AI and Economics: Solution Methods

Mahdi Ebrahimi Kahou[1]

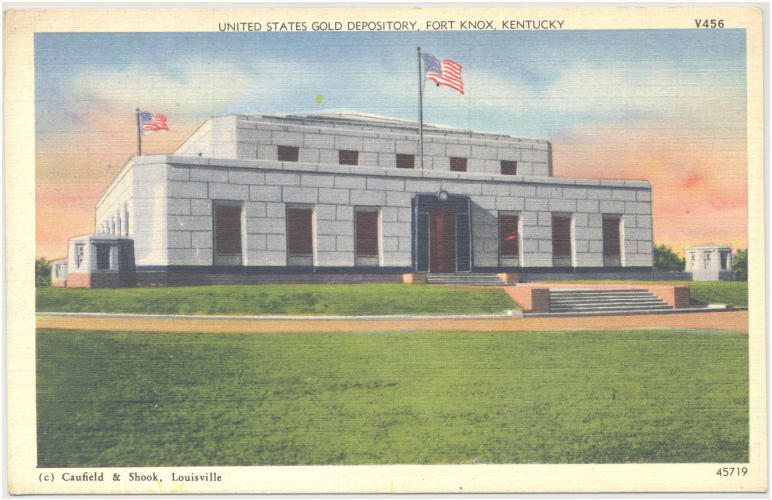
-

[1]Bowdoin College

Mathematical Models: Motivation

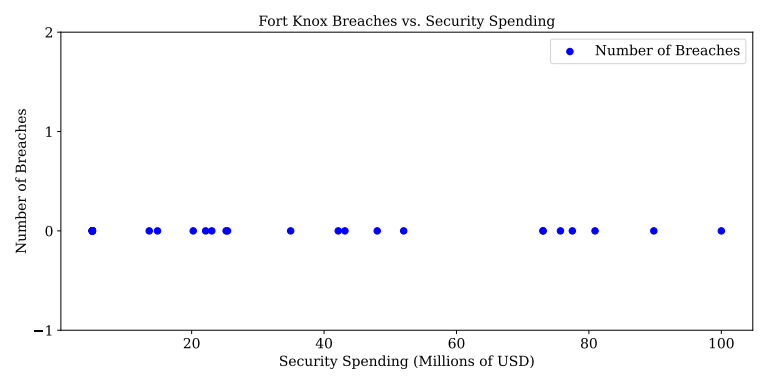
Why mathematical models?

- Where is this?



Why mathematical models?

- Half a trillion dollars' worth of gold.



Caution: The data on the x-axis is fictitious, but the y-axis data is accurate.

Why mathematical models?

- **Policy Question**:
  By how much does decreasing the security budget by 70 million dollars change the probability of a breach?
- How are you going to answer this?
- These are called **counterfactuals**.

Why mathematical models?

Scenarios similar to this:

- Predicting Economic Outcomes from Policy Changes:
  - **Scenario**: You want to understand the impact of a new tax policy on national income.
  - **Problem**: You cannot definitively assess how national income would have changed if the tax policy hadn't been implemented.
  - **Why** is it a problem? You need to know relationships between tax rates, consumer behavior, production, and other economic variables.

Why mathematical models?

Scenarios similar to this:

- Evaluating the Impact of a Minimum Wage Increase on Employment:
  - **Scenario**: You want to determine how raising the minimum wage affects employment levels.
  - **Problem**: It's impossible to predict how different sectors or regions would be affected by the wage increase.
  - **Why** is it a problem? You need to know factors labor demand elasticity, substitution effects, or the interaction between wages and other economic variables.

Why Mathematical Models?

You cannot answer these questions without a (mathematical) **model**.

- Model the incentives of the thieves, or their "payoff".
- Model the incentives of the security team, and their decision-making process.
- Model the deterrence effects of security spending.

Why mathematical models?

Many of these problems are inherently **dynamic**. Consider the Fort Knox example:

- Technology improves over time.
- Fort Knox security planners anticipate that thieves will gain access to better technology in the future.
- Thieves, in turn, expect security measures (and spending) to increase in response.

Why Mathematical Models?

Should we take our models **seriously***?*

- When experiments are not feasible, modeling is your only option.
- Even when experiments are possible, causal inference alone doesn't reveal the mechanisms behind cause and effect.

Economic Models

Economic Models

If we have to take our models seriously what happens when they **dont** *accept a* **closed-form** *solution?*

- Using numerical methods.
- Using the numerical solutions to study the counterfactuals.

Economic Models

Problems with numerical methods in economics

1. They **fail** in complex and high-dimensional problems.
2. The **dynamic** nature of the problem adds another layer of **complexity**.

Economic Models and Recent Advances in Machine Learning

Deep Residual Learning for Image Recognition

Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun
Microsoft Research
{kahe, v-xiangz, v-shren, jiansun}@microsoft.com

Abstract

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset we evaluate residual nets with a depth of up to 152 layers—8× deeper than VGG nets [41] but still having lower complexity. An ensemble of these residual nets achieves 3.57% error on the ImageNet test set. This result won the 1st place on the ILSVRC 2015 classification task. We also present analysis on CIFAR-10 with 100 and 1000 layers.

The depth of representations is of central importance for many visual recognition tasks. Solely due to our extremely deep representations, we obtain a 28% relative improvement on the COCO object detection dataset. Deep residual nets are foundations of our submissions to ILSVRC & COCO 2015 competitions, where we also won the 1st places on the tasks of ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation.

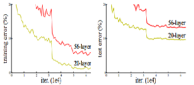


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

greatly benefited from very deep models.

Driven by the significance of depth, a question arises: *Is learning better networks as easy as stacking more layers?* An obstacle to answering this question was the notorious problem of vanishing/exploding gradients [1, 9], which hamper convergence from the beginning. This problem, however, has been largely addressed by normalized initialization [23, 9, 37, 13] and intermediate normalization layers

ARTICLE

doi:10.1038/nature16961

Mastering the game of Go with deep neural networks and tree search

David Silver[1]*, Aja Huang[1]*, Chris J. Maddison[1], Arthur Guez[1], Laurent Sifre[1], George van den Driessche[1], Julian Schrittwieser[1], Ioannis Antonoglou[1], Veda Panneershelvam[1], Marc Lanctot[1], Sander Dieleman[1], Dominik Grewe[1], John Nham[2], Nal Kalchbrenner[1], Ilya Sutskever[2], Timothy Lillicrap[1], Madeleine Leach[1], Koray Kavukcuoglu[1], Thore Graepel[1] & Demis Hassabis[1]

The game of Go has long been viewed as the most challenging of classic games for artificial intelligence owing to its enormous search space and the difficulty of evaluating board positions and moves. Here we introduce a new approach to computer Go that uses 'value networks' to evaluate board positions and 'policy networks' to select moves. These deep neural networks are trained by a novel combination of supervised learning from human expert games, and reinforcement learning from games of self-play. Without any lookahead search, the neural networks play Go at the level of state-of-the-art Monte Carlo tree search programs that simulate thousands of random games of self-play. We also introduce a new search algorithm that combines Monte Carlo simulation with value and policy networks. Using this search algorithm, our program AlphaGo achieved a 99.8% winning rate against other Go programs, and defeated the human European Go champion by 5 games to 0. This is the first time that a computer program has defeated a human professional player in the full-sized game of Go, a feat previously thought to be at least a decade away.

Economic Models and Machine Learning

Problems with numerical methods in economics

1. They **fail** in complex and high-dimensional problems.
   - Recent advances in AI, promises in solving complex and high-dimensional problem.
   - Not going to talk about this today.
2. The **dynamic** nature of the problem adds another layer of **complexity**.
   - Focus of the talk today.

