Modern Wage Dynamics ODD

J M Applegate
version 1.0.0
Summer School 2022: "Mathematics of Complex Social Systems"
June 9-17, 2022
Zuse Institute, Berlin

0.1 Model Purpose

The Modern Wage Dynamics Model is a generative model of coupled economic production and allocation systems. Each simulation describes a series of interactions between a single aggregate firm and a set of households through both labour and goods markets. The firm produces a representative consumption good using labour provided by the households, who in turn purchase these goods as desired using wages earned, thus the coupling. The model employs a variant of efficiency wage theory where worker effort is a function of the wage they receive, and production is based on effective effort rather than worker hours. The households have independent and dynamic effort-wage response functions. The firm has incomplete information with regards to the aggregate households' effort response function and demand, and attempts to learn these relationships over time.

Each model iteration the firm decides wage, price and labour hours requested. Given price and wage, households decide both effort and hours worked based on their effort response functions and a utility function for leisure and consumption. A labour market construct chooses the minimum of hours required and aggregate hours supplied, and aggregates the effort provided. The firm then uses these inputs to produce goods. Given the hours actually worked, the households decide actual consumption and a market chooses the minimum of goods supplied and aggregate demand. The firm uses information gained through observing market transactions about effort and consumption demand to refine their conceptions of the population's effort-wage response and demand.

The purpose of this model is to explore the general behaviour of an economy with coupled production and allocation systems, as well as to explore the effects of various policies on wage and production, such as minimum wage, tax credits, unemployment benefits, and universal income.

0.2 Model Overview

The model consists of a single aggregate firm representing all economic production and multiple households. The firm produces a single aggregate good, called *sugar*. The households provide wage labour to the firm and consume sugar. The model begins with the firm choosing the wage, ω, price of sugar, p, and hours requested, H_D. Given ω, each household choses a level of effort, e_i, and the hours provided, H_i, according to preferences for leisure and consumption. A labour market determines H_M as the lesser of H_D and H_S. Given H_M and the various e_i levels, the firm supplies a

