Surrogate modelling of a recursive-dynamic single country computable general equilibrium model

Wolfgang Britz, Hugo Storm
University of Bonn

GTAP Virtual Seminar, Sep 8, 2026

UNIVERSITÄT BONN

AGENDA

Part 1

- Machine Learning in Agricultural Economics
- Deep Learning properties and promises

Part 2

- Optimizing policy instruments based on CGE model analysis
- Code Demo
- Take-home message

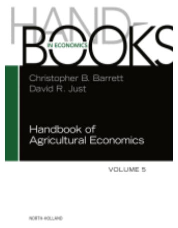
MACHINE LEARNING IN AGRICULTURAL ECONOMICS

Application areas

1. Econometrics
2. Simulation Models

Deep Learning key properties and promises

1. Representation Learning
2. Flexible function form / Universal function approximator



Baylis, Kathy, Thomas Heckelei, and Hugo Storm 2021. “Chapter 83 - Machine Learning in Agricultural Economics.” In Handbook of Agricultural Economics, edited by Christopher B. Barrett and David R. Just, 5:4551–4612. Elsevier. https://doi.org/10.1016/bs.hesagr.2021.10.007

Storm, Hugo, Kathy Baylis, and Thomas Heckelei. 2019. Machine Learning in Agricultural and Applied Economics. European Review of Agricultural Economics 47 (3): 849–92. https://doi.org/10.1093/erae/jbz033

UNIVERSITÄT BONN

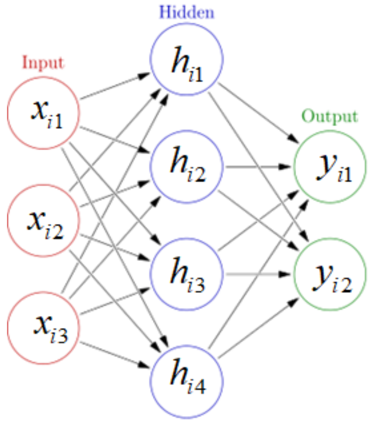
NEURAL NETWORKS STRUCTURE AND NOTATION

- NN consist of several ***layers***
- Each layer has several ***neurons***
- Also called: “Feedforward neural network”, ”fully connected/dense NN”, “Vanilla neural network”

$$y = f(\boldsymbol{x}) = f^{(3)}(f^{(2)}(f^{(1)}(\boldsymbol{x})))$$

$$f^{(k)}(x) = \boldsymbol{h}^{(k)} = g^{(k)}(\boldsymbol{W}^{(k)\top}\boldsymbol{h}^{(k-1)} + \boldsymbol{b}^{(k)})$$

$$\boldsymbol{h}^{(0)} = \boldsymbol{x}$$



$$\underset{(4\times1)}{h_i} = \sigma^{[1]}\left(\underset{(4\times3)}{W^{[1]}}\underset{(3\times1)}{x_i} + \underset{(4\times1)}{b^{[1]}}\right)$$

$$\underset{(2\times1)}{y_i} = \sigma^{[2]}\left(\underset{(2\times4)}{W^{[2]}}\underset{(4\times1)}{h_i} + \underset{(2\times1)}{b^{[2]}}\right)$$

Possible activation functions

$$\sigma(z) = relu(z) = max\{0, z\}$$

$$\sigma(z) = sigmoid(z) = \frac{1}{1+e^{-z}}$$

$$\sigma(z) = identity(z) = z$$

DEEP NEURAL NETWORKS

- A deep neural network is a neural network with many hidden layers
- Such networks are much more flexible and can learn more complex non-linear relationships
- With increasing size of the network the number of parameters increases and training becomes more difficult
- Modern tools allow highly efficient training using gradient-based optimization

Example LLMs:

