Experimental study of Pac-Man conditions for learn-ability of discrete linear dynamical systems

by

Zhaksybek Damiyev

Submitted to the Department of Mathematics
in partial fulfillment of the requirements for the degree of

Master of Applied Mathematics

at the

NAZARBAYEV UNIVERSITY

May 2019

Author .....
Department of Mathematics
May 1, 2019

Certified by .....
Rustem Takhanov
Assistant Professor
Thesis Supervisor

Accepted by .....
Vassilios D. Tourassis
Dean, School of Science and Technology

Experimental study of Pac-Man conditions for learn-ability of discrete linear dynamical systems

by

Zhaksybek Damiyev

Submitted to the Department of Mathematics
on May 1, 2019, in partial fulfillment of the
requirements for the degree of
Master of Applied Mathematics

Abstract

In this work, we are going to reconstruct parameters of a discrete dynamical system with a hidden layer, given by a quadruple of matrices (A, B, C, D), from system's past behaviour. First, we reproduced experimentally the well-known result of Hardt et al. that the reconstruction can be made under some conditions, called Pac-Man conditions. Then we demonstrated experimentally that the system approaches the global minimum even if an input x is a sequence of i.i.d. random variables with a non-gaussian distribution. We also formulated hypotheses beyond Pac-Man conditions that Gradient Descent solves the problem if the operator norm (or alternatively, the spectral radius) of transition matrix A is bounded by 1 and obtained the negative result, i.e. a counterexample to those conjectures.

Thesis Supervisor: Rustem Takhanov
Title: Assistant Professor

Contents

Chapter 1

INTRODUCTION

Capabilities of the artificial intelligence have been expanded with the help of machine learning. Most of the problems standing in front of machine learning are to identify model that best describes mapping of input data to output observations. Properly stated assumption is essentially helpful in solving problems related to the stock market prices prediction, face and speech recognition, text modifications and others.

One of the most important recent updates in machine learning is the introduction of recurrent neural networks (RNN) [1–4]. RNNs are a group of non-linear sequential models. The main idea is to adapt model to data, the adaptation can be formalized via a loss function. We have to minimize this loss function; thus, we will have optimization problem. Generally, this would mean we need convex function. For this case we can construct stochastic gradient descent via back propagation [5]. As the practice shows, this method produces suitable solutions even for non-convex cases. Main issue is examining theoretical foundation of the success of this phenomena. In our work we will attempt to show that the loss function will approach optimal solution even for non-convex case.

Dropping all nonlinearities from a given neural net, we obtain it in the form of linear dynamical system. Beyond the machine learning theory, essential studies in this area are made by control theorists. The theory introduces a variety of tools to distinguish and manage linear systems.

Deep research in field of stochastic gradient descent would broaden our under-

standings of recurrent neural networks.

1.1 Importance of the subject

The concept of convexity is the basis for a lot of beautiful mathematics. When combined with computation, we will get the area of convex optimization that made a huge impact on understanding and improving world and our lives. Nevertheless, convexity does not provide all the answers. Many techniques in machine learning, statistics, deep learning, protein folding successfully solve non-convex problems that are NP-hard. Moreover, often nature or we (humans) choose techniques that are inefficient in the worst case to solve problems.

Is it possible to develop a theory to solve this problems between reality and the predictions of worst-case analysis?

Idea of the work is to analyze different natural or human-made methods, see their pathway and origin. By doing this, we will try to maximize efficient usage of computational technologies beyond scientific areas.

1.2 The concept of a discrete dynamical system, basic definitions.

A system can be referred to as "dynamic" if it changes position depending on time and it can be discovered and defined with a certain approach.

In our work, by discrete dynamic system we mean a process taking place in discrete time $\{1, \ldots, T\}$ and can be characterized by the following vectors (the index corresponds to a moment in time):

- $x_1, \ldots, x_T$-input variables,
- $h_1, \ldots, h_T$-hidden variables,
- $\xi_1, \ldots, \xi_T$-Gaussian noise with standard normal distribution and the following