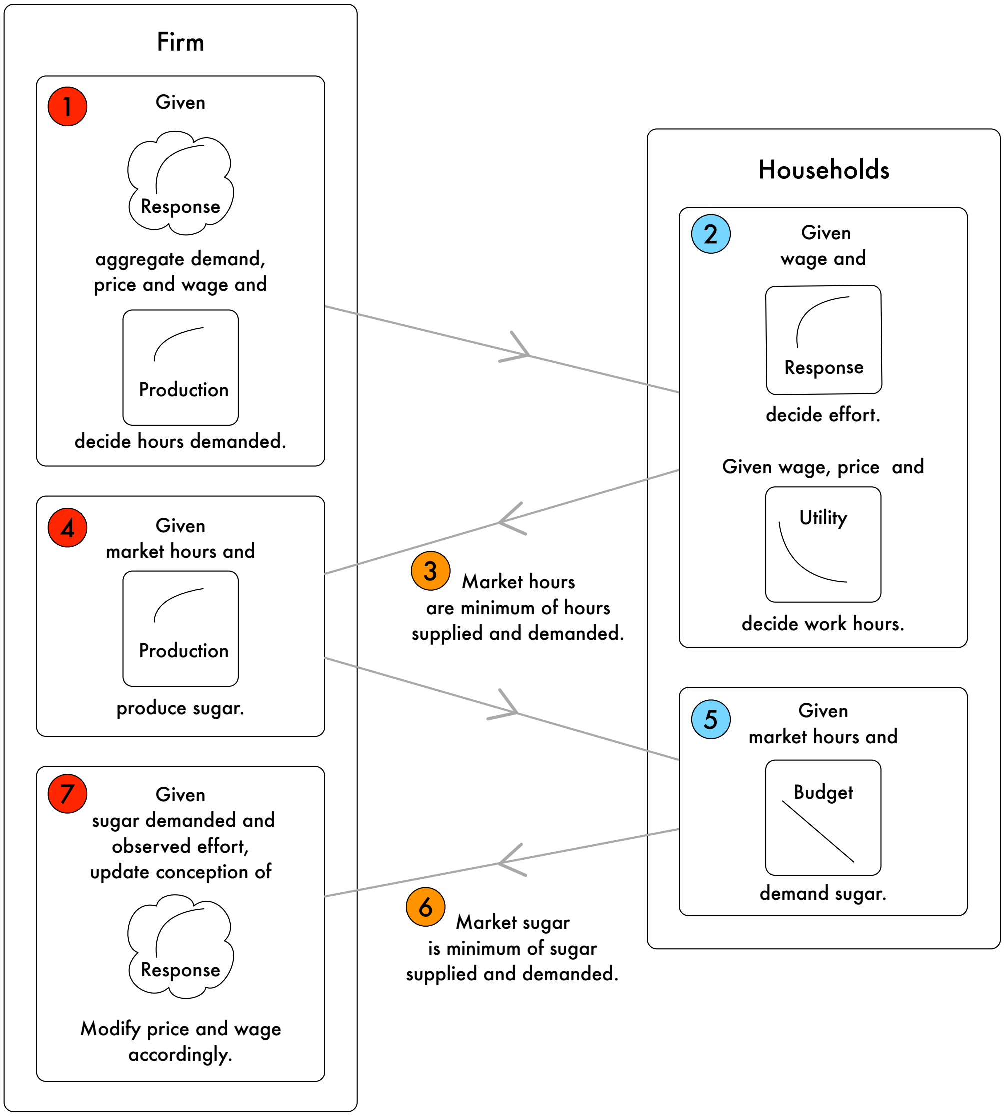

Figure 1: Modern Wage Dynamics Model Flow

quantity of sugar to the households, S_S, based on its production function and effective effort. Each household then decides how much sugar it desires according to its budget, S_i. The market then aggregates the household demand and determines the quantity of sugar sold as the lesser of S_S and S_D. Finally, firm updates its expectations of effective effort and demand, and makes modifications to price and wage as appropriate. Figure 1 shows a schematic overview of the model process.

0.2.1 The Firm's Production Function and Hours Decision

The firm starts with an initial price and wage value, as well as an expectation of sugar demand from the current model state. Firm production is modeled as a variant of the Solow production function,

$$Y = A(eH)^{\gamma} \tag{1}$$

where A combines the total factor productivity coefficient as well as the capital contribution. Thus, Y is solely a function of labour, or more specifically in our case, of total effective effort, $N = eH$, the proportion of labour hours dedicated to effort. The firm has an expectation of the aggregate household effort response curve, the relationship between effort and wage, modeled in our case by a neural network η such that $\hat{e} = \eta(\omega)$. The firm also has an expectation of aggregate household demand, and in its role as representing all economic production will attempt to meet that demand. Thus, the hours requested by the firm are

$$H_D = \frac{1}{\hat{e}} \left(\frac{S_D}{A} \right)^{\frac{1}{\gamma}}. \tag{2}$$

For self production, $\gamma = 1$ and $e = 1$, max hours worked at max effort. Our model assumes that when households produce for the firm rather than itself, that effort will depend on the wage. Thus the model requires gamma to be increasing returns to scale, which is suitable as the division of labour claims higher labour productivity than each household producing for itself.

0.2.2 The Households' Effort and Hours Decision

Each household is ascribed an individual effort response curve relating the wage, ω, to effort, e_i, given a particular disutility of effort, d_i. Based on this response curve, a household will maximize its utility for labour by minimizing its effort. Thus the household effort response curve is

$$e_i = 1 - \left(\frac{d_i}{\omega + d_i} \right)^{\frac{1}{2}} \tag{3}$$

and e is a value between 0 and 1 representing the percentage of a work hour that is effortful. Each household also decides the quantity of work hours it will provide to the firm at this effort. These hours are determined based on a Cobb-Douglas utility function for consumption and leisure. Since sugar is the aggregate consumption good in the model, the quantity of household consumption is not completely arbitrary as the household requires a necessary base level of consumption to survive, S_N. Therefore, each household will work a number of hours regardless of the utility from those hours, H_N.[1]