- GPT 3: 96 layers, 175.0B parameters
- GPT 4: >10 trillion parameters

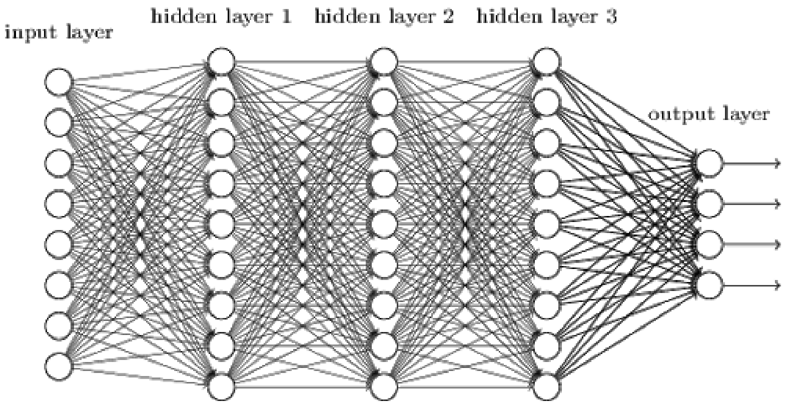


Image: http://neuralnetworksanddeeplearning.com/chap6.html

USE CASES IN AGRICULTURAL ECONOMICS

For Econometrics

- Making use of novel data sources (particularly images, audio, text)
- Causal Machine learning (functional form flexibility)

For Simulation Models

- Post-model result analysis (Lengers & Britz 2014)
- Model integration (Shang et al. 2024, Britz & Hertel 2011)
- Calibration (Krisztin & Ringwald 2025)
- (Policy) Optimization (Britz & Hertel 2011; Britz & Storm 2026)

Lengers, B., Britz, W., Holm-Müller, K. 2014. What drives marginal abatement costs of greenhouse gases on dairy farms? A meta-modelling approach. *Journal of Agricultural Economics*, 65(3): 579-599.

Britz, W., Hertel, T.W. 2011. Impacts of EU biofuels directives on global markets and EU environmental quality: An integrated PE, global CGE analysis. *Agriculture, Ecosystems & Environment*, 142(1-2): 102-109

Shang, L., Wang, J., Schäfer, D., Heckelei, T., Gall, J., Appel, F., Storm, H. (2024). Surrogate modelling of a detailed farm-level model using deep learning. *Journal of Agricultural Economics*, 75:235–260.

Krisztin, T., Ringwald, L. 2025. Calibration of Economic Land-Use Models with Surrogate Approaches: Enhancing the GLOBIOM Model, Presentation XVIII EAAE Congress, August 25th - 29th 2025, Bonn, Germany.

Britz, W., Storm, H. (2026). Surrogate modelling of a recursive-dynamic single country computable general equilibrium model. *Journal of Global Economic Analysis*, 11(01), 49-85.

DEEP LEARNING REQUIREMENTS

For Econometrics

- Making use of novel data sources
- Causal Machine learning

=> Deep Learning only makes sense with large datasets

For Simulation Models

- Post-model result analysis
- Model integration
- Calibration
- (Policy) Optimization

=> Generate large datasets based on sensitivity analysis

POLICY OPTIMIZATION

- Existing applications in literature:
  - Take CGE model's equations as constraints, render some taxes endogenous (more variables than constraints) and optimize objective function
  - Few tax rates, comparative-static or rec-dyn with some periods
- Direct policy optimization over long-time periods with many instruments new:
  - Computationally challenging: many equations, highly non-linear
  - Impossible without code changes if model is solved year by year

=> Deep learning as promising alternative

DATA GENERATION

- Successful training of a larger NN requires many observations, each generated by a model run. Effective sampling strategy is key:
  - Inputs ranges of the inputs according to intended use of meta-model, specific for optimization: drop observations with objective below a base case
  - Stratified random sampling such as Latin Hypercube Sampling (LHS) to cover effectively the input range
- Computing intensive:
  - Exploit options for parallel execution including use of multiple machines
  - Only store required inputs and outputs for training to safe disk space

