Week 4 — Deep Equilibrium Networks I

The DEQN Principle & the Brock–Mirman Growth Model

Dr Juan Zurita

Deep Learning for Macroeconomics
The University of Edinburgh · School of Economics

From VFI to DEQNs

In PNM you solved the growth model by **value-function iteration** on a grid.

DEQN idea: parameterise the policy function as a neural network and train it to satisfy the equilibrium conditions.

Azinovic, Gaegauf & Scheidegger (2022)

"Deep Equilibrium Nets" — *International Economic Review*

The DEQN Recipe

1. **Define** the economic model (states, controls, equilibrium conditions)
2. **Parameterise** policy functions: $c = \mathcal{N}_\theta(k)$
3. **Construct loss**: squared Euler-equation residuals
4. **Train**: minimise the loss with Adam
5. **Verify**: compare with known solutions or refined grids

No grid, no discretisation of the state space — just sample points.

Brock–Mirman Model

$\max \sum_{t=0}^{\infty} \beta^t \ln c_t \quad \text{s.t. } k_{t+1} = k_t^{\alpha} - c_t$

Euler equation: $\frac{1}{c_t} = \beta \frac{\alpha k_{t+1}^{\alpha-1}}{c_{t+1}}$

Why this model?

It has a **closed-form solution**: $c_t = (1 - \alpha\beta)\, k_t^{\alpha}$.

We can verify the neural network against the true answer.

Loss Function Design

The Euler-equation residual at a sample point k:

$$R(k;\boldsymbol{\theta}) = 1 - \beta \, \frac{\alpha k'^{\alpha-1} \cdot c(k;\boldsymbol{\theta})}{c(k';\boldsymbol{\theta})}$$

where $k' = k^{\alpha} - c(k;\boldsymbol{\theta})$.

Loss: $\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^{N} R(k_i;\boldsymbol{\theta})^2$

Train until $\mathcal{L} < 10^{-8}$ — machine-precision accuracy.

Summary

- DEQNs replace grid-based VFI with neural-network policy functions
- The loss function encodes economic equilibrium conditions
- Brock–Mirman provides a verifiable first test case
- The method is grid-free and scales to higher dimensions

Next week: adding uncertainty — stochastic models with Gauss–Hermite quadrature.