

Week 3 — Automatic Differentiation for Economics

Forward Mode, Reverse Mode & PyTorch Autograd

Dr Juan Zurita

Deep Learning for Macroeconomics

The University of Edinburgh · School of Economics

Why Differentiation Matters

Economic models are systems of equations involving derivatives:

- Euler equations: $u'(c_t) = \beta \mathbb{E}_t[u'(c_{t+1})(1 + r_{t+1})]$
- FOCs of optimisation problems
- Jacobians for stability analysis

Automatic differentiation computes exact derivatives — not numerical approximation, not symbolic algebra.

Three Ways to Differentiate

Method	Pros	Cons
Symbolic	Exact, interpretable	Expression swell
Numerical	Simple (finite diff.)	Approximation error
Automatic	Exact, efficient	Requires code graph

Autodiff decomposes any program into elementary operations and applies the chain rule to each.

Forward vs Reverse Mode

Forward mode: propagate derivatives input $\rightarrow$ output.

Cost: one pass per input variable. Good when $n_{\text{inputs}} \ll n_{\text{outputs}}$.

Reverse mode (backpropagation): propagate derivatives output $\rightarrow$ input.

Cost: one pass per output variable. Good when $n_{\text{outputs}} \ll n_{\text{inputs}}$.

Key insight

Neural networks have **one** scalar loss and **millions** of parameters $\Rightarrow$ reverse mode is vastly more efficient.

In practice

`torch.autograd` builds a computational graph and computes gradients automatically.

Key functions:

- `torch.autograd.grad()` — compute $\partial y / \partial x$
- `torch.autograd.functional.jacobian()` — full Jacobian matrix
- `torch.autograd.functional.hessian()` — second derivatives

This is what makes DEQNs possible: we can differentiate through the entire economic model.

Summary

- Autodiff gives **exact** derivatives at machine precision
- Reverse mode is efficient for scalar-output functions (losses)
- PyTorch autograd handles everything behind the scenes
- We can differentiate Euler equations, value functions, and equilibrium conditions

Next week: Deep Equilibrium Networks — using all of this to solve dynamic models.