Dynamic Models You Have Seen before

$$\ddot{\theta} + \omega^2 \sin(\theta) = 0$$

or

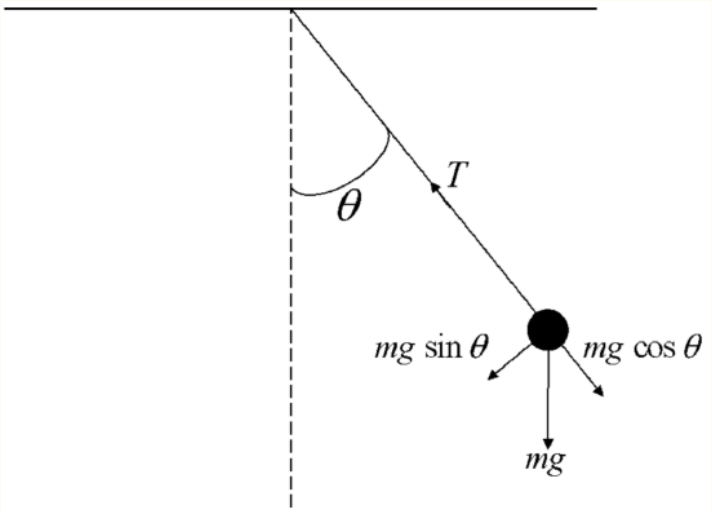
$$\dot{\theta} = \nu$$
$$\dot{\nu} = -\omega^2 \sin(\theta)$$

Dynamic Models You Have Seen before

$$\ddot{\theta} + \omega^2 \sin(\theta) = 0$$

or

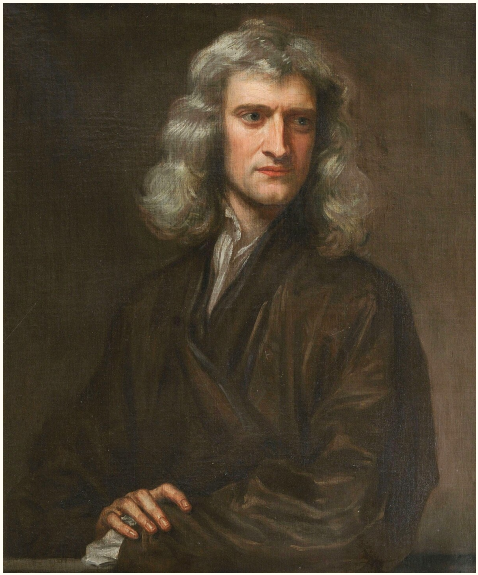
$$\dot{\theta} = \nu$$
$$\dot{\nu} = -\omega^2 \sin(\theta)$$



Dynamic Models You Have Seen before

A typical physicist to economists:

- "We solve problems like this every day."
- "What's all the fuss about?"



Dynamic Models in Economics

Dynamic models in Physics

$$x(t+1) = f\left(x(t), y(t)\right)$$
$$Y(t+1) = g\left(x(t), y(t)\right)$$
$$x(0) \text{ is given}$$
$$y(0) \text{ is given}$$

Dynamic models in Economics

$$x(t+1) = f\left(x(t), y(t)\right)$$
$$y(t+1) = g\left(x(t), y(t)\right)$$
$$x(0) \text{ is given}$$
$$\lim_{t\to\infty} h\left(x(t), y(t)\right) = 0$$

- $\lim_{t\to\infty} h\left(x(t), y(t)\right) = 0$ is a **long-run** boundary condition.

Background: Economic Models, Deep learning and inductive bias

Economic Models: functional equations

Many theoretical models can be written as functional equations:

- Economic object of interest: f, where $f : \mathcal{X} \to \mathcal{R} \subseteq \mathbb{R}^N$
  - e.g., asset price, investment choice, best-response, etc.
- Domain of f: $\mathcal{X}$
  - e.g. space of dividends, capital, opponents state or time in sequential models.
- The "Economics model" error: $\ell\big(x, f\big)$
  - e.g., Euler and Bellman residuals, equilibrium FOCs.

Then a **solution** is $f^* \in \mathcal{F}$ where $\ell(x, f^*) = \mathbf{0}$ for all $x \in \mathcal{X}$.

Approximate solution: deep neural networks

1. Sample $\mathcal{X}$: $\mathcal{D} = \{x_1, \cdots, x_N\}$
2. Pick a family of parametric functions (e.g., deep neural networks) $f_\theta(\cdot) \in \mathcal{H}(\theta)$:
   - θ: parameters for optimization (i.e., weights and biases).
3. To find an approximation for f solve:

$$\min_\theta \frac{1}{N} \sum_{x \in \mathcal{D}} \| \underbrace{\ell(x, f_\theta)}_{\text{Econ model error}} \|_2^2$$

   - Deep neural networks are highly over-parameterized: formally, $|\theta| \gg N$

Deep Neural Networks

Deep learning is **highly-overparameterized** $\mathcal{H}(\Theta)$ ($M \gg D$) class of functions.

- Example: one layer neural network, $f_\theta : \mathbb{R}^Q \to \mathbb{R}$:

$$f_\theta(x) = W_2 \cdot \sigma\left(W_1 \cdot x + b_1\right) + b_2$$

- $W_1 \in \mathbb{R}^{P \times Q}$, $b_1 \in \mathbb{R}^{P \times 1}$, $W_2 \in \mathbb{R}^{1 \times P}$, and $b_2 \in \mathbb{R}$.
- $\theta \equiv \{b_1, W_1, b_2, W_2\}$ are the coefficients, in this example $M = PQ + P + P + 1$.
- $\sigma(\cdot)$ is a nonlinear function applied element-wise (e.g., $\max\{\cdot, 0\}$).
- Making it "deeper" by adding another "layer": $f_\theta(x) \equiv W_3 \cdot \sigma(W_2 \cdot \sigma(W_1 \cdot x + b_1) + b_2) + b_3$.

Over-parameterized interpolation

- Being over-parameterized ($|\theta| \gg N$), the optimization problem can have many solutions.
- Since individual θ are irrelevant it is helpful to think of optimization directly within $\mathcal{H}$

$$\min_{f_\theta \in \mathcal{H}} \frac{1}{N} \sum_{x \in \mathcal{D}} \|\ell(x, f_\theta)\|_2^2$$

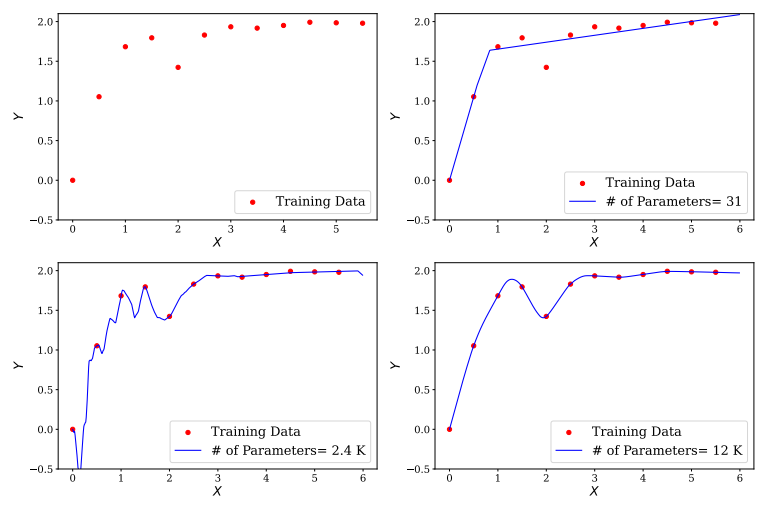
- But which f_θ?
- **Mental model:** chooses min-norm interpolating solution for a (usually) unknown functional norm ψ

$$\min_{f_\theta \in \mathcal{H}} ||f_\theta||_\psi$$
$$\text{s.t. } \ell(x, f_\theta) = 0, \qquad \text{for all } x \in \mathcal{D}$$

- That is what we mean by **inductive bias** (see Belkin, 2021 and Ma and Yang, 2021).
- Characterizing ψ (e.g., Sobolev norms or semi-norms?) is an active research area in ML.