Given and maximum number of available hours per household, H_{max}, leisure can be described as $(H_{max} - H_N - H_O)$ were H_O are discretionary hours, and the household utility function is

$$U(H_O, S_O) = (H_{max} - H_N - H_O)^{\alpha} (S_O)^{\beta} \tag{4}$$

where $\alpha + \beta = 1$ and S_O represents discretionary sugar consumption. The total household budget, B, is the wage income, $\omega(H_N + H_O)$ plus any positive monetary holdings, $\min(0, m)$.[2]

[1] These hours can be described as *tribute hours*, a la Graeber, since they are not available for the free use of the worker, but owed to someone else in order to obtain material goods for survival.

[2] Given that households require a base level of consumption, S_N, households may experience negative holdings.

Tribute hours, H_N, are calculated as

$$H_N = \max\left(0,\ \frac{pS_N - m}{\omega}\right) \tag{5}$$

and we end up with two different cases, one where tribute hours are 0 and one where they are greater than 0. If $H_N > 0$ the household doesn't have enough non-wage funds to cover S_N, and anything they do have goes toward S_N and reducing H_N. Therefore discretionary consumption is based on the wage earned from discretionary hours alone. If $H_N = 0$, then the household has enough non-wage funds to cover necessary consumption, and can apply the remainder toward discretionary consumption. In the first case,

$$U(H_O) = (H_{max} - H_N - H_O)^{\alpha}\left(\frac{\omega H_O}{p}\right)^{\beta} \tag{6}$$

and maximising utility yields

$$H_O = \beta(H_{max} - H_N) \tag{7}$$

Total hours supplied by the household will be $H_O + H_N$, or

$$H = \beta H_{max} + \alpha H_N \tag{8}$$

Given that

$$H_N = \frac{pS_N - m}{\omega}, \tag{9}$$

we find

$$H = \beta H_{max} + \frac{\alpha}{\omega}\left(pS_N - m\right) \text{ for } H_N > 0. \tag{10}$$

Note that since $pS_N - m$ is positive, as ω increases H decreases, and as ω decreases H increases.

In the second case, $H_N = 0$, there will be a remainder of the non-wage funds that can be applied toward discretionary consumption, such that

$$U(H_O) = (H_{max} - H_O)^{\alpha}\left(\frac{\omega H_O}{p} + \frac{m}{p} - S_N\right)^{\beta}. \tag{11}$$

Utility maximising yields

$$H_O = \beta H_{max} - \frac{\alpha}{\omega}\left(m - pS_N\right). \tag{12}$$

Since $H_N = 0$,

$$H = \beta H_{max} - \frac{\alpha}{\omega}\left(m - pS_N\right) \text{ for } H_N = 0. \tag{13}$$

Note that since $m - pS_N$ is positive, H increases and decreases with ω, which differs from the $H_N > 0$ case. Thus we have two different hour response regimes to wage.

0.2.3 Labour is Sold Through the Market

The market aggregates the hours offered by the households into H_S, as

$$H_S = \sum_i^n H_i$$

and completes the labor market exchange such that H_M is the lesser of H_D and H_S. This market also determines hours worked per household, H_W, and aggregate effective effort, N.

$$H_W = \begin{cases} H & \text{for } H_S \leq H_D, \\ \dfrac{H_D}{H_S} H & \text{for } H_S > H_D \end{cases} \tag{14}$$

such that in the case of labour supply greater than labour demand, each household provides a proportional amount of labour to meet demand. Aggregate effective effort is then

$$N = \sum_i^n e_i H_{W,i}. \tag{15}$$

0.2.4 The Firm Produces Sugar

The quantity of sugar produced, S_S, is given by the production function of Equation 1 using for eH the N found by Equation 15 above.

