight into the performance of the NN methodology, we show the impact of the time-to-maturity on the portfolio weight depicted on the right-hand side of the upper row of Figure 5. For this graph, we set the value of the risk-free rate to 1.5%, the risky asset return to 2.5%, and the value of wealth to the starting wealth of 1.0. We use three values for the log dividend price ratios, namely -3.6, -3.6, and -3.7. Finally, we vary the value for the period under consideration from 0 to 20. As expected, the optimal exposure to the risky asset is downward sloping across time, i.e., it falls if the time-to-maturity gets shorter. It is interesting to see that the policy function is also monotonically downward sloping, as the optimal solution. However, the slope is steeper than optimal. The lower row in Figure 5 shows the violin plots for the portfolio weight for the policy function method superimposed on the approximately optimal solutions on the left-hand side. The optimal solutions are in red, the policy function outcomes in blue, and the overlapping part in purple. We can see that the variation in optimal portfolio policies is large since it depends on the time to maturity as well as on the value of the dividend-price ratio. We see again, that the optimal policy is downward sloping with time passing and that the slope for the average decision is slightly steeper for the policy function, depicted in blue. The right-hand side of the second row in Figure 5 shows the optimal portfolio decision across the states of the world after 10 periods, i.e., halfway through the 5 years. We plot the histogram of the log dividend-price ratio using 100 bins in grey and compute the average portfolio decision for each bin in blue, for the optimal decision, and in red, for the outcome of the NN methodology. The portfolio decision should be constant for each bin since the log dividend-price ratio is the only state variable in our economy. However, note that we also pass redundant state variables to the neural network, whose exposures are not estimated to be exactly zero, which leads to small variations in the weights for the NN methodology. We see that the policy function results closely follows the optimal solution across a large part of the distribution. Only for the right tail do we see small differences, as the policy function is slower in approaching the upper limit of 1.

Finally, we show the average portfolio policies across periods for the optimal values and the policy function values in Table 2. We see that the averages differ by less than 3 percentage points and that the slope of the policy function is steeper, it starts at 39.2% at Period 0 and ends at 15.5% at Period 20, whereas the optimal method goes from 36.9% to 18.6%.

In summary, we note that the NN methodology approximates the solution using backward induction well. The differences in certainty equivalent returns are economically insignificant. We also see that basic properties like a downward-sloping portfolio weight over time are also present when estimating the policy function using a neural network. We want to note again that we have not used any prior knowledge when setting up the structure of the network. Like in the i.i.d. case, there are redundant state variables that we pass to the policy function. These are the asset returns and

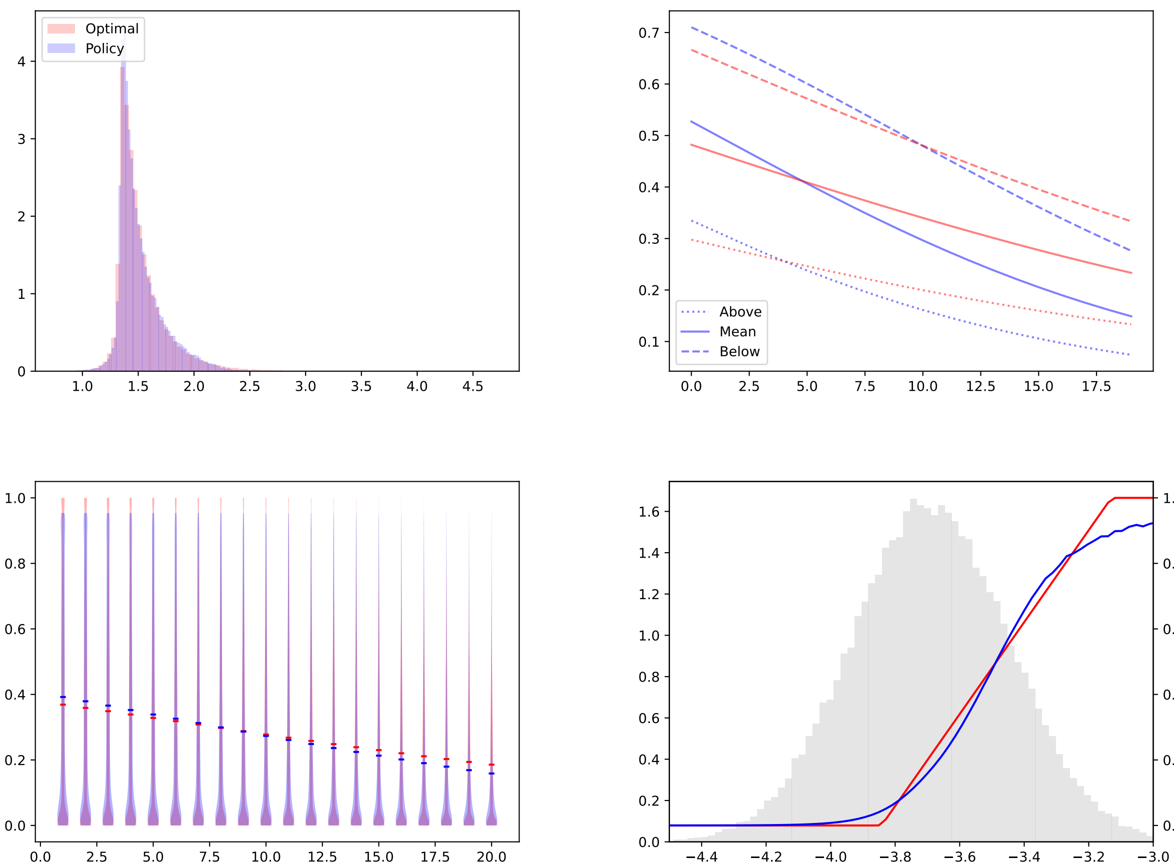


Figure 5: This figure shows the performance of the NN methodology in the case the investor maximizes the utility of terminal wealth and the economy follows an i.i.d. process. The upper row shows the distribution of the terminal wealth for each sequence of the test data when using the optimal solution in light red and when using the policy function in light blue. For the case in which both distributions overlap, the color turns light purple. The right side of the upper row shows the distribution of the weight on the risky asset for the policy function across all time points and sequences of the test data in light blue, the optimal solution is plotted as a red line. The lower row of the figure shows a violin plot for the portfolio weights at each point in time as well as the optimal solution as a red line.