Over-parameterization and smooth interpolation

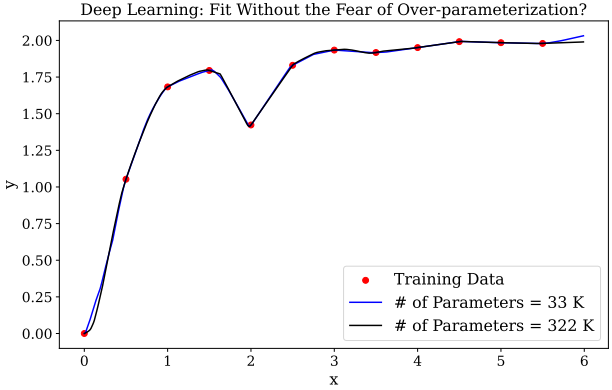
- Intuition: biased toward solutions which are flatter and have smaller derivatives



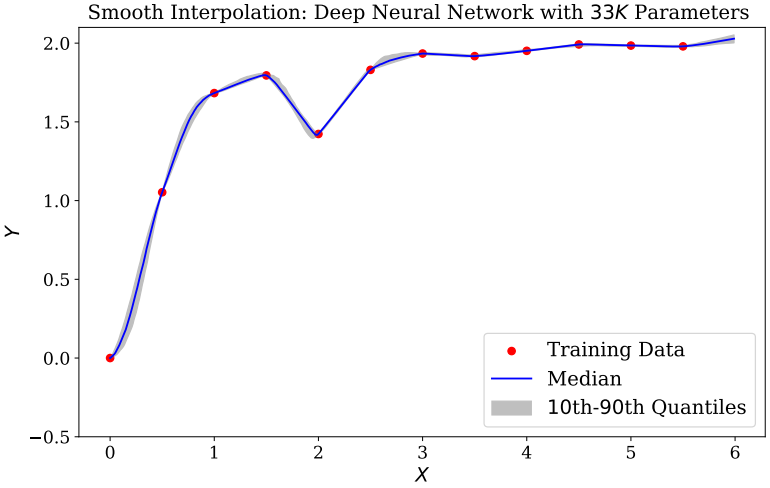
Deep Learning: "Fit Without Fear"?

- *"I remember my friend Johnny von Neumann used to say, with four parameters I can fit an elephant, and with five I can make him wiggle his trunk."* Enrico Fermi

- *"The best way to solve the problem from practical standpoint is you build a very big system ... basically you want to make sure you hit the zero training error"* Ruslan Salakhutdinov



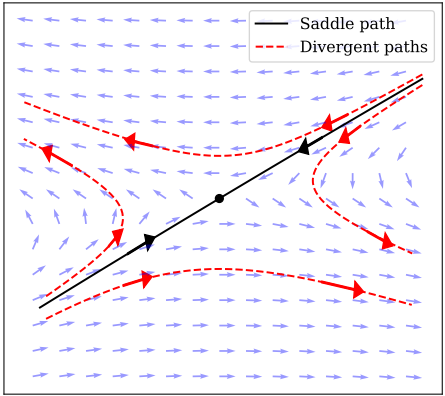
Deep Learning: random initialization and non-convex optimization



Different initialization

Intuition of the paper

- **Minimum-norm inductive bias**:
  - Over-parameterized models (e.g., large neural networks) interpolate the train data.
  - They are biased towards interpolating functions with smaller norms.
  - So they dont like explosive functions.
- **Violation of economic boundary conditions**:
  - Sub-optimal solutions diverge (explode) over time.
  - This is due to the **saddle-path** nature of econ problems.
  - The long-run boundary conditions rule out the explosive solutions.



Outline

Outline of the talk

To explore how we can ignore the long-run boundary conditions, we show deep learning solutions to

1. Classic linear-asset pricing model.
2. Sequential formulation of the neoclassical growth model.
3. Sequential formulation of the neoclassical growth model with non-concave production function.
4. Equivalent for a recursive formulation of the neoclassical growth model.

Linear asset pricing and the no-bubble condition

Linear asset pricing: setup

- The risk-neutral price, $p(t)$, of a claim to a stream of dividends, $y(t)$, is given by the recursive equation:

$$p(t) = y(t) + \beta p(t+1), \quad \text{for } t = 0, 1, \cdots$$

- $\beta < 1$, and $y(t)$ is exogenous, $y(0)$ given.
- A two dimensional dynamical system with unknown initial condition $p(0)$. This problem is **ill-posed**.
- A family of solutions

$$p(t) = \underbrace{p_f(t)}_{\text{fundamentals}} + \underbrace{\zeta \left(\frac{1}{\beta}\right)^t}_{\text{explosive bubble}}$$

- $p_f(t) \equiv \sum_{\tau=0}^{\infty} \beta^\tau y(t+\tau)$. Each solution corresponds to a different $\zeta > 0$.

Linear asset pricing: the long-run boundary condition

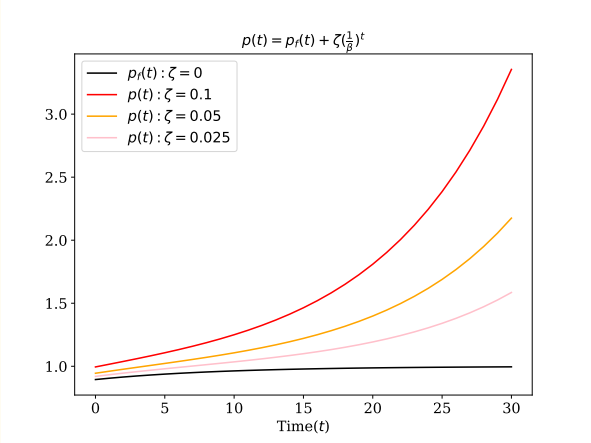
- Long-run boundary condition that rule out the explosive bubbles and chooses $\zeta = 0$

$$\lim_{t \to \infty} \beta^t p(t) = 0.$$

- Any norm that preserve monotonicity, like L_p and Sobolev (semi-)norms

$$\min_{\zeta \geq 0} \|p\|_\psi = \|p_f\|_\psi$$

- Ignoring the no-bubble condition and using a deep neural network provides an accurate approximation for $p_f(t)$.



Linear asset pricing: numerical method

- Sample for time: $\mathcal{D} = \{t_1, \cdots, t_N\}$.
- Generating the dividend process: $y(t+1) = c + (1+g)y(t)$, given $y(0)$.
- An over-parameterized neural network $p_\theta(t)$, **ignore** the non-bubble condition and solve

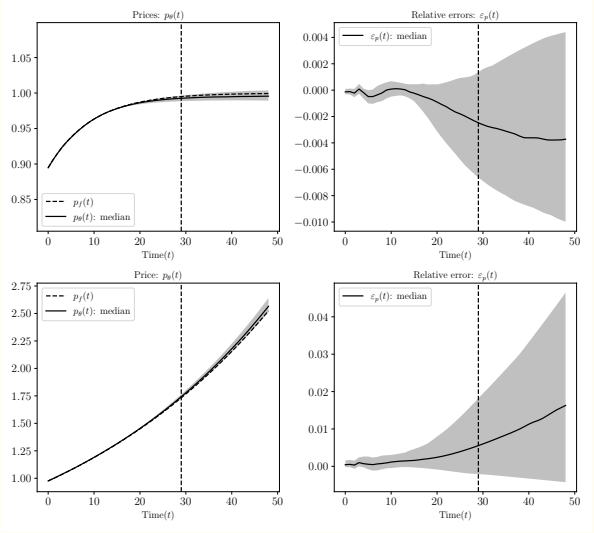
$$\min_\theta \frac{1}{N} \sum_{t \in \mathcal{D}} [p_\theta(t) - y(t) - \beta p_\theta(t+1)]^2$$

- This minimization should provide an accurate short- and medium-run approximation for price based on the fundamentals $p_f(t)$.