0.2.5 The Households Plan Consumption

The households will buy the sugar they can afford above the necessary quantity, S_N, so

$$S_i = \max\left(S_N,\ \frac{1}{p}(\omega H_i + \tau_i + \min(0, m_i))\right) \tag{16}$$

and the aggregate demand is

$$S_D = \sum_i^n S_i. \tag{17}$$

0.2.6 Sugar is Sold Through the Market

The market chooses the lesser of S_S and S_D as the quantity of sugar exchanged. In the case of $S_S < S_D$, the sugar supplied by the firm is divided proportionately between the households and notated as S_C, sugar consumed.

$$S_C = \begin{cases} S & \text{for } S_D \leq S_S, \\ \dfrac{S_S}{S_D} S & \text{for } S_D > S_S. \end{cases} \tag{18}$$

Any unspent wages are saved for future consumption and added to m_i, and any unsold sugar is stored as inventory, I, which is unused in future steps. Households will incur debt if necessary in order to purchase S_N, which is represented by negative m values, though this debt does not affect future consumption.

0.2.7 The Firm Updates Expectations

The firm adjusts its neural net expectations of the wage and effort relationship by including an additional learning step given information it receives through the market interactions with the households.

The observed effort for the given wage is observed by

$$e_{obs} = \frac{S_S^{\frac{1}{\gamma}}}{H_M} \tag{19}$$

and the tuple (ω, e) is added to the training set for $\eta(\omega)$ and the neural net is refit with the updated information. The firm has a memory parameter, μ, which controls the number of observations maintained in the training set. After each update, the first element in the set is dropped to make room for the next observation.

The sugar demand at the offered price is observed as S_D, and the household updates the values of p or ω in order to match supply to demand. The heuristics used by the firm to modify p and ω values are based on both the observed sugar demand and the profit from the last market exchange,

$$\pi = pS_M - \omega H_M. \tag{20}$$

In its attempt to balance sugar supply and demand, the firm adjusts wage or price by a given percentage, *pct*. If $S_S < S_D$ and $\pi > 0$, the firm will increase wage by *pct* to entice more effort and hours. If $\pi < 0$, the firm will raise price to lower demand. If $S_S > S_D$ and $\pi > 0$, the firm will decrease price to increase demand, and if $\pi < 0$ the firm will decrease wage to reduce hours.

The neural net model and the logic updating price and wage values provide the firm with unique heuristics for incorporating observed aggregate responses from the households into useful relationships for its own decision making.

0.3 Model Initialisation

At the start of the simulations, households, η and S_D are initialised. Initial values for wage and price are specified as input parameters.

The fixed unique household parameters α, β and d are found by random draws of appropriate bounded distributions at the start of the model run.

The neural network expectation model for the firm is initialised by selecting at random a single household to provide values for e given a sample range of wages. From these values we construct the initial training set for $\eta(\omega)$. η is then trained starting with random weights. After initialisation, μ elements of the training set are retained for future updates.

Sugar demand is initalised by using the same randomly selected a household and calculating household demand with initial wage and price, then multiplying this value by the number of households.

0.4 Entities, State Variables and Scale

The model entities are a single firm, n households, and a market through which the firm and the households interact. All household values are stored as vectors indexed by household number, which allows for numerous opportunites to employ vectorised operations instead of iterating through all households. Table 1 gives the models parameters that are set at the start of the simulation and remain constant. Table 2 gives the model state values that update at varying times over the course of the simulation.

Parameter	Description	Values
ω_{max}	firm's maximum wage for initialising η	~ 100
μ	firm's memory length for updating η	$\mu \geq 3$
A	production function coefficient	3
γ	firm's production function exponent	$\gamma \geq 1$
pct	percentage change for wage and price adjustments	.1
H_{max}	maximum available household hours per step	400
S_N	amount of consumption necessary for household survival	$\sim 26,000$
e_{min}	minimum household effort	.2
d_{max}	maximum disutility of effort	5
d_i	disutility of effort for household i	$d_i > 1$
α_i	exponent of leisure term for utility function for household i	$\alpha = \beta - 1$
β_i	exponent of consumption term for utility function for household i	$\beta = \alpha - 1$

Table 1: Modern Wage Dynamics Parameter Values.