Table 2: This table shows the average value of weight on the stock for the backward induction method and the policy function method.

Period	Back. Ind.	Policy
Period 0	0.369	0.392
Period 1	0.359	0.379
Period 2	0.349	0.366
Period 3	0.339	0.352
Period 4	0.328	0.339
Period 5	0.318	0.326
Period 6	0.308	0.313
Period 7	0.298	0.299
Period 8	0.288	0.287
Period 9	0.278	0.273
Period 10	0.268	0.261
Period 11	0.258	0.249
Period 12	0.248	0.236
Period 13	0.239	0.224
Period 14	0.230	0.213
Period 15	0.220	0.202
Period 16	0.211	0.190
Period 17	0.203	0.180
Period 18	0.194	0.169
Period 19	0.186	0.159

the value of the wealth for the model under consideration in this section.

3.4 Solving the Consumption and Portfolio Problem

For this case, we have no closed-form solution benchmark. The closest setup to ours can be found in Campbell and Viceira (1999). In that paper, the investor is infinitely lived and has recursive utility. By setting the intertemporal elasticity of substitution (IES) equal to the coefficient of risk aversion the solution applies to a CRRA investor. We will not directly compare the outcomes of the NN methodology to the approximate optimal strategy developed in Campbell and Viceira (1999) since we assume a finite-lived investor. Therefore, the outcomes using the NN methodology will be superior to the strategy that assumes an infinite-lived agent when the subjective discount factor is 0.96. Instead of re-calibrating the subjective discount factor, we will compare our results to the paper by Campbell and Viceira (1999) qualitatively.

The outcomes using the NN methodology can be found in Figure 6. The graphs on the first row of the figure show the distribution of portfolio weights for each period on the left-hand side and the distribution of consumption levels on the right. We see that the average portfolio weight is downward sloping, whereas the consumption level has no clear trend.

More of interest is the analysis in the following rows of the figure. The second row shows how the weight of the stock in the portfolio. We show the decisions across all test sequences after 10 periods, i.e., halfway through the 5 years. We compute the cumulative stock return for each sequence and create 100 bins, which form the basis for the histogram shown in grey on the left-hand side of the second row and third row. For each of the bins, we compute the average portfolio weight that the investor chooses (second row) or the consumption decisions (third row) and plot these decisions as the black line in the graphs. The right-hand side of the second row and the last row show the portfolio and consumption decisions as a function of the dividend-price ratio.

Concerning the portfolio decision in the second row, we see that the investor reacts to changing investment opportunities, as the portfolio decisions are no longer constant across states. As expected, we see that the weight of the stock goes up in case expected returns increase. Since the coefficient on the log dividend-price ratio in Equation (17) is positive the expected return increases with the value of this state variable. This explains the functional form of the line in the graph on the right side of row two in Figure 6. The portfolio weight is close to zero for very small values of the log dividend-price ratio and increases quickly when this variable increases beyond -3.8, approaching one at the right end of the graph. This is qualitatively what Campbell and Viceira (1999) show in their Figure I on p. 457. The main difference to our result is that the portfolio weights are restricted to be between zero and one, which is not the case in the analysis of Campbell and Viceira (1999), leading to a linear relation between the state variable and the portfolio weight. The graph on the left-hand side of row two shows the portfolio decisions as a function of the past cumulative returns