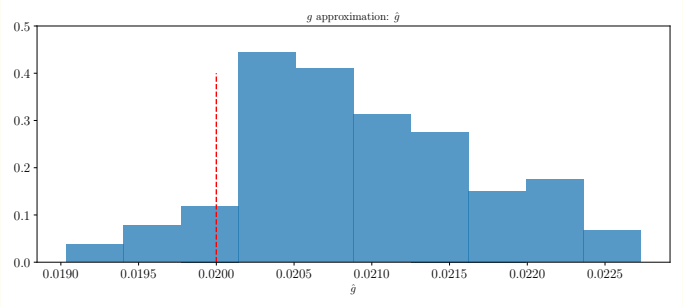
Linear asset pricing: results

- Two cases: $g < 0$ and $g > 0$.
- Relative errors: $\varepsilon_p(t) \equiv \frac{p_\theta(t) - p_f(t)}{p_f(t)}$.
- for $g > 0$: $p_\theta(t) = e^{\phi t} NN_\theta(t)$, ϕ is "learnable".

- Results for 100 different seeds (initialization of the parameters):
  - important for **non-convex** optimizations.
- Very accurate short- and medium-run approximation.



Learning the growth rate



- $\hat{g} = e^{\phi} - 1$.
- Slightly biased due to small sample size, i.e., $\mathcal{D} = \{0, 1, \cdots, 29\}$.

Sequential neoclassical growth model and the transversality condition

Neoclassical growth model: setup

- Total factor productivity $z(t)$ exogenously given, capital $k(t)$ with given $k(0)$, consumption $c(t)$, production function $f(\cdot)$, depreciation rate $\delta < 1$, discount factor β :

$$\underbrace{k(t+1) = z(t)^{1-\alpha} f\left(k(t)\right) + (1-\delta)k(t) - c(t)}_{\text{feasibility constraint}},$$

$$\underbrace{c(t+1) = \beta c(t)\left[z(t+1)^{1-\alpha} f'\left(k(t+1)\right) + 1 - \delta\right]}_{\text{Euler equation}}.$$

- A three dimensional dynamical system with unknown initial condition $c(0)$. This problem is **ill-posed**.

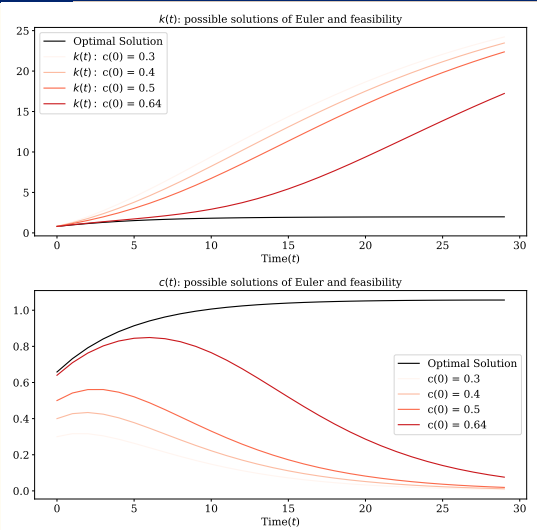
- A family of solutions, each solution corresponds to a different $c(0)$. Only one of them is the optimal solution.

Neoclassical growth model: the long-run boundary condition

- To rule out sub-optimal solutions, transversality condition

$$\lim_{t\to\infty} \beta^t \frac{k(t+1)}{c(t)} = 0.$$

- Using a deep neural network and ignoring the transversality condition provides a an accurate approximation for the optimal capital path.



Neoclassical growth model: numerical method

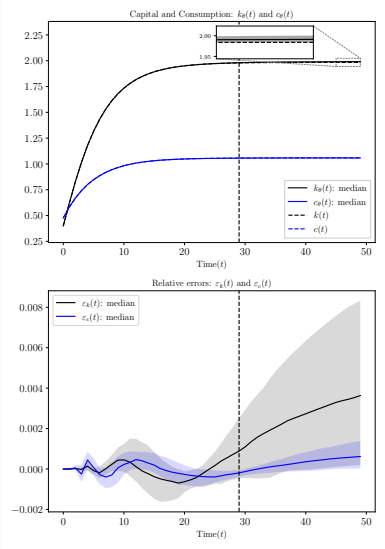
- Sample for time: $\mathcal{D} = \{t_1, \cdots, t_N\}$.
- TFP process: $z(t+1) = (1+g)z(t)$, given $z(0)$.
- A over-parameterized neural network $k_\theta(t)$,
- Given $k_\theta(t)$, define the consumption function $c(t; k_\theta) = z(t)^{1-\alpha} f(k_\theta(t)) + (1-\delta)k_\theta(t) - k_\theta(t+1)$
- **Ignore** the transversality condition and solve

$$\min_{\theta \in \Theta} \left[\frac{1}{N} \sum_{t \in \mathcal{D}} \left(\underbrace{\frac{c(t+1; k_\theta)}{c(t; k_\theta)} - \beta \left[z(t+1)^{1-\alpha} f'\big(k_\theta(t+1)\big) + (1-\delta) \right]}_{\text{Euler residuals}} \right)^2 + \left(\underbrace{k_\theta(0) - k_0}_{\text{Initial condition residual}} \right)^2 \right]$$

- This minimization should provide an accurate short- and medium-run approximation for the optimal capital and consumption path.

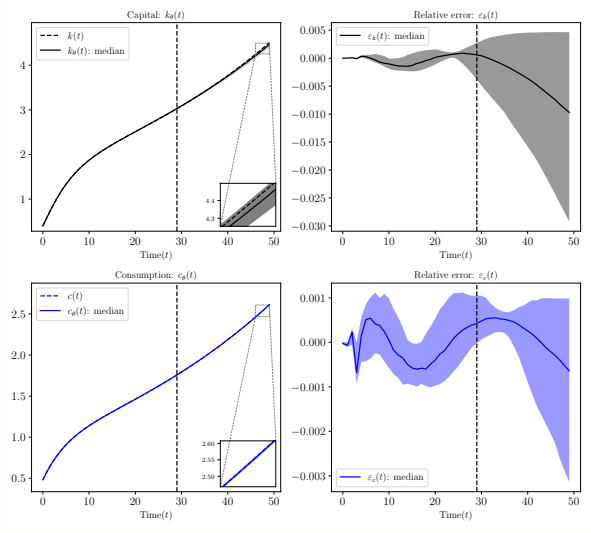
Neoclassical growth model, no TFP growth: results

- $g = 0$, $z(0) = 1$.
- $\varepsilon_k(t) \equiv \frac{k_\theta(t) - k(t)}{k(t)}$, and $\varepsilon_c(t) \equiv \frac{c(t; k_\theta) - c(t)}{c(t)}$
- Benchmark solution: value function iteration.
- Results for 100 different seeds.
- Very accurate short- and medium-run approximation.