SHORT OVERVIEW ON MANAGE-WB

- Open-source recursive-dynamic single-country CGE model realized in GAMS, fully documented, developed by team at World Bank, used in many different projects
- Flexible nesting structures for production functions, final demands and factor markets, AEZ implementation, GHG emission accounting and taxing, multiple households, government account with debt dynamics, different closures for B.o.T.
- Modular, on demand extensions such as capital vintages, income dependent parameters of the CDE demand function, endogenous technical progress, abatement curves for process emissions
- Data (SAM), parameters, nesting structure inputted via an EXCEL workbook. Bottom up or generated automatically from different version of the GTAP Data base (Africa 3 as a free option, GTAP CE or Power)
- Graphical User Interface: model set-up, result exploitation (graphs, tables, maps across multiple scenarios)

TRAINING DATA GENERATION WITH MANAGE-WB

- Sensitivity analysis with user interface built into MANAGE-WB:
  - Variables (parameters, shocks) to consider
  - Their ranges and distributions to use
- LHS based stratified random sampling
- Automated execution of the desired number of draws (=model runs), parallel execution supported
- Results automatically combined into a GDX container

=> Demo

TRAINING

- Python as a widely used option where powerful packages are available, include support for GPUs for large-scale applications
- Supplementary material comprises script developed:
  - Arguments are documented in annex, details also installation
  - Allow to adjust NN structure (# of hidden layers and # of neurons) and to choose non-default settings for training process
  - Lists of inputs and outputs and # of observations read along with data from two CSV files, here generated by GAMS code merging samples from sensitivity analysis
  - Step-wise training process with varying training parameters
  - Results stored to GDX file for interaction with GAMS

=> Demo

IMPLEMENTATION OF NN AND OPTIMIZATION

- NN structure simple enough to be easily coded in different software languages
- We used an implementation in GAMS:
  - Eases to add objective function and additional constraints for optimization
  - Access to performant solvers
  - Python code writes out GDX container with all information for GAMS code to replicate any NN (# of layers, # of neurons, estimated parameters, mapping sets to link NN structure to three dimensional input variables, factors used for normalization)
- GAMS code in annex:
  - Implementation of NN generic
  - Code for optimization (objective function, additional constraints) application specific

APPLICATION CASE

- Illustrative application, not thought to produce policy relevant results
- Reduce climate change emissions - both from CO2 and Non-Co2 - in Tanzania compared to a baseline as an example:
  - Cumulative emissions over time targeted
    <= impact on climate depends not only on emissions in one future year, but on all
  - Policy choices:
    - Distribution of reductions over time, i.e., yearly emission ceilings. Determines yearly endogenous emission tax.
    - Carbon tax recycling: shares of factor taxes, direct taxes, sales taxes or government savings

APPLICATION CASE: OBJECTIVE FUNCTION

- Summing real GDP changes over time not a good indicator: comprise changes in investment demand that drive capital accumulation ⬄ implicit double counting

⇒ Sum over time of changes in private and government consumption per capita instead, normalized in each year with results of baseline

  - Per-capita removes impact of population growth
  - Implicit dis-counting as real GDP per capita increases over time
  - Compared to some economy-wide utility function with explicit dis-counting rate easier to communicate