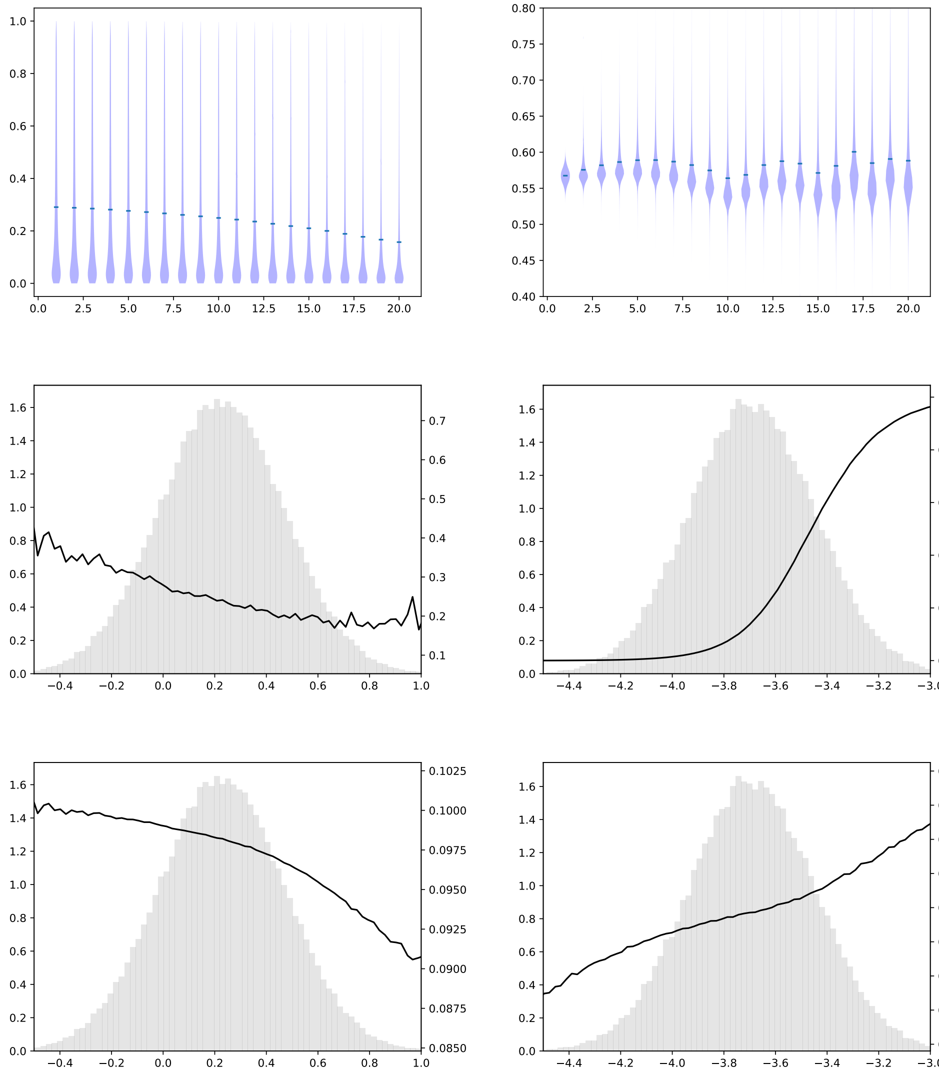


Figure 6: This figure shows the performance of the NN methodology in the case the investor maximizes the utility of terminal wealth and the economy follows an i.i.d. process. The upper row shows the distribution of the realized utility results for each sequence of the test data when using the optimal solution in light red and when using the policy function in light blue. For the case in which both distributions overlap, the color turns light purple. The right side of the upper row shows the distribution of the weight on the risky asset for the policy function across all time points and sequences of the test data in light blue, the optimal solution is plotted as a red line. The left-hand side of the lower row of the figure shows the violin plots for the portfolio weights at each point in time as well as the optimal solution as a red line. On the right-hand side of the lower row, we see the violin plots of the consumption-wealth ratio for each period in blue together with the optimal value depicted in red.

on the stock. Since the return process is mean reverting, we see that the average portfolio weight tends to decrease with increasing prior cumulative returns.

Turning to the third row of Figure 6, we see the variation of the consumption-wealth ratio across states. The investor increases the consumption-wealth ratio in case expected returns go up, which can be seen in the graph on the left-hand side of the third row of the figure. The consumption-wealth ratio is increasing with the value of the log dividend-price ratio, where the latter implies a higher expected return going forward. A similar relation can be found in Figure II.b on p. 463 of Campbell and Viceira (1999). In their paper, the relation between the expected log return and the consumption-wealth ratio increases if the IES parameter is below one. This is the case for our analysis, since we have $\gamma = 10$ and $IES = 1/\gamma$. The left-hand side of the third row shows the consumption-wealth ratio as a function of past cumulative returns. As mentioned above, the mean reversion property of the return process implies that expected returns going forward are lower if past returns are large. This results in a downward-sloping consumption-wealth ratio.

In summary, we note that, although we are not able to compare the direct numerical results of the NN methodology to an approximate closed-form solution, we see that the outcomes of the method can replicate portfolio and consumption decisions qualitatively that have been found in the literature. Taking all results for the CRRA investor into account, we are confident that the NN methodology is an efficient algorithm to determine policy functions for the optimization of life-cycle models.

4. LOSS-AVERSION UTILITY FUNCTION

We now move on from the benchmark specification of a CRRA investor and assume the utility of the investor follows the two-part power utility function developed by Tversky and Kahneman (1992). In particular, we assume that

$$U(C_t) = \begin{cases} (C_t - \theta)^{\gamma_G} & C_t \geq \theta \\ -\kappa\,(\theta - C_t)^{\gamma_L} & C_t < \theta \end{cases} \tag{20}$$

with $\gamma_G \in (0, 1)$ and $\gamma_L > 0$ denoting the curvature parameters for the gain and the loss domain, respectively. The parameter κ is the loss-aversion parameter and θ denotes the reference level of consumption, or terminal wealth, for the case the investor aims to maximize the utility of terminal wealth.

The solution to the dynamic optimization problem increases in complexity since the decisions are no longer wealth-independent. This can be seen by the fact that relative risk-aversion is no longer constant, but depends on how far the current consumption level is from the reference level, i.e.,

$$\frac{-C_t U_{CC}(C_t)}{U_C(C_t)} = \begin{cases} (1-\gamma_G)\,\frac{C_t}{C_t-\theta}, & \text{if } C_t > \theta \\ +\infty, & \text{if } C_t = \theta \\ -\kappa\,(1-\gamma_L)\,\frac{C_t}{\theta-C_t}, & \text{if } C_t < \theta \end{cases} \tag{21}$$

The main impact of solving the problem using backward induction methods is that the decision at each time t depends on the current wealth level, which is a function of past decisions that are not known when using backward induction. The usual way to deal with this problem is to solve the problem on a grid, which increases the computational burden considerably. This is one of the main advantages of the NN methodology since we use an algorithm that goes forward in time.

Although there is no approximate closed-form solution for discrete-time models, Berkelaar et al. (2004) and Van Bilsen et al. (2020) provide solutions for continuous-time models. The paper by Berkelaar et al. (2004) solves the problem of maximizing the utility of terminal wealth, whereas Van Bilsen et al. (2020) solves the consumption and portfolio problem. Both papers provide closed-form solutions for i.i.d. economies and unrestricted portfolio weights. Our approach differs in terms of the economic model. We assume a discrete-time model and it is straightforward to include non-constant investment opportunities into the model. Second, we restrict the portfolio weights to avoid values for the portfolio weights that could not be implemented in practice.