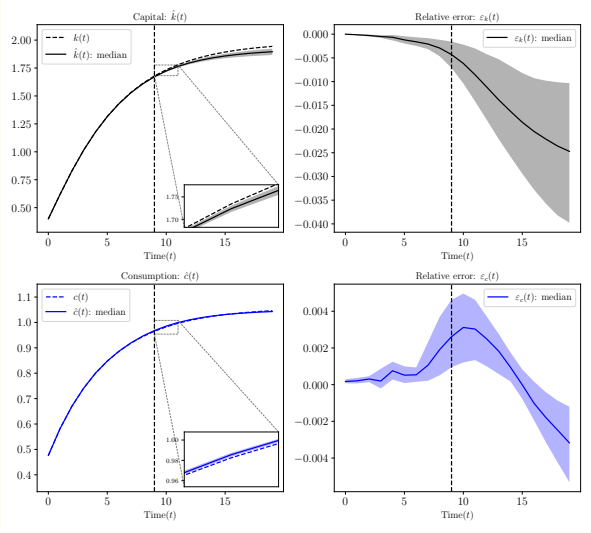
Neoclassical growth model with TFP growth: results

- $g > 0$ and $z(0) = 1$.
- $k_\theta(t) = e^{\phi t} NN_\theta(t)$, ϕ is "learnable".
- Results for 100 different seeds.
- Very accurate short- and medium-run approximation.



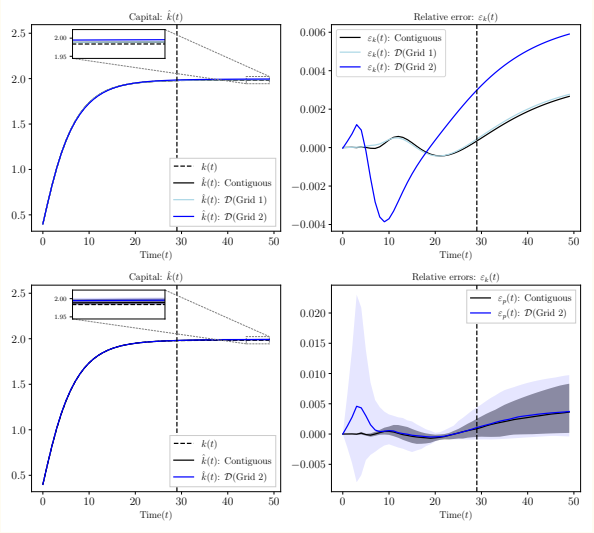
But, seriously "in the long run, we are all dead"

- So far, we have used long time-horizon $\mathcal{D} = \{0, 1, \cdots, 29\}$.
- In other methods, choosing the time-horizon T is a challenge:
  - Too large $\rightarrow$ accumulation of errors, and numerical instability. We don't have that problem.
  - Too small $\rightarrow$ convergence to the steady state too quickly.
- An accurate short-run solution, even for a medium-sized T.



Do we need a dense and contiguous grid?

- We have used a dense $\mathcal{D} = \{0, 1, \cdots, 29\}$.
- What if
  - $\mathcal{D}(\text{Grid 1}) = \{0, 1, 2, 4, 6, 8, 12, 16, 20, 24, 29\}$
  - $\mathcal{D}(\text{Grid 2}) = \{0, 1, 4, 8, 12, 18, 24, 29\}$
- An accurate short-run solution, even for a sparse grid.

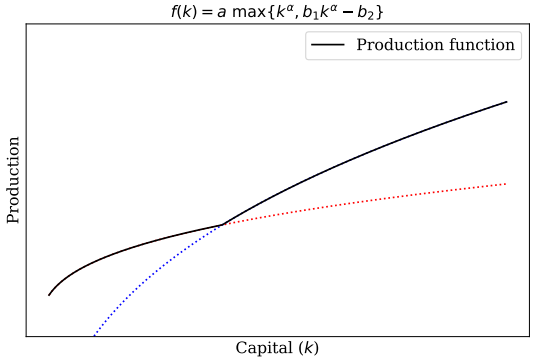


Neoclassical growth model: multiple steady-states and hysteresis

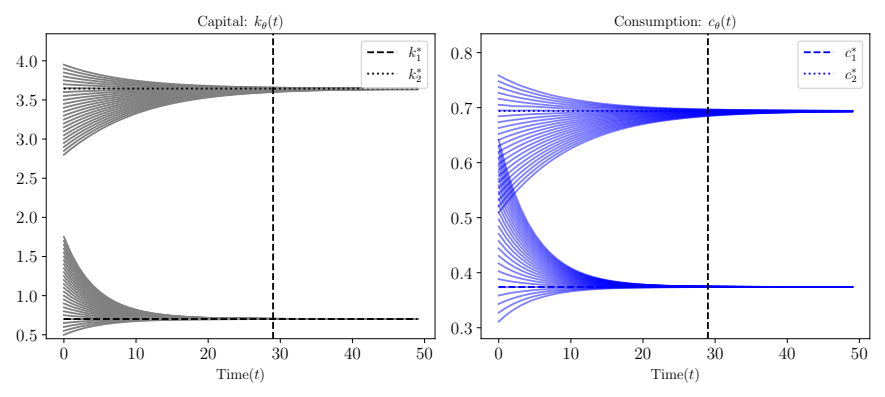
- When there are multiple steady states with saddle-path stability, each with its domain of attraction:
  - Can the inductive bias detect there are multiple basins of attraction?
  - How does the inductive bias move us toward the correct steady state for a given initial condition?
- Consider a non-concave production function:

$$f(k) \equiv a \max\{k^{\alpha}, b_1 k^{\alpha} - b_2\}$$

- Two steady-states k_1^* and k_2^*.
- The same numerical procedure.



Neoclassical growth model with non-concave production function: results



- Different initial conditions in $k_0 \in [0.5, 1.75] \cup [2.75, 4]$.
- In the vicinity of k_1^* and k_2^* the paths converge to the right steady-states.

Deep learning is not the only option

Deep learning is not the only option: kernels

- Deep learning might be too "spooky".
- We can use kernels methods, $K(\cdot, \cdot)$, instead of neural networks and control the RKHS norms.
- Focusing on continuous time equivalent of these problems.
- The same results, theoretical guarantees, very fast and robust.
- With J Perla, R Childers, and G Pleiss.

How Inductive Bias in Kernel Methods Aligns with Optimality in Economic Dynamics

SUBMISSION 764

This paper examines the alignment of inductive biases in machine learning (ML), such as kernel machines, with structural models of economic dynamics. Unlike dynamical systems found in physical and life sciences, economic models are often specified by differential equations with a mixture of easy-to-enforce initial conditions and hard-to-enforce infinite horizon boundary conditions (e.g. transversality and no-ponzi-scheme conditions). We investigate algorithms using ridgeless kernel methods trained to fulfill the differential equations without explicitly fulfilling the boundary conditions. Our findings provide theoretical guarantees for cases where the inductive biases of these ML models are sufficient conditions to fulfill the infinite-horizon conditions. We then provide empirical evidence that ridgeless kernel methods are not only theoretically sound with respect to economic assumptions, but may even dominate classic algorithms in low to medium dimensions.