0.5 Process Overview and Scheduling

Each model step represents a month time duration, and iterative operations are conducted in consecutive household order. The market interactions are based on aggregate household outcomes, and a single household outcome does not impact another household during any given operation, though aggregate outcomes do affect all households over time.

We run the model by calling `main.py`, without any arguments. All parameters including output directory and filenames are specified in the `series_params.py` file. The `main.py` file also calls `initialisation_functions.py`, `firm_functions.py`, `household_functions.py` and `market_function.py`.

The process overview is:

```
main.py reads parameter sets from series_params.py
```

Variable	Description
η	neural network representing the expected effort response curve
$\hat{e}$	firm's expected aggregate household effort
ω	wage set by the firm
p	price of sugar set by the firm
H_D	quantity of hours demanded by the firm
π	firm profit
I	quantity of sugar in the firm's inventory
H_i	desired number of hours supplied by household i
H_S	aggregate number of hours supplied by the households
H_M	number of work hours exchanged in the labour market
$H_{W,i}$	number of hours worked by the household i
N	aggregate effective labour provided by the households
S_S	quantity of sugar the firm produces according to available effective labour
ι_i	wage income for household i
S_i	quantity of sugar demanded by household i
S_M	quantity of sugar exchanged in the sugar market
$S_{C,i}$	quantity of sugar consumed by household i
m_i	monetary holdings for household i
U_i	utility for household i

Table 2: Modern Wage Dynamics State Variables.

```
for each distinct parameter set:
   for each repetition of parameter set:
      assign parameter values
      initialise households
      initialise η
      initialise S_D
      save initial values
      for each step:
         firm decides hours demanded
         households decide effort and hours supplied
         market conducts labour exchange
         firm produces sugar with market labour
         households determine sugar demand
         market conducts sugar exchange
         firm calculates profit
         households calculate utility
         households update ledgers
         firm updates inventory
         firm updates η
         firm adjusts wage or price values
```

```
        save step values
    write dataframes to files
```

0.6 Considerations and Explorations

The model can currently be run with multiple parameter values and assumptions.

1. The model can be run without necessary consumption, in which case $S_N = 0$.
2. The model is currently coded to allow for numerous policy experiments, such as
   (a) minimum wage, ω_{min},
   (b) unemployment benefits, υ, which results in households having a reserve wage, ω_r,
   (c) earned income tax credits (EITC), τ,
   (d) and universal income supplement, σ.

   These policy experiment slightly modify the equations presented earlier. We add to our representation of a household's employment rent not only wage and unemployment benefits, but also any earned income tax credits (EITC) it receives to supplement its income. The household employment rent is therefore

$$r = \omega - \upsilon + \tau, \tag{21}$$

   where υ is the standard unemployment compensation the household expects if it loses employment, and τ is the earned income credit. The household's effort response curve is now

$$e_i = \begin{cases} 1 - \left(\dfrac{d_i}{\omega - \upsilon + \tau_i + d_i} \right)^{\frac{1}{2}} & \text{for } \omega > \omega_{r,i} \\ 0 & \text{for } \omega \leq \omega_{r,i} \end{cases} \tag{22}$$

   where

$$\omega_r = \frac{2\upsilon}{H_{max}}. \tag{23}$$

   Unemployment benefits are modeled as a constant quantity,[3] but τ is modeled as a tripartite function with an increasing, constant and decreasing range of supplementary income, shown in Figure 2.[4] All policy parameters need to be defined in per hour terms, so unemployment benefits, universal income supplement and EITC benefits need to be converted. We do this by dividing the empirical quantities by half of the household's maximum work hours to obtain a metric comparable to the wage the worker would obtain for those same hours. Thus household

[3] Median weekly unemployment benefits for 2019 for the US was \$450, or \$1800 per month. *Need source.

[4] From https://crsreports.congress.gov/product/pdf/R/R43805/11, we estimate annual 2020 EITC for a married household with two children as \$0 - \$15k of income receives increasing supplement from \$0 to \$6k, then constant supplement of \$6k through \$30k, then dropping to \$0 by \$55k.

