Week 7 — Physics-Informed Neural Networks

Continuous-Time Economics & the HJB Equation

Dr Juan Zurita

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

From Discrete to Continuous Time

Many economic models are naturally continuous-time:

$$\rho\, V(x) = \max_c \left\{ u(c) + V'(x)\, \mu(x, c) + \tfrac{1}{2}\, V''(x)\, \sigma^2(x) \right\}$$

This is the **Hamilton–Jacobi–Bellman** (HJB) equation.

Traditional approach: finite-difference grids. PINN approach: train a neural network to satisfy the PDE.

The PINN Principle

Raissi, Perdikaris & Karniadakis (2019)

Parameterise the solution as $V(x) = \mathcal{N}_{\boldsymbol{\theta}}(x)$ and minimise the PDE residual.

$$\text{Loss} = \underbrace{\|PDE\ \text{residual}\|^2}_{\text{interior}} + \underbrace{\|BC\ \text{residual}\|^2}_{\text{boundary}}$$

Derivatives V', V'' are computed via **autodiff** — no finite differences needed.

Cake-Eating as an HJB

$\rho\, V(W) = \max_c \{u(c) + V'(W) \cdot (-c)\}$

FOC: $u'(c) = V'(W) \Rightarrow c^* = (V'(W))^{-1/\gamma}$

Substitute back: $\rho\, V = u(c^*) - c^*\, V'$

Boundary condition

$V(0) = 0$ (no cake left $\Rightarrow$ no value).

The PINN learns $V(W)$ that satisfies both the HJB and the boundary.

Soft vs Hard Boundary Conditions

Soft BC: include boundary violations in the loss (weighted penalty).

Hard BC: embed the boundary in the architecture: $V(W) = W \cdot \mathcal{N}_{\boldsymbol{\theta}}(W)$.

Hard BCs guarantee satisfaction by construction. Soft BCs are more flexible but may violate boundaries during training.

For economics: hard BCs are preferred when boundary conditions have clear economic meaning.

Summary

- PINNs solve PDEs by training neural networks on residuals
- Autodiff provides exact spatial derivatives — no grids needed
- The HJB equation is the natural target for continuous-time models
- Hard boundary conditions embed economic constraints directly

Next week: heterogeneous agents — where deep learning truly shines.