Contents

Optimal control framework

Consider the following problem arising in optimal control:

$$\begin{aligned}
\dot{\boldsymbol{x}} &= \boldsymbol{F}\left(\boldsymbol{x}(t), \boldsymbol{y}(t)\right) \\
\dot{\boldsymbol{\mu}} &= r\boldsymbol{\mu}(t) - \boldsymbol{\mu}(t) \odot \boldsymbol{G}\left(\boldsymbol{x}(t), \boldsymbol{\mu}(t), \boldsymbol{y}(t)\right) \\
\mathbf{0} &= \boldsymbol{H}\left(\boldsymbol{x}(t), \boldsymbol{\mu}(t), \boldsymbol{y}(t)\right) \\
\boldsymbol{x}(0) &= \boldsymbol{x}_0
\end{aligned}$$

- State variables $\boldsymbol{x}(t) \in \mathbb{R}^M$, initial condition $\boldsymbol{x}_0$; co-state variables $\boldsymbol{\mu}(t) \in \mathbb{R}^M$; jump variables $\boldsymbol{y}(t) \in \mathbb{R}^P$
- This problem is **ill-posed** and can have infinitely many solutions.

Transversality condition: an asymptotic boundary condition

$$\lim_{t \to \infty} e^{-rt} \boldsymbol{x}(t) \odot \boldsymbol{\mu}(t) = \mathbf{0}$$

- The transversality condition is an asymptotic boundary condition.
- We typically assume a finite time horizon T and shoot for the finite steady state $\boldsymbol{x}^*$, $\boldsymbol{\mu}^*$, and $\boldsymbol{y}^*$.
- This approach is straightforward in low dimensions but becomes significantly more challenging in high-dimensional settings.

Optimal control framework: an example, Ramsey–Cass–Koopmans model

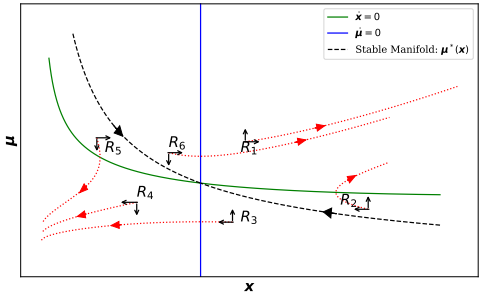
- Classic Ramsey–Cass–Koopmans

$$
\begin{aligned}
\dot{k}(t) &= f\left(k(t)\right) - c(t) - \delta k(t) \\
\dot{\mu}(t) &= r\mu(t) - \mu(t)\left[f'(k(t)) - \delta\right] \\
0 &= c(t)\mu(t) - 1 \\
k(0) &= k_0 \\
0 &= \lim_{t\to\infty} e^{-rt} k(t)\mu(t)
\end{aligned}
$$

- $f(\cdot)$ is the production function, r discount rate, and δ is the depreciation.

What does the violation of the transversality condition look like?

- All paths solve the ordinarily differential equations ans the algebraic equation.
- The solutions that violate the transversality condition $\lim_{t\to\infty} \dot{\mu} = \infty$ and $\lim_{t\to\infty} \mu = \infty$
  - Diverges faster than e^{rt}.



Kernel approximation

Approximating the derivatives with a kernel:

$$\hat{\boldsymbol{x}}(t) = \boldsymbol{x}_0 + \int_0^t \hat{\dot{\boldsymbol{x}}}(\tau)d\tau, \qquad \hat{\boldsymbol{\mu}}(t) = \hat{\boldsymbol{\mu}}_0 + \int_0^t \hat{\dot{\boldsymbol{\mu}}}(\tau)d\tau, \qquad \hat{\boldsymbol{y}}(t) = \hat{\boldsymbol{y}}_0 + \int_0^t \hat{\dot{\boldsymbol{y}}}(\tau)d\tau,$$

$$\hat{\dot{\boldsymbol{x}}}(t) = \sum_{j=1}^{N} \boldsymbol{\alpha}_j^x K(t, t_j), \qquad \hat{\dot{\boldsymbol{\mu}}}(t) = \sum_{j=1}^{N} \boldsymbol{\alpha}_j^\mu K(t, t_j), \qquad \hat{\dot{\boldsymbol{y}}}(t) = \sum_{j=1}^{N} \boldsymbol{\alpha}_j^y K(t, t_j)$$

- $\boldsymbol{x}_0$ is given.
- $\hat{\boldsymbol{\mu}}_0$, $\hat{\boldsymbol{y}}_0$, $\boldsymbol{\alpha}^x$, $\boldsymbol{\alpha}^\mu$, and $\boldsymbol{\alpha}^y$ are learnable parameters.
- $K(\cdot, \cdot)$ is the kernel.

Approximate solution: Algorithm

$$\min_{\substack{\hat{\boldsymbol{x}}(t)\in\mathcal{H}^M,\hat{\boldsymbol{\mu}}(t)\in\mathcal{H}^M,\\ \hat{\boldsymbol{y}}(t)\in\mathcal{H}^P}} \left(\sum_{m=1}^{M} \|\hat{\dot{X}}^{(m)}\|_{\mathcal{H}}^2 + \sum_{m=1}^{M} \|\hat{\dot{\mu}}^{(m)}\|_{\mathcal{H}}^2\right)$$

$$\text{s.t. } \hat{\dot{\boldsymbol{x}}} = \boldsymbol{F}\left(\hat{\boldsymbol{x}}(t), \hat{\boldsymbol{y}}(t)\right)$$

$$\hat{\dot{\boldsymbol{\mu}}} = r\hat{\boldsymbol{\mu}}(t) - \hat{\boldsymbol{\mu}}(t) \odot \boldsymbol{G}\left(\hat{\boldsymbol{x}}(t), \hat{\boldsymbol{\mu}}(t), \hat{\boldsymbol{y}}(t)\right)$$

$$\boldsymbol{0} = \boldsymbol{H}\left(\hat{\boldsymbol{x}}(t), \hat{\boldsymbol{\mu}}(t), \hat{\boldsymbol{y}}(t)\right)$$

- The objective function penalizes explosive paths.
- Constraints solve the "first order conditions".

Application: Growth with human and physical capital, a mid-size problem

$$\dot{k}(t) = i_k(t) - \delta_k k(t), \qquad \dot{h}(t) = i_h(t) - \delta_h h(t),$$
$$\dot{\mu}_k(t) = r\mu_k(t) - \mu_k(t)\left[f_k\left(k(t), h(t)\right) - \delta_k\right], \qquad \dot{\mu}_h(t) = r\mu_h(t) - \mu_h(t)\left[f_h\left(k(t), h(t)\right) - \delta_h\right]$$
$$0 = \mu_k(t)c(t) - 1, \qquad 0 = \mu_k(t) - \mu_h(t)$$
$$0 = f\left(k(t), h(t)\right) - c(t) - i_k(t) - i_h(t),$$

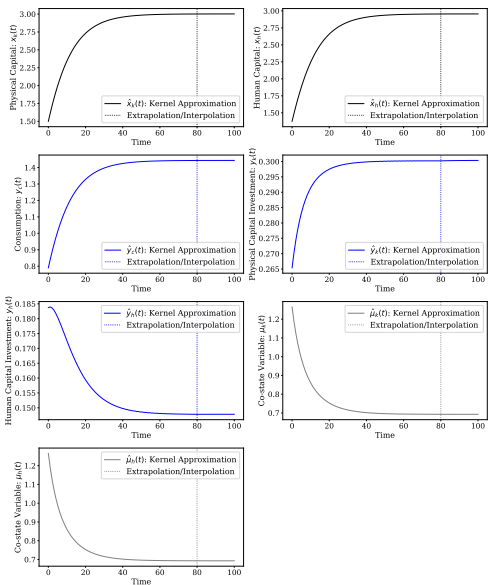
for given initial conditions $k(0) = k_0$, $h(0) = h_0$, and two transversality conditions

$$0 = \lim_{t\to\infty} e^{-rt} k(t)\mu_k(t), \qquad 0 = \lim_{t\to\infty} e^{-rt} h(t)\mu_h(t).$$

- $\boldsymbol{x}(t) = [k(t), h(t)]^T$, $\boldsymbol{\mu}(t) = [\mu_k(t), \mu_h(t)]^T$, $\boldsymbol{y}(t) = [i_k(t), i_h(t), c(t)]^T$

Results

- Accurate short- and medium-run solution.
- The solution “learns” the steady state.