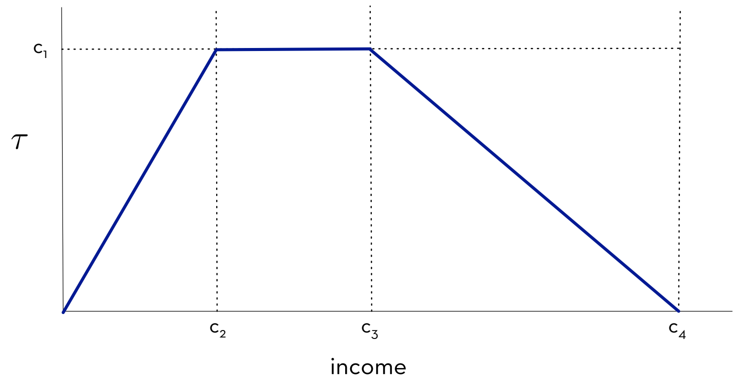

Figure 2: Earned income tax credit, τ, is modeled as a tripartite function with parameters c_1, c_2, c_3 and c_4.

EITC is represented in terms of parameters c_1, c_2, c_3 and c_4 and household income ι_i as

$$\tau_i = \begin{cases} \frac{c_1}{c_2}\iota_i & \text{for } 0 < \iota_i < c2, \\ c_1 & \text{for } c_2 \leq \iota_i \leq c_3, \\ -\frac{c_1}{c_4 - c_3}\iota_i + \frac{c_1 c_4}{c_4 - c_3} & \text{for } c_3 < \iota_i < c4 \\ 0 & \text{for } \iota_i \geq c_4. \end{cases} \tag{24}$$

The calculation of tribute hours is likewise modified to include τ and σ in the budget,

$$H_N = \max\left(0,\ \frac{pS_N - (m + \tau + \sigma)}{\omega}\right), \tag{25}$$

as well as household hours supplied,

$$H = \begin{cases} \beta H_{max} + \frac{\alpha}{\omega}\left(pS_N - (m + \tau + \sigma)\right) & \text{for } H_N > 0, \\ \beta H_{max} - \frac{\alpha}{\omega}\left(m + \tau + \sigma - pS_N\right) & \text{for } H_N = 0, \end{cases} \tag{26}$$

and household sugar demanded,

$$S_i = \begin{cases} \max\left(S_N,\ \frac{1}{p}(\omega H_i + \tau_i + m_i)\right) & \text{for } H_i > 0, \\ \max\left(S_N,\ \frac{1}{p}(\upsilon + m_i)\right) & \text{for } H_i = 0. \end{cases} \tag{27}$$

Tables 3 and 4 show additional parameters and state variables for policy implementations.

3. The model is very sensitive to production function parameters, A and γ. The parameter value for A is derived from making it possible that household could self-produce S_N with

Parameter	Description	Values
ω_{min}	firm's minimum wage	7
υ	monthly unemployment benefit	~ 500
$EITC$	flag for implementation of EITC policy	(0, 1)
c_1	parameter one of EITC function	6000
c_2	parameter two of EITC function	15,000
c_3	parameters three of EITC function	30,000
c_4	parameters four of EITC function	55,000
σ	universal income supplement	1200

Table 3: Policy Parameter Values.

Variable	Description
τ_i	amount of earned income tax credit for household i
$\omega_{r,i}$	reservation wage for household i

Table 4: Policy State Variables.

max hours, full effort and constant returns to scale, such that

$$A = \frac{S_N}{H_{max}}. \tag{28}$$

The model value for γ needs to be large enough that multiple households working tribute hours can produce S_N for all households, or

$$\gamma \geq \frac{\ln\left(\frac{nS_N}{A}\right)}{\ln(H_N)}. \tag{29}$$

How does the model behaviour change with A, γ and n? What meaning could these changes have? Is it possible to quantify these relationships formulaically?

4. How does the model behavoiur change if the firm uses existing inventory to meet demand, thus decreasing production and hours demanded if $I > 0$?