4.1 Identically and Independently Distributed Returns

4.1.1 Maximizing Utility of Terminal Wealth

We use the same data-generating process as described in Section 3.1. As expected, the portfolio strategy of the loss-averse investor differs considerably from the CRRA results. We summarize the results in Figure 7, where we use the results of applying the NN methodology on the 100,000 test sequences. The graph in the first row on the left-hand side of Figure 7 shows the distribution of wealth at maturity. We see that the outcomes are concentrated in a tight interval above the reference level. To understand this outcome, we plot the portfolio decisions and values of wealth as a function of realized cumulative returns in the lower row of Figure 7. We show the decisions across all test sequences after 10 periods, i.e., halfway through the 5 years. We compute the cumulative stock return for each sequence and create 100 bins, which form the basis for the histogram shown in grey in both graphs of the lower row. For each of the bins, we compute the average portfolio weight that the investor chooses and plot these decisions as the black line in the left-hand graph of the second row in Figure 7.

The first striking result is that the portfolio decision is not constant, even in the case of an i.i.d. economy. In case of low past cumulative returns, the investor increases the portfolio weight on the

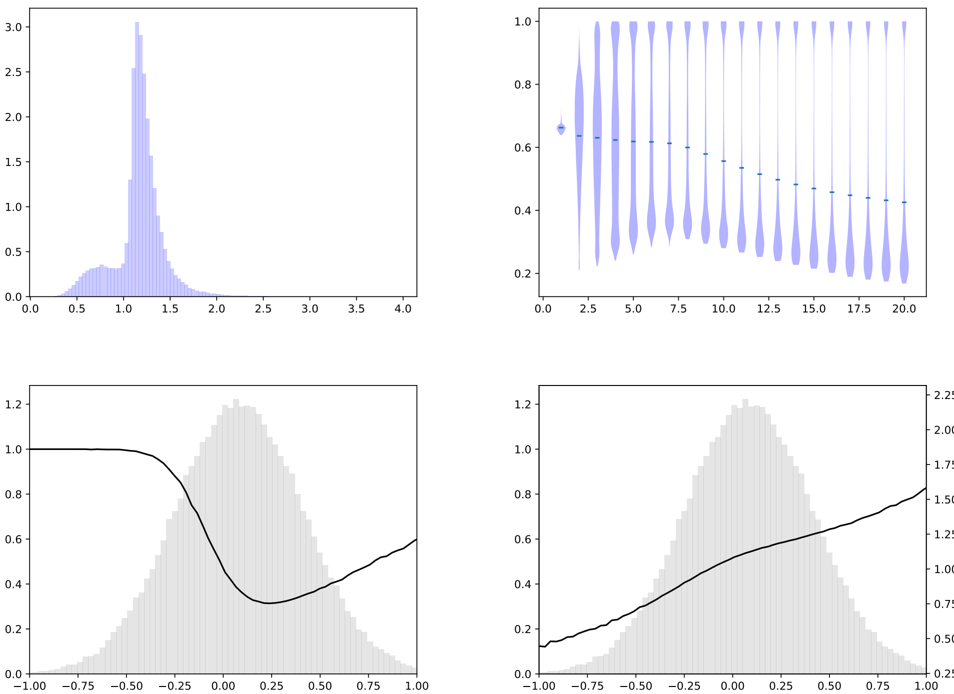


Figure 7: This figure shows the performance of the NN methodology in the case the loss-averse investor maximizes the utility of terminal wealth and the economy follows an i.i.d. process. The upper row shows the distribution of the terminal wealth for each sequence of the test data when using the policy function in light blue. The right side of the upper row shows a violin plot for the portfolio weights at each point in time. In the second row, we plot the portfolio decisions and values of wealth as a function of realized cumulative returns. We show the decisions across all test sequences after 10 periods, i.e., halfway through the 5 years. We compute the cumulative stock return for each sequence and create 100 bins, which form the basis for the histogram shown in grey in both graphs of the lower row. For each of the bins, we compute the average portfolio weight that the investor chooses and plot these decisions as the black line.

stock up until the limit of 100% for large negative returns. The reason is, that the initial wealth in these cases is below the threshold level θ, as can be seen on the right-hand graph of the same line in the figure. For these states of the world, the investor will increase the portfolio allocation to the stock, to push the level of wealth up towards the value of θ. In states of the world where the cumulative stock return is in the middle of the distribution, i.e., where the value of wealth is close to the reference level, the investor scales down the portfolio weight of the stock, which results in a large concentration of the distribution of terminal wealth around the threshold. If returns become large the portfolio allocation to the stock increases again. This portfolio strategy results in a level of wealth that does not increase linearly with the realized stock return, as can be seen in the right-hand side graph. Finally, we see a maturity effect on the portfolio allocation in the right-hand side of the graph in the first row of Figure 7. This graph plots the violin plots for the portfolio decisions at each point in time. We see that the average portfolio allocation to the stock decreases with time.

4.1.2 Solving the Consumption and Portfolio Problem

The results of this exercise are collected in Figure 8. As before, we plot the outcomes after using the policy function of the NN method on the 100,000 sequences of the test sample. The first row of the figure shows the box plots for the portfolio allocation to the stock on the left-hand side and the consumption decision on the right-hand side. We see on the left side, that the average weight on the stock is increasing with time. Equally striking is the picture of the consumption levels on the right-hand side. We see that the average consumption levels are held almost constant with tight bands around the reference level for the first 15 periods. Only then does the average decrease and the variance around the mean increase, which results due to the subjective discount factor.