Back to deep learning: Recursive neoclassical growth model and the transversality condition

Recursive formulation (with a possible BGP)

Skipping the Bellman formulation and going to the first order conditions in the state space , i.e., (k, z)

$$
\begin{aligned}
& u'(c(k,z)) = \beta u'(c(k'(k,z),z'))\left[z'^{1-\alpha} f'(k'(k,z)) + 1 - \delta\right] \\
& k'(k,z) = z^{1-\alpha} f(k) + (1-\delta)k - c(k,z) \\
& z' = (1+g)z \\
& k' \geq 0 \\
& 0 = \lim_{T\to\infty} \beta^T u'(c_T) k_{T+1} \quad \forall (k_0, z_0) \in \mathcal{X}
\end{aligned}
$$

- Preferences: $u(c) = \frac{c^{1-\sigma}-1}{1-\sigma}$, $\sigma > 0$, $\lim_{c\to 0} u'(c) = \infty$, and $\beta \in (0,1)$.
- Cobb-Douglas production function: $f(k) = k^{\alpha}$, $\alpha \in (0,1)$ before scaling by TFP z.

Interpolation problem: the optimization problem

- A set of points $\mathcal{D} = \{k_1, \ldots, k_{N_k}\} \times \{z_1, \ldots, z_{N_z}\}$.
- A family of over-parameterized functions $k'(\cdot, \cdot; \theta) \in \mathcal{H}(\Theta)$.
- Use the feasibility condition and define $c(k, z; k') \equiv z^{1-\alpha} f(k) + (1-\delta)k - k'(k, z)$.

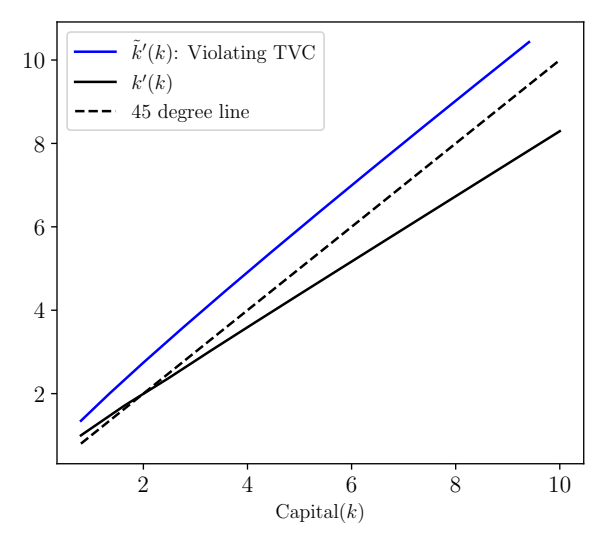
In practice we minimize the Euler residuals:

$$\min_{\theta \in \Theta} \frac{1}{|\mathcal{D}|} \sum_{(k,z) \in \mathcal{D}} \left[\underbrace{\frac{u'\Big(c\big(k, z; k'(.;\theta)\big)\Big)}{u'\Big(c\big(k'(k, z; \theta), (1+g)z; k'(.;\theta)\big)\Big)} - \beta \left[((1+g)z)^{1-\alpha} f'\left(k'(k, z; \theta)\right) + 1 - \delta \right]}_{\text{Euler residual}} \right]^2$$

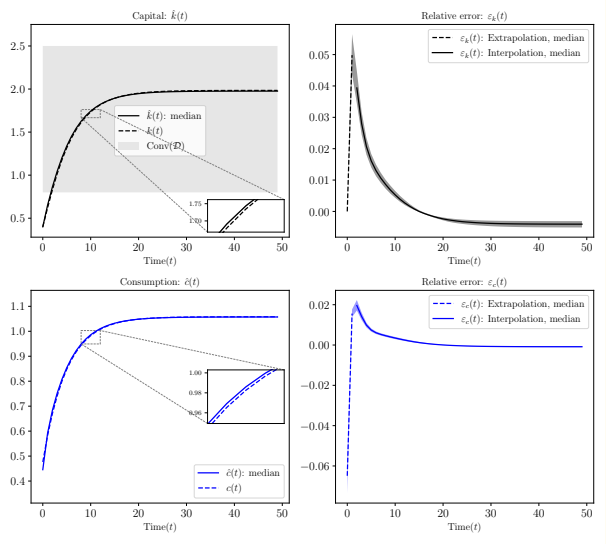
Interpolation problem: without the transversality condition

- This minimization **does not contain** the transversality condition.
  - Without the transversality condition it has more than one minima.
- **No explicit** norm regularization.
- Does the implicit bias weed out the solutions that violate the transversality condition? Let's analyze this more rigorously.

Is the transversality condition necessary? Case of g = 0, z = 1



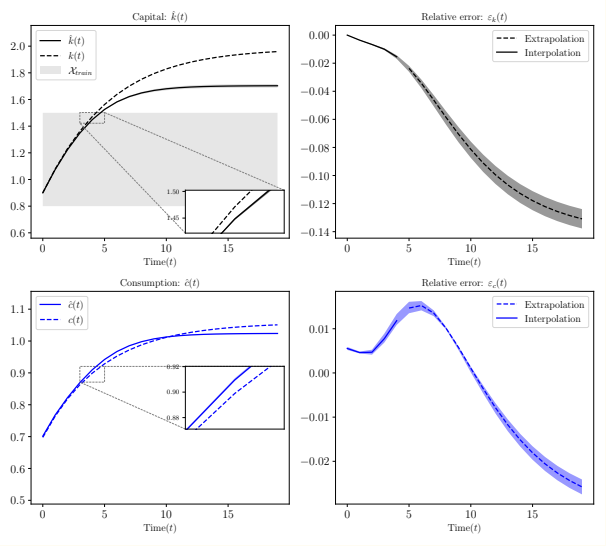
Results: one initial condition



- Picking $\mathcal{D} = [0.8, 2.5] \times \{1\}$ and $k_0 = 0.4 \notin \mathcal{D}$ is "extrapolation" $\alpha = \frac{1}{3}$, $\sigma = 1$, $\beta = 0.9$, and $g = 0$. .
- Low generalization errors, even without imposing transversality condition.
- Results for 100 different seeds (initialization of the parameters)

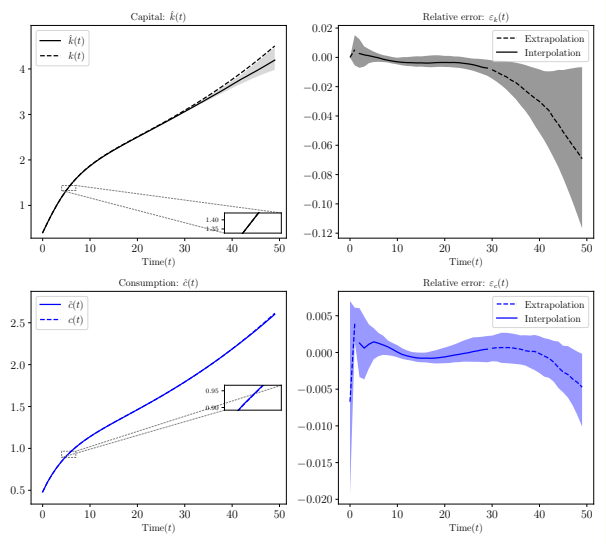
⏩ For all $k \in \mathcal{D}$

Far from the steady state



- Picking $\mathcal{D} = [0.8, 1.5]$, $k^* \notin [0.8, 1.5]$.
- A local grid around the k_0 is enough.
  - Accurate solutions in the interpolation region.
- Generalization errors are not bad.
- Results for 100 different seeds (initialization of the parameters)

Growing TFP



- Picking $\mathcal{D} = [0.8, 3.5] \times [0.8, 1.8]$ but now $g = 0.02$.
- Choosing $k'(k, z; \theta) = zNN(\frac{k}{z}, z; \theta)$.
  - Here we used economic intuition to design the $\mathcal{H}(\Theta)$.
- Relative errors are very small inside the grid.
- Small generalization errors.