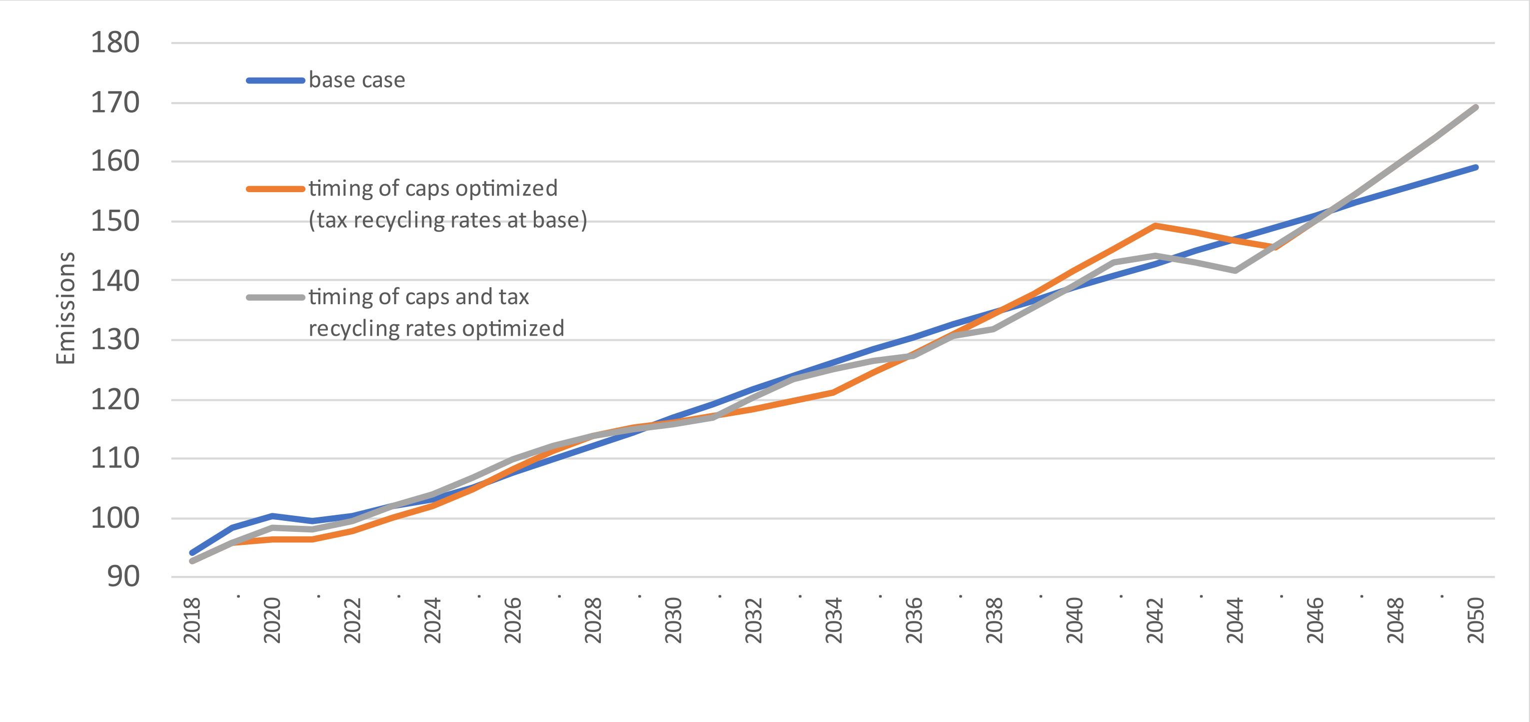
=> Brief look at code and demo run

APPLICATION: CHALLENGES

- Inter-temporal optimization of policy instruments but no agent in CGE model forward looking
- Tax adjustments could be beneficial independent without carbon taxes
  - Argument against based on "double-dividend hypotheses": Carbon tax revenues open window of opportunity for tax reforms
- Some reduction pathways yield infeasible solutions
  - These input combinations removed from training set and later optimization