There are also a variety of model modifications and extensions that would be of value to explore.

1. The model can be run without efficiency wages with minor modifications to restrict $N = H_M$ or e and $\hat{e} = 1$.

2. The single representative firm could have a different goal other than matching sugar supply to demand, such as employing all labour hours supplied, or maximising profit.

3. The firm's expectation of household effort is modeled as a neural network, η, but its expectation of demand given price and wage is simply the last observed value. We could also implemented the firm's demand expectation as a neural net, but with two input parameters, such that $\rho(p, \omega) = \hat{S}_D$.

4. The US is understood to have a 'dual economy,' where there are two distinct classes of wages: one high and with high effort and high returns to labour, and one low with low effort and low labour returns. This could implemented perhaps by redefining the productivity function such as
$$Y = A_h(e_h H_h)^{\gamma_h} + A_l(e_l H_l)^{\gamma_l} \tag{30}$$
so where $\gamma_l < \gamma_h$, $A_l < A_h$, and effort response curve $e(\omega)_h$ lies above $e(\omega)_l$.

0.7 Inputs and Outputs

The model reads simulation parameter values from the `series_params.py` and sends simulation results to a designated `.csv` file. Input parameters are described in Table 5, and some parameters accept multiple values to allow for numerous parameter sets within the same series.

Parameter	Description	value
`directory`	name of output file directory	single
`series_name`	filename for series results	single
`verbose`	true or false, whether the model prints variables each step	single
`seed`	rng seed for replicable results	single
`t_max`	the number of time steps the model will run	single
`omega_max`	firm's maximum wage for initialising η	single
`H_max`	maximum available household hours per step	single
`mu`	firm's memory length for updating η	single
`pct`	percentage change for wage and price adjustments	single
`e_min`	minimum household effort	single
`c1`	parameter one of the earned income tax credit function	single
`c2`	parameter two of the earned income tax credit function	single
`c3`	parameter three of the earned income tax credit function	single
`c4`	parameter four of the earned income tax credit function	single
`n`	number of households	multi
`omega_0`	initial wage value	multi
`A`	value of production function coefficient	multi
`gamma`	value of production function exponent	multi
`EITC`	flag for implementation of EITC policy	multi
`upsilon`	unemployment benefit	multi
`omega_min`	minimum wage	multi
`sigma`	universal income supplement	multi
`S_N`	required household consumption in units of sugar	multi
`p_0`	initial price value	multi
`d_max`	maximum household disutility for effort	multi

Table 5: Input parameters specified in the `series_params.py` file.

For each time step the model saves the following simulation parameter and variable values to a dataframe: `set`, `run`, `step`, `n`, `omega_max`, `H_max`, `A`, `gamma`, `mu`, `pct`, `e_min`, `EITC`, `upsilon`, `omega_min`, `sigma`, `S_N`, `d_max`, `omega_0`, `p_0`, `I`, `pi`, `total_pi`, `omega`, `p`, `e_hat`, `H_D`, `S_S`, `S_P`, `perceived_effort`, `household`, `d`, `omega_r`, `alpha`, `beta`, `e`, `H_N`, `H`, `H_W`, `m`, `S`, `S_C`, `income`, `tau`, `U`, `expenditure`, `N`, `H_S`, `H_M`, `S_D`, `S_M`. The variable *set* is the specific combination of parameter values, and *run* is the iteration of those values. The model also saves the state of η at each time step described by: `set`, `run`, `step`, `omega`, `effort`.

0.8 More on the Neural Networks

We model the firm's expectation of effort as a neural network, an unspecified non-linear model that is trained with known (x, y) tuples, and once trained acts as a black box predictor of y given x, illustrated in Figure 3.

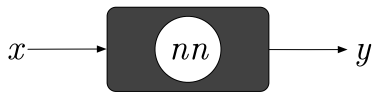

Figure 3: A neural network serves as a black box function, predicting y from a given x.