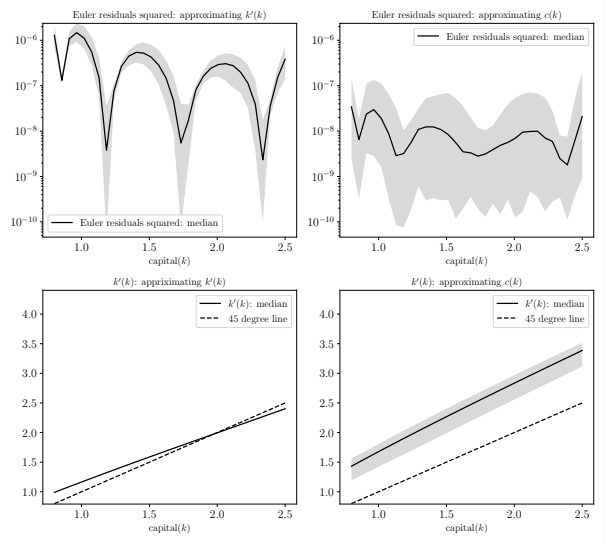
Are Euler and Bellman residuals enough?

Euler residuals are not enough

- We picked a grid $\mathcal{D}$ and approximated $k'(k)$ with an over-parameterized function.
  - The approximate solutions do not violate the transversality condition.
- What happens if we approximate the consumption functions $c(k)$ with an over-parameterized function.
  - We get an interpolating solution, i.e, very small Euler residuals.
  - However, the solutions **violate** the transversality condition.

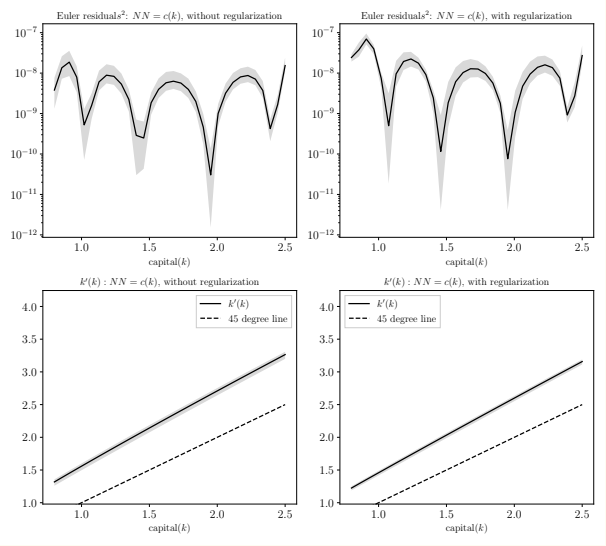
Intuition: consumption functions with low derivatives leads to optimal policies for capital with big derivatives.

Small Euler residuals can be misleading



- Left panels: approximating $k'(k)$ with a deep neural network.
  - The solutions do not violate the TVC.
  - $k'(k)$ intersects with 45° line at $k^* \approx 2$.
- Right panels: approximating $c(k)$ with a deep neural network.
  - The solutions **violate** the TVC.
  - $k'(k)$ intersects with 45° line at $\tilde{k}_{\max} \approx 30$.
  - Euler residuals are systematically lower.

Can regularization fix this problem?



- Left panels: approximating $c(k)$ with a deep neural network without explicit regularization.
- What does happen with L_2 regularization?
  - Penalizing $\sum_{\theta_i \in \Theta} \theta_i^2$.
- Right panels: approximating $c(k)$ with a deep neural network **with** explicit regularization.
- Using deep learning requires understanding the **inductive bias** and **economic theory**.

Conclusion

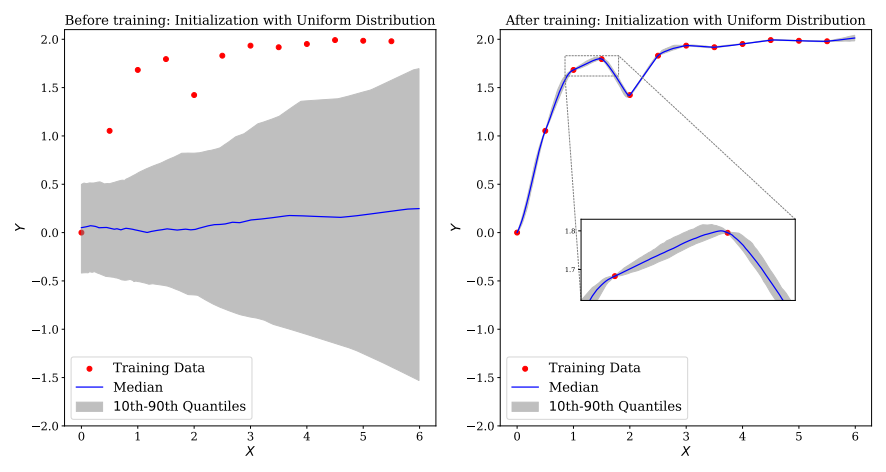
- Short- and medium-run accurate solutions can be obtained **without** strictly enforcing the long-run boundary conditions on the model's dynamics.
- Long-run (**global**) conditions can be replaced with appropriate regularization (**local**) to achieve optimal solutions, hence the title of the paper.
- Inductive bias provides a foundation for modeling forward-looking behavioral agents with self-consistent expectations.

Discussion: where to go from here?

- Can inductive bias/regularization be thought of as an equilibrium selection device?
  - In this paper it is used to select solutions.
- This method (mostly the kernel method) can be used for sampling high-dimensional state spaces when there is stochasticity.
  - Solve the deterministic in short-run and use the points as sample of the state-space.
  - Then solve the stochastic problem.

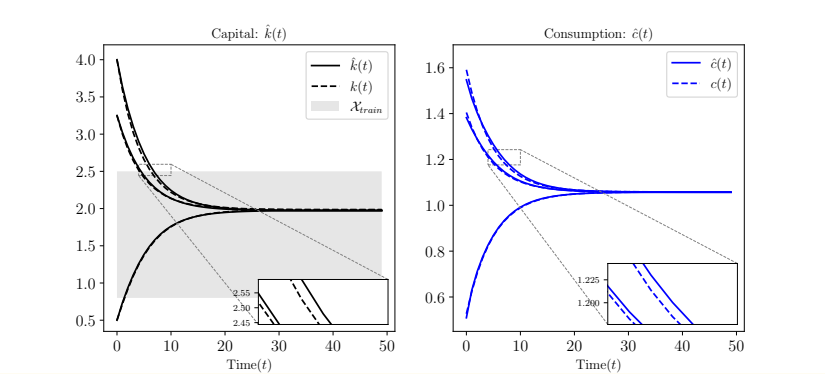
Appendix

Deep Learning: random initialization and non-convex optimization



Back

Results: initial conditions over the state space



- The solution has to satisfy the transversality condition for all points in $\mathcal{X}$
  $\lim_{T\to\infty} \beta^T u'(c(T))k(T+1) = 0 \quad \forall\ k_0 \in \mathcal{X}$
- Three different initial condition for capital, all outside of $\mathcal{X}$.

back