AI for Economic Research: Dynamic Models, Language, and Agents
Lecture 3.1: Why ML for Economists & Pattern Recognition

Zhigang Feng

2026-07-07 10:05:10

Lecture 3 in One Map

One lecture, four decks, one goal: make neural networks a working tool for economists — why they matter, what they are, how they learn, and how to make them respect economics.

you are here ▼

3.1 WHY — the case for ML in econ → **3.2 BUILD** — neurons · activations · UAT → **3.3 TRAIN** — backprop · SGD · dbl. descent → **3.4 SHAPE** — ICNN · monotone · lab

3.1 WHY	3.2 BUILD	3.3 TRAIN	3.4 SHAPE
the case for ML in econ	neurons · activations · UAT	backprop · SGD · dbl. descent	ICNN · monotone · lab
why should an economist care about ML?	*what is a neural network, and why does it work?*	*how does it learn — and why not overfit?*	*how do I build economics in — and run it myself?*
REPRESENT (UAT · Barron)		OPTIMIZE (loss · gradient)	GENERALIZE + CONSTRAIN

upstream: Lec01–02 — AI as cognitive capital; deep learning as the engine of the predictive stack

downstream: Lec04–06 — these networks become Bellman/Euler solvers for dynamic macro models

Act 3.1 (here) *makes the case for ML in economic research — unstructured data, better prediction and controls, high-dimensional models — then builds the core intuition with pattern recognition: digits live on a low-dimensional manifold, and networks learn the encoder.*

Why ML for Economists?

Four doors ML opens: unstructured data, causal inference, forecasting, and high-dimensional models

Harnessing the Power of Unstructured Data

- Economics has traditionally relied on structured, tabular data. However, a wealth of economic information is locked in unstructured formats: text, images, satellite imagery.
- ML algorithms, particularly deep learning, provide tools to systematically process and quantify this unstructured data into economic indicators.
- **Applications:**
  - Measuring economic sentiment and uncertainty from financial news and social media
  - Analyzing satellite images to track economic activity (nighttime lights $\approx$ GDP proxy, ships in ports $\approx$ trade flows)
  - Using image recognition on product pictures to understand non-standardized product features
- **Course arc:** Lec. 1 T1 framed AI as cognitive capital — the *why*. This lecture delivers the *what* (NN architectures, training algorithms); Lec. 4 delivers the *how* (solving dynamic equilibrium models).

Extracting Insights from Textual Data

- Natural Language Processing (NLP) offers powerful techniques to analyze large volumes of text — going beyond simple word counts.
- **Applications:**
  - Gauging dovish or hawkish stance of central bankers from public communications
  - Identifying key risk factors from corporate annual reports
  - Quantifying the evolution of economic policy uncertainty from news articles
- **For macro:** FOMC minutes, Beige Books, and congressional records contain rich information that traditional econometrics cannot easily exploit.

Enhancing Causal Inference and Forecasting

- **Causal Inference:**
    - "Double Machine Learning" allows flexible inclusion of high-dimensional controls, reducing omitted variable bias
    - ML identifies *heterogeneous treatment effects*: how does a policy impact vary across subgroups?
- **Forecasting & Nowcasting:**
    - ML captures complex, non-linear relationships that time-series models miss
    - Regularization (e.g., LASSO) enables "kitchen sink" regression without overfitting
    - Nowcasting: real-time economic indicators from credit card transactions, web searches, satellite imagery

Solving Large-Scale Macro and Finance Models

- Cutting-edge models (heterogeneous agents, portfolio choice) feature:
  - High dimensionality and complex non-linear dynamics
  - Traditional grid-based methods hit the **curse of dimensionality**
- ML techniques provide powerful new numerical methods:
  - **Deep learning:** parameterize policy/value functions with neural networks
  - **Reinforcement learning:** solve sequential decision problems via simulation
- **Examples:**
  - Optimal fiscal/monetary policy in heterogeneous agent models
  - Life-cycle portfolio choice with realistic income uncertainty
  - Krusell–Smith (1998) with neural network approximation of the law of motion

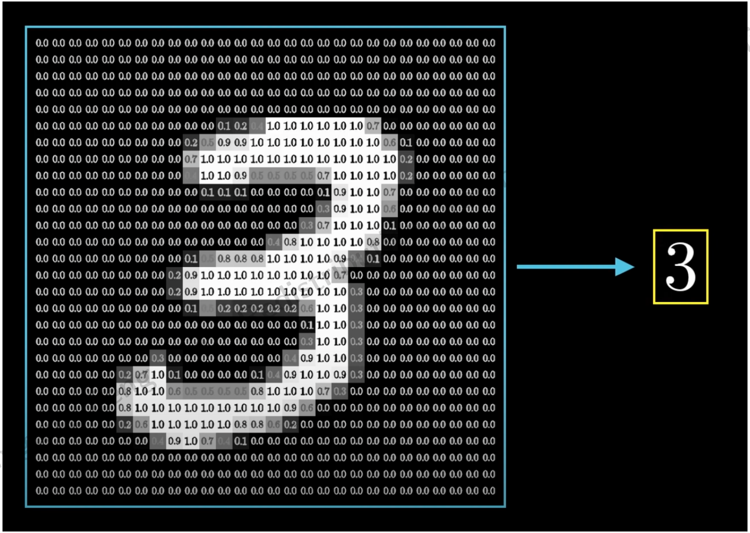
ML and the Curse of Dimensionality in Macro