Inside the black box are a series of weights (W), biases (b) and activation functions. Each layer (A) can have numerous nodes. Figure 4 illustrates a very simple regression neural network with a single hidden layer with two nodes, and with sigmoid and identity activation functions. The sigmoid activation function incorporates an element of non-linearity which allows for very flexible regression training.

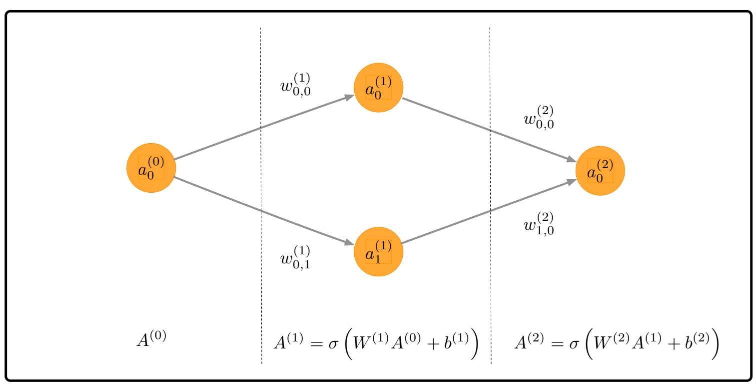

Figure 4: Inside the black box: a basic neural network with an input layer (x), a single hidden layer with two nodes, and an output layer (y). The first and second activation functions are the σ function. This model contains six tunable parameters.

In general, the values of each of the nodes in layer $n + 1$ are found using

$$A^{(n+1)} = f^{(n+1)}\left(W^{(n+1)}a^n + b^{(n+1)}\right).$$

In the case of the single neural network shown in Figure 4,

$$W^{(1)} = \begin{bmatrix} w^{(1)}_{0,0} \\ w^{(1)}_{0,1} \end{bmatrix}, \quad \text{and} \quad W^{(2)} = \begin{bmatrix} w^{(2)}_{0,0} & w^{(2)}_{1,0} \end{bmatrix}$$

Putting this all together,

$$A^{(1)} = \sigma\left(\begin{bmatrix} w_{0,0}^{(1)} \\ \\ w_{0,1}^{(1)} \end{bmatrix} a_0^{(0)} + \begin{bmatrix} b^{(1)} \\ \\ b^{(1)} \end{bmatrix}\right) = \begin{bmatrix} \sigma(w_{0,0}^{(1)} a_0^{(0)} + b^{(1)}) \\ \\ \sigma(w_{0,1}^{(1)} a_0^{(0)} + b^{(1)}) \end{bmatrix}$$

and

$$A^{(2)} = \sigma\left(\begin{bmatrix} w_{0,0}^{(2)} & w_{1,0}^{(2)} \end{bmatrix} \begin{bmatrix} \sigma(w_{0,0}^{(1)} a_0^{(0)} + b^{(1)}) \\ \\ \sigma(w_{0,1}^{(1)} a_0^{(0)} + b^{(1)}) \end{bmatrix} + \begin{bmatrix} b^{(2)} \end{bmatrix}\right)$$

$$= \sigma\left(w_{0,0}^{(2)}\, \sigma\left(w_{0,0}^{(1)}\, a_0^{(0)} + b^{(1)}\right) + w_{1,0}^{(2)}\, \sigma\left(w_{0,1}^{(1)}\, a_0^{(0)} + b^{(1)}\right) + b^{(2)}\right).$$

The learning consists of updating values for the six parameters: $w_{0,0}^{(1)}$, $w_{0,1}^{(1)}$, $w_{0,0}^{(2)}$, $w_{1,0}^{(2)}$, $b^{(1)}$ and $b^{(2)}$. In the specific instance of $\eta(\omega)$,

$$\hat{e} = \eta(\omega) = \sigma\left(w_{0,0}^{(2)}\, \sigma\left(w_{0,0}^{(1)}\, \omega + b^{(1)}\right) + w_{1,0}^{(2)}\, \sigma\left(w_{0,1}^{(1)}\, \omega + b^{(1)}\right) + b^{(2)}\right)$$