The second and third rows of Figure 7 show the decisions across states instead of across time. Moving to the second row, we see the decisions on the portfolio weight after 10 periods across the distribution of the prior cumulative stock returns. The computation of these is identical to the description in Section 4.1.1. The portfolio weight is again increasing and equal to one if the cumulative return of the stock is negative, as in the last section. However, the portfolio weight does not increase if the stock return has been high in the periods before. The result of this investment strategy on the wealth of the investor can be seen on the right-hand side of the second row. We see that wealth increases with past stock returns, however, the slope decreases since the portfolio weight is reduced in case past stock returns are high. Finally, we see how a loss-averse investor adjusts the consumption to prior stock market outcomes. In case stock market returns are low, the wealth-consumption ratio, shown on the left-hand graph of the figure, adjusts upward. The slope of the curve decreases in absolute value across the return distribution, with the main effect being found around the center. The impact on consumption levels can be seen in the graph on the right-hand side of the third row of Figure 8. We see that consumption levels are constant at the center of the distribution, showing that the investor tries to maintain a consumption level at the

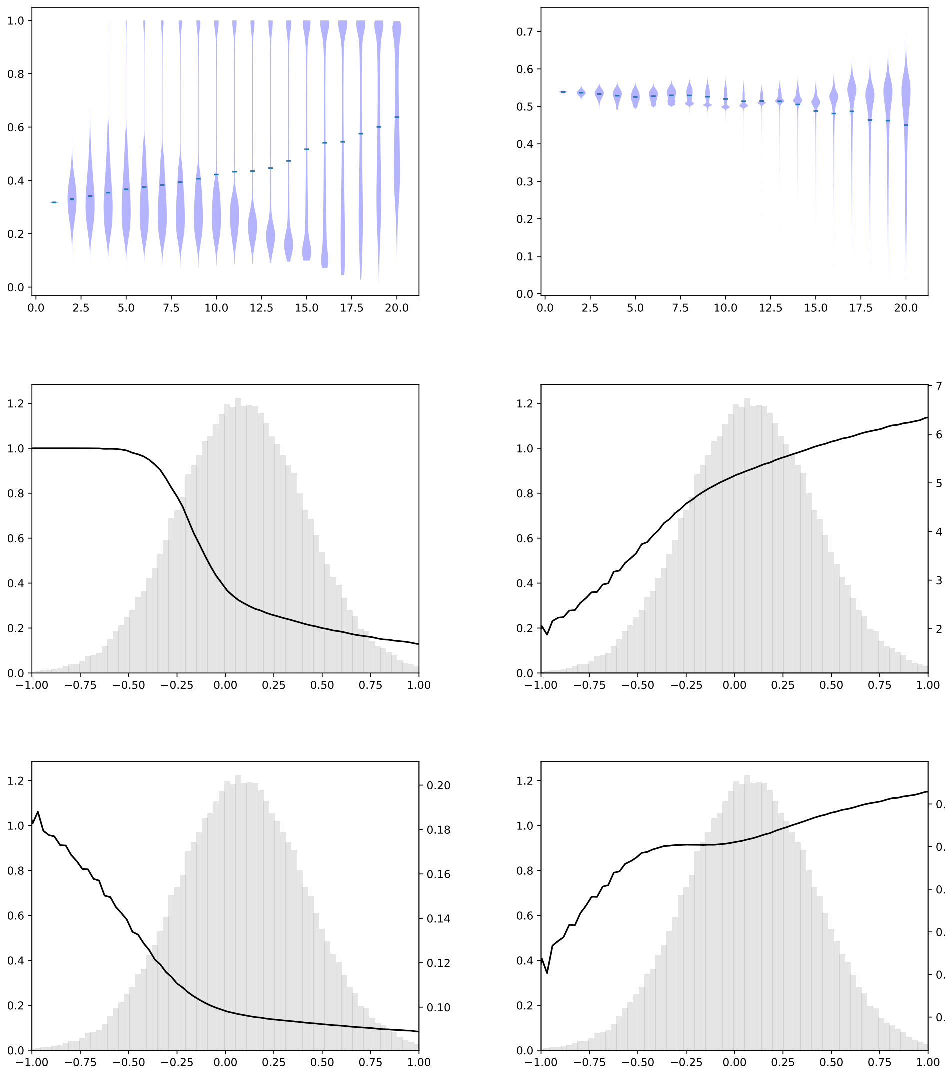


Figure 8: This figure shows the performance of the NN methodology in the case the loss-averse investor solves the consumption and portfolio problem and the economy follows an i.i.d. process. The first row shows the violin plots for the optimal portfolio weights across time on the left-hand side and the optimal consumption levels across time on the right-hand side. In the second row, we plot the portfolio decisions and values of wealth as a function of realized cumulative returns. We show the decisions across all test sequences after 10 periods, i.e., halfway through the 5 years. We compute the cumulative stock return for each sequence and create 100 bins, which form the basis for the histogram shown in grey in both graphs of the lower row. For each of the bins, we compute the average portfolio weight that the investor chooses and plot these decisions as the black line. The third row shows the consumption-wealth ratios across states on the left-hand side and the consumption levels on the right.

reference level even in the case of stock market losses. The result of this approach explains the pattern observed in the graph on the first row, namely that the variation is large for later periods, where the decrease in consumption levels in case of bad stock returns cannot be avoided anymore.

4.2 Predictable Returns

4.2.1 Maximizing Utility of Terminal Wealth

We use the same data-generating process as described in Section 3.2. The results for this exercise can be found in Figure 9. The first row of the figure shows the distribution of terminal wealth across the 100,000 sequences of the test dataset on the left-hand side and the distribution of portfolio weights on the risky asset across time on the right-hand side. We see a spike in the distribution of terminal wealth at the value of about 1.5, which is above the reference level. We can also see that the probability of a realization below the reference level is small since the overwhelming part of the distribution lies to the right of the starting wealth of one. On the right-hand side of row one, we see that the portfolio weights are concentrated mostly in the tails of the distribution, i.e., are either close to zero or close to one.