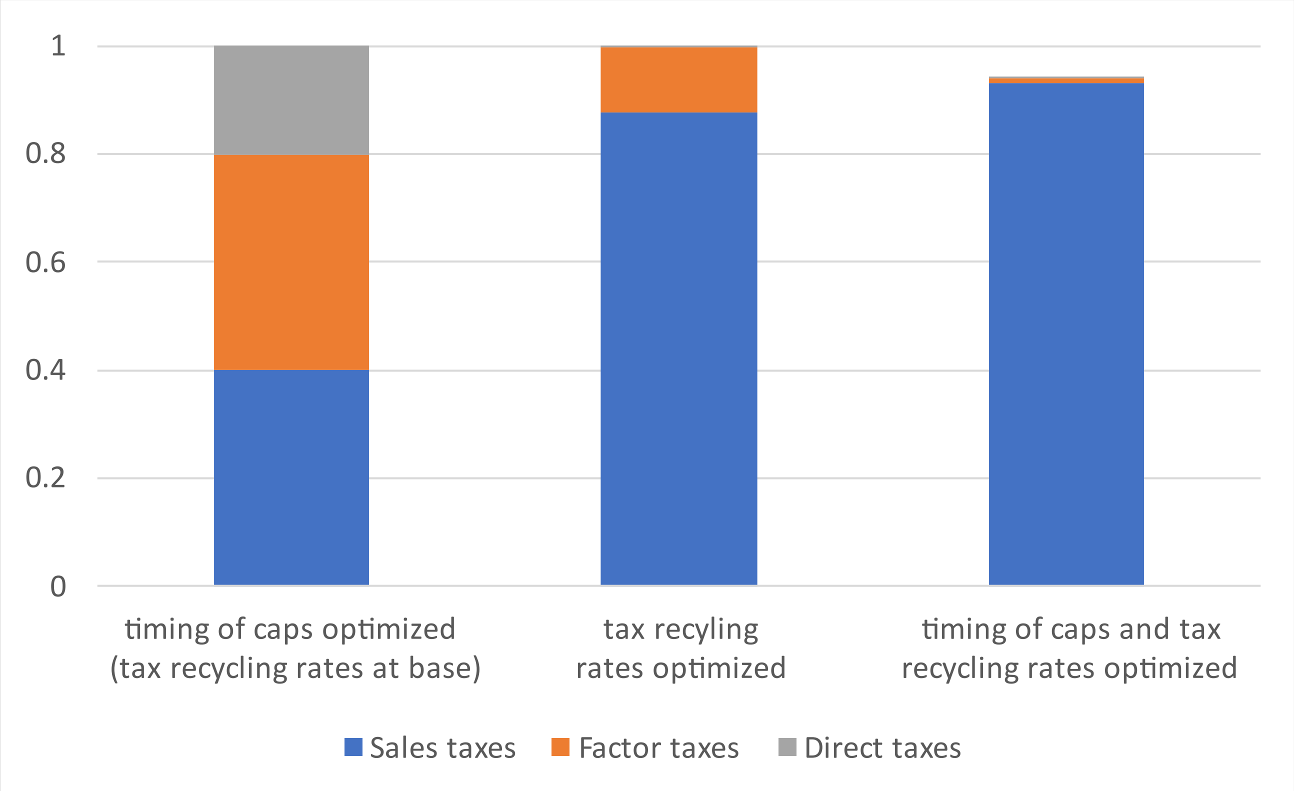
APPLICATION: MAIN FINDINGS

- Larger reduction of emissions compared to base in later periods beneficial:
  - Large share of hard-to-abate process emissions in agriculture. Their share drops over time due to structural change
  - Technical progress also reduces per unit emission factors and cost over time



APPLICATION: MAIN FINDINGS

- Optimal tax recycling strategy
  - Focuses on sales taxes
  - Not all revenues recycled to stabilize government budget



TAKE HOME MESSAGES

- NN seem able to replicate the simulation behavior of a CGE to any desired accuracy
- Can be easily implemented in different programming languages
- Training requires many simulation runs, clever design of considered input ranges important
- NN proven useful for model-coupling and optimization of policy instruments, potential game changer to improve parameterization in back-casting exercises
- Annex of paper comprises quite generic python code and an implementation in GAMS

Britz, W., & Storm, H. (2026). Surrogate modelling of a recursive-dynamic single country computable general equilibrium model. *Journal of Global Economic Analysis*, *11*(01), 49-85. https://doi.org/10.21642/JGEA.110102AF

UNIVERSITÄT BONN