Week 2 — Deep Learning Fundamentals

Gradient Descent, Backpropagation & Training Neural Networks

Dr Juan Zurita

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

The Learning Problem

Given data or a known function, find network parameters $\boldsymbol{\theta}$ that minimise a loss function:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$$

In economics: the "data" is often **equilibrium conditions** rather than observed samples.

Gradient Descent

Update rule: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}$

- α: learning rate — too large diverges, too small crawls
- **SGD**: use a random mini-batch instead of all data
- **Adam**: adaptive learning rates per parameter — the workhorse

From PNM

You already know gradient descent and Newton's method. Adam is an adaptive extension that works well for neural networks.

Backpropagation

Backprop = chain rule applied systematically through the computational graph.

For a composition $f = f_L \circ f_{L-1} \circ \cdots \circ f_1$:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_\ell} = \frac{\partial \mathcal{L}}{\partial f_L} \cdot \frac{\partial f_L}{\partial f_{L-1}} \cdots \frac{\partial f_{\ell+1}}{\partial f_\ell} \cdot \frac{\partial f_\ell}{\partial \boldsymbol{\theta}_\ell}$$

PyTorch computes this automatically via `loss.backward()`.

Activation Functions

Name	Formula	Use case
ReLU	$\max(0, x)$	Default hidden layers
Sigmoid	$1/(1+e^{-x})$	Output in $[0, 1]$
Tanh	$(e^{x}-e^{-x})/(e^{x}+e^{-x})$	Output in $[-1, 1]$
Softplus	$\ln(1+e^{x})$	Smooth ReLU

For economics: **softplus** is useful when outputs must be positive (e.g. consumption).

Overfitting and Regularisation

- **Overfitting**: the network memorises training points but generalises poorly
- **Weight decay** (L_2 regularisation): penalise large weights
- **Dropout**: randomly zero out neurons during training
- **Early stopping**: halt when validation loss stops improving

In economics, overfitting is less of a concern when the "data" comes from equilibrium conditions — but training stability still matters.

Next week: automatic differentiation — the engine behind backpropagation.