This brings us to the second row of Figure 9 where we can see the portfolio decisions after 10 periods across the states of the economy represented by different values of the log dividend-price ratio. We compute the average portfolio weight for each of the 100 bins that form the histogram of the value of the log dividend-price ratio at the beginning of period 11. The graph shows that as in Section 3.3, the investor increases the portfolio weight when future expected returns go up. However, the adjustment is much more pronounced than in the CRRA case, i.e., the loss-averse investor does so more aggressively. This explains the concentration of portfolio weights in the tails that we observe in the graph on row one of the figure. Finally, we see the average value of starting wealth across the different states of the economy after 10 periods on the right-hand side of the second row in Figure 9. There is almost no impact as we see a only weakly decreasing function.

4.2.2 Solving the Consumption and Portfolio Problem

The results of this analysis can be found in Figure 10. The first row of the figure shows the distribution of the portfolio weights across time on the left-hand side as well as the distribution of the consumption level on the right-hand side. The portfolio weights exhibit the same behavior as in Section 4.2.1, namely that the weights are concentrated at the lower or upper bound of the distribution. On the right side, we see that the average levels of consumption are flat across time, with the distributions skewing to the top and exhibiting a low number of realizations that are below the reference level.

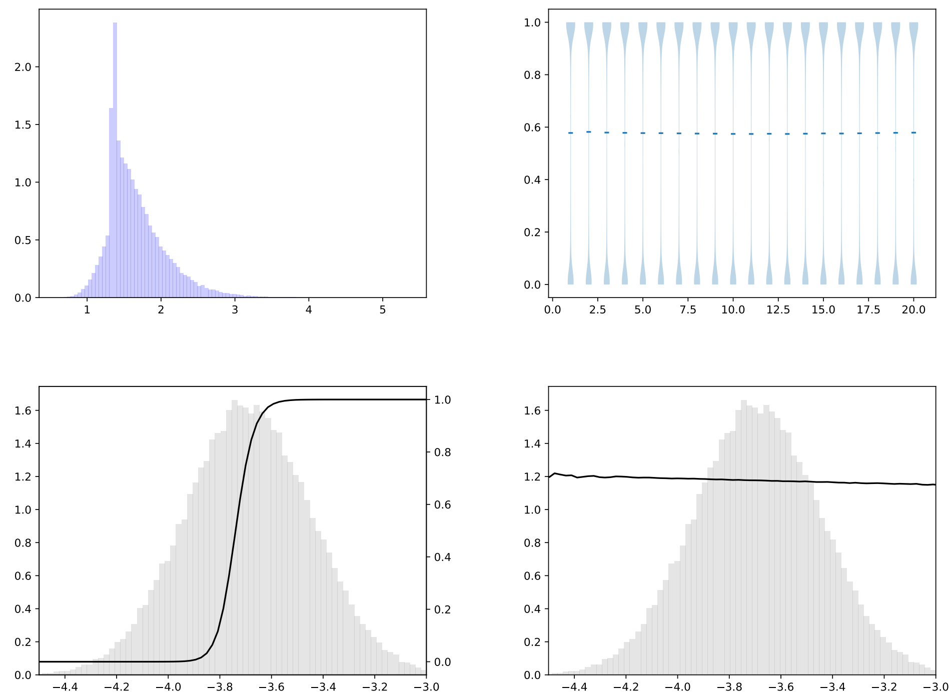


Figure 9: This figure shows the performance of the NN methodology in the case the loss-averse investor maximizes the utility of terminal wealth and the economy follows the VAR process of Equations (17) and (18). The upper row shows the distribution of the terminal wealth for each sequence of the test data when using the policy function in light blue. The right side of the upper row shows a violin plot for the portfolio weights at each point in time. In the second row, we plot the portfolio decisions and values of wealth as a function of the log dividend-price ratio. We show the decisions across all test sequences after 10 periods, i.e., halfway through the 5 years. We separate the value for the log dividend-price ratio at the beginning of the next period for each sequence and create 100 bins, which form the basis for the histogram shown in grey in both graphs of the lower row. For each of the bins, we compute the average portfolio weight that the investor chooses and plot these decisions as the black line.

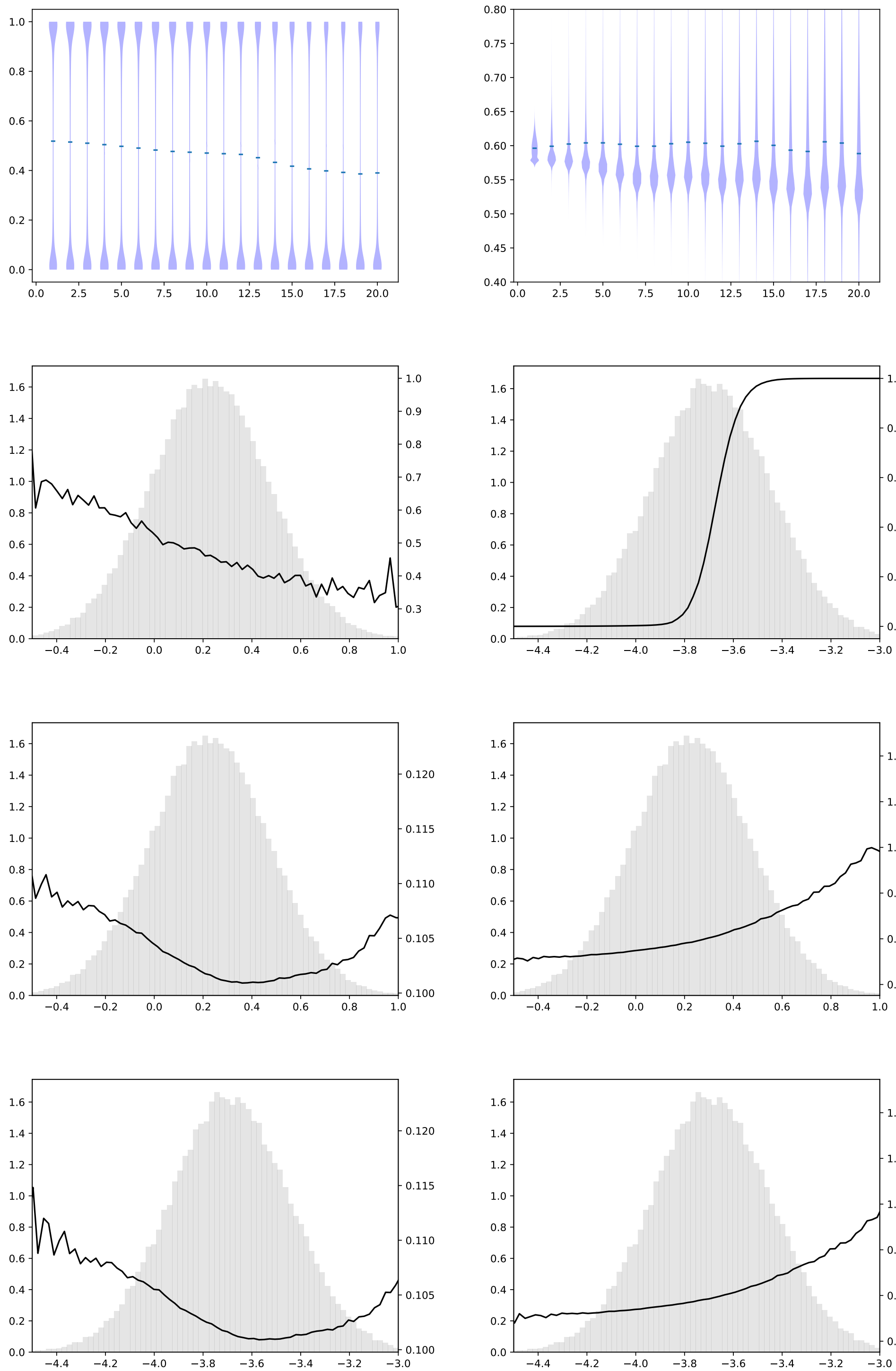


Figure 10: This figure shows the performance of the NN methodology in the case the loss-averse investor solves the consumption and portfolio problem. The first row shows the violin plots for the optimal portfolio weights across time on the left-hand side and the consumption level across time on the right. The second row shows the portfolio decisions across states of the world. The left side determines the states of the world using prior realized cumulative stock returns, the states are measured using the log price-dividend ratio on the right. The second row shows the consumption-wealth ratio on the left side and the consumption level across the distribution of prior cumulative returns and the last row shows the same consumption decisions across values of the log dividend-price ratio.

In row two of Figure 10 we see the portfolio decisions across the states for the 100,000 sequences in the test dataset. We have the average portfolio weight on the stock in each bin of the histogram for the prior realized cumulative stock market return after 10 periods. We see that the weight is going down as a function of this cumulative return. This can be explained by the mean-reverting property of the return dynamics of the model in Equation 17. The same explanation can be used to interpret the observations on the right-hand side of the same row in the figure. We see that the portfolio weight increases in case of an increase in the log dividend-price ratio. As in the results discussed in Section 4.2.1 we see a quick adjustment from close to zero to close to one in case the log dividend-price ratio increases above about -3.8.

The third row of Figure 10 shows the results for the consumption decisions across the states of the world which are measured as prior cumulative stock returns. We see the results for the consumption-wealth ratio on the left-hand side and the corresponding consumption levels on the right-hand side. We observe a U-shaped relation between the prior returns and the consumption-wealth ratio. This can be explained by the fact that the investor targets a reference level of wealth. As shown on the right-hand side, the consumption level is almost flat in the left part of the distribution and only increases after the prior cumulative returns become positive. In the case of large positive prior returns, we see that the consumption-wealth ratio as well as the consumption level increases considerably.

The last row of Figure 10 shows the consumption decisions as a function of the log dividend-price level. The relationship is very similar to the decisions across prior cumulative returns. We do not observe the striking difference as in the case when deciding on the portfolio weight.

5. CONCLUSION

We propose a new forward-solving algorithm to solve life-cycle models. Our method uses neural networks to train the parameters of a policy function that returns the optimal decisions as a function of state variables. The main advantages are that we do not need to specify a functional form for these policy functions as in, e.g., Chacko et al. (2005) since the flexibility in the activation function allows the algorithm to adapt the form of the policy function to the problem at hand. In addition, the setup allows the solution to go forward in time, which makes the inclusion of endogenous state variables, like in the case of a loss-averse investor, easy to handle. Finally, our method only relies on the ability to simulate the economic model as well as computation of the criterion function and is therefore straightforward to apply to more complicated setups as discussed in this paper.

We show that the NN methodology returns solutions that are very close to the optimal solutions in cases where we can compute closed-form approximations of these optimal solutions. We also show that the algorithm can perform this well even if we do not use prior knowledge to set up the policy function. All the results we show in Section 3 pass redundant state variables to the policy function as, e.g., the wealth level, which is not important in the CRRA case. This shows that the neural network can learn to distinguish between important and redundant states.

Finally, we apply the NN methodology to cases where there are no known closed-form solutions. In particular, we analyze the decisions of a loss-averse investor who is short-sale and credit-constrained, finitely lived, and can only make decisions in discrete intervals.

A. IMPLEMENTATION OF THE NETWORK

We use the functional API in Keras, which is implemented within the Tensorflow platform in Python. The first step is to simulate a large time series of the economic model, i.e., we construct a $\Theta \times S$ matrix containing asset returns, state variables, and information on the respective time to maturity. We set $\Theta = (T+1) \cdot N_{sim} \cdot N_{batch}$ with N_{sim} denoting the number of sequences included within each batch and N_{batch} denotes the sum of the number of batches used for training and validation.

The second step is to prepare the simulation data in a way that can be used by the Keras API. We use the `keras.utils.timeseries_dataset_from_array` method for this. We pass the simulated time series to the method and it partitions the data into batches of size $N_{sim} \times T + 1 \times S$. The training data has N_{train} number of batches and the validation data has N_{val} number of batches. In addition to the simulated data, we pass a starting value for the wealth at time $t = 0$, so that the output is conditional on this starting value.

Setting up the network proceeds as follows. Keras sets up a symbolic version of the network that allows for a flexible number of sequences within each batch. This results in the construction of tensors of size $None \times T + 1 \times S$ for the input data, with $None$ denoting the flexible number of sequences used. Since we need to pass the starting wealth in a conformable tensor, the starting wealth will be included in a $None \times 1$ tensor.

The next step consists of setting up the layers containing trainable parameters. Since the values of the parameters do not vary across time points, we set up the layers outside the for loop that iterates over time. Each layer is set up using the method `keras.layers.Dense`, where we pass the number of nodes used as well as the information on the activation function. These layers are then used within a for loop that collects and uses the information over the time periods. At each point in time, we pass the respective state variable values concatenated with the current value for the wealth, resulting in a $None \times S + 1$ tensor, to the hidden layers. The outputs of the hidden layers, which are $None \times N_{nodes}$ are then passed to the respective decision layers, which return a $None \times K$ and a $None \times 1$ tensor, respectively. The final step in the for loop is to update the wealth value, which is done using a custom layer, using the `keras.layers.Lambda` method. This custom layer uses the concatenated information on the value of the wealth at the beginning of the period, the output of the consumption and portfolio decision layer, and the value for the next period asset returns (and possibly income), as input and returns a $None \times 1$ tensor containing the wealth at the end of the period.

The final step is the construction of the loss function that is used to train the model to maximize the expected utility of the investor. This loss function accepts a tensor of dimension $None \times T + 1$ containing the information on the consumption levels at each point in time as well as the value of terminal wealth. The loss function computes the utility values for each of these values, computes the mean over the dimension that contains the sequences within the batch, i.e., the $None$ dimension, and finally computes the sum over the time dimension, resulting in the expected utility given the consumption and portfolio decisions. The total loss at the end of each epoch is computed as the average loss across the batches, with the optimization method computing the gradients for each value within the batch.

After compilation of the network, using the method `compile`, the method `fit` optimized the network parameters and fills the symbolic representation of the tensors with values, i.e., changes the $None$ to the number of sequences within each batch. To fit the network, we also use the callback option and select the parameters that result in the best validation value.

References

Berkelaar, A., Kouwenberg, R., and Post, T. (2004), "Optimal Portfolio Choice under Loss Aversion," *Review of Economics and Statistics*, 86, 973–987.

Brandt, M., Goyal, A., Santa-Clara, P., and Stroud, J. (2005), "A Simulation Approach to Dynamic Portfolio Choice with an Application to Learning About Return Predictability," *Review of Financial Studies*, 18, 831–873.

Campbell, J. (2018), *Financial Decisions and Markets*, Princeton University Press.

Campbell, J. and Viceira, L. (2002), *Strategic Asset Allocation: Portfolio Choice for Long-Term Investors*, Oxford University Press.

Campbell, J. Y. and Viceira, L. M. (1999), "Consumption and portfolio decisions when expected returns are time varying," *The Quarterly Journal of Economics*, 114, 433–495.

Chacko, G., Desai, M., Golts, M., and Novakovsky, V. (2005), "A Forward-Solving Numerical Technique for Dynamic Consumption and Portfolio Allocation Problems," SSRN Working Paper 687622.

Chen, A., Pelsser, A., and Vellekoop, M. (2011), "Modeling Non-Monotone Risk Aversion using SAHARA Utility Functions," *Journal of Economic Theory*, 146, 2075–2092.

Chen, Y., Cosimano, T. F., and Himonas, A. A. (2014), "On Formulating and Solving Portfolio Decision and Asset Pricing Problems," in *Handbook of Computational Economics*, Elsevier, vol. 3, pp. 161–223.

Judd, K. (1998), *Numerical Methods in Economics*, MIT Press.

Jurek, J. and Viceira, L. (2011a), "Optimal Value and Growth Tilts in Long-Horizon Portfolios," *Review of Finance*, 15, 29–74.

Jurek, J. W. and Viceira, L. M. (2011b), "Optimal value and growth tilts in long-horizon portfolios," *Review of Finance*, 15, 29–74.

Pennacchi, G. (2008), *Theory of Asset Pricing*, New York: Pearson Addison Wesley, 1st ed.

Tversky, A. and Kahneman, D. (1992), "Advances in Prospect Theory: Cumulative Representation of Uncertainty," *Journal of Risk and Uncertainty*, 5, 297–323.

Van Bilsen, S., Laeven, R., and Nijman, T. (2020), "Consumption and Portfolio Choice under Loss Aversion and Endogenous Updating of the Reference Level," *Management Science*, 66, 3799–4358.