- **CoD #1: Dimension of the state space**
  - $k \to \{k, a, h, \ldots\}$; the distribution Γ is infinite-dimensional
  - *Traditional:* sparse grids, local approximation — scale poorly
  - *ML:* neural networks are **grid-free**; recover equilibrium functions via Monte Carlo sampling
- **CoD #2: Unfolding of uncertainty**
  - $\mathbb{E}$ over future contingencies; the space for Γ is unknown
  - *Traditional:* discretize shocks, impose Markov structure
  - *ML:* focus on the ergodic distribution; direct simulation Monte Carlo
- **Key insight:** ML shifts the computational bottleneck from *grid construction* to *function representation* — a much more scalable paradigm.
- **Where this gets concrete in the course:** Lec. 4 T1 applies it to the optimal-growth state (k, z); Lec. 5 T1 generalizes to MDPs (S, A, P, R, γ); Lec. 6 attacks high-dim Aiyagari and Krusell–Smith with the methods you'll learn in T2–T4.

Pattern Recognition as Motivating Illustration

Digits, manifolds, encoders — the intuition behind every network in this course

Example: Handwritten Digit Recognition



Digit Recognition as a Mathematical Problem

- **Dataset:** $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ with images $\mathbf{x}_i \in [0,1]^{784}$ and labels $y_i \in \{0, \ldots, 9\}$
- **Input–Output Mapping:**

$$f_\theta : \mathbf{x} \longmapsto \hat{\mathbf{p}} \in \Delta^9, \qquad \hat{y} = \arg\max_k \hat{p}_k$$

  where Δ^9 is the 10-dimensional probability simplex

- **Learning Objective (Empirical Risk + Regularization):**
  - $$\min_\theta \; \frac{1}{N} \sum_{i=1}^{N} \underbrace{\mathcal{L}\big(f_\theta(\mathbf{x}_i), y_i\big)}_{\text{cross-entropy}} + \lambda \|\theta\|_p$$
- **Search Strategy:** Stochastic gradient descent & variants (Adam, RMSProp)

Constraints and Structure in Handwritten Digits

- **Anatomical constraints:** Muscles, tendons, and joints dictate possible strokes — leading to systematic patterns in how digits are formed.
- **Geometric properties:** Each digit has defining features — '0' has a closed loop, '1' a vertical stroke, '8' two stacked loops.
- **Spatial relationships:** Relative positions, sizes, and orientations of components maintain digit identity even with individual variation.
- **Lesson for economists:** Just as economic theory constrains the space of plausible models, structural constraints on handwriting reduce the effective dimensionality of the recognition problem.
- ML algorithms can **learn** these constraints from data — data augmentation, regularization, and architecture choices all exploit this structure.

From 784 Pixels to a Low-Dimensional Manifold

- **Manifold Hypothesis:** Natural digit images occupy a smooth, d-dimensional manifold $\mathcal{M} \subset \mathbb{R}^{784}$ with $d \ll 784$.
- **Plain-English version:** only a few "writing-style" degrees of freedom matter: slant, stroke thickness, loop closure, and local translation.
- The other $780+$ pixel directions are nearly empty: random pixel noise almost never looks like a digit.
- **Macro analogy:** finding a low-dimensional state in a high-dimensional model is the same problem. A good representation preserves the few variables that matter for the next decision.

Encoder View of Digit Recognition

- **Two-stage model:**

$$\mathbf{x} \xrightarrow{\pi:\mathbb{R}^{784}\to\mathbb{R}^d} \mathbf{z} \xrightarrow{g_\phi:\mathbb{R}^d\to\Delta^9} \hat{\mathbf{p}}, \qquad \hat{y} = \arg\max_k \hat{p}_k$$

  - π: *encoder* — flattens the manifold to a latent code $\mathbf{z} \in \mathbb{R}^d$
  - g_ϕ: *classifier head* — 1–2 FC layers + softmax → class probabilities

- **Choices for π:**
  - **Linear:** PCA, SVD, random projections
  - **Non-linear:** Auto-encoder, t-SNE, UMAP, first layers of a CNN

- **Economist's translation:** the encoder learns a nonlinear factor model instead of imposing a fixed sufficient statistic. Sufficient statistics for prediction — by learning.

Defining a Deep Neural Network for Digit Recognition

- **Network Architecture:**
  - Input Layer: 784 neurons corresponding to pixel intensities
  - Hidden Layers: Multiple layers applying nonlinear transformations to learn complex patterns
  - Output Layer: 10 neurons, each representing one digit class (0–9)
- **Activation Functions:**
  - Nonlinear functions (ReLU, sigmoid) between layers
  - Customizable depth and width based on task complexity
- **Economic parallel:** The network structure mirrors the recursive structure in macro models — state variables are transformed layer by layer into decisions, just as $k_t \to c_t, l_t$ through policy functions.

Bridge: From Why ML to What a Network Is

Act 3.1 made the case. ML earns its place in the economist's toolkit where classical methods stall — unstructured data, flexible controls and nowcasts, high-dimensional models — and pattern recognition showed the mechanism: learn a low-dimensional representation, then decide.

- You have seen *why*: four doors ML opens for economic research, and one running illustration (digits $\rightarrow$ manifold $\rightarrow$ encoder) of how a network learns structure from data instead of having it hand-built.
- **Next (3.2):** open the box — neurons, layers, and activation functions, plus the theorems (universal approximation; Barron's dimension-free rate) that explain why these objects fit the functions economists care about.

Arc: **3.1 WHY** *(the case for ML)* $\rightarrow$ *3.2 BUILD (architecture + UAT)* $\rightarrow$ *3.3 TRAIN (backprop + SGD)* $\rightarrow$ *3.4 SHAPE (economics-aware nets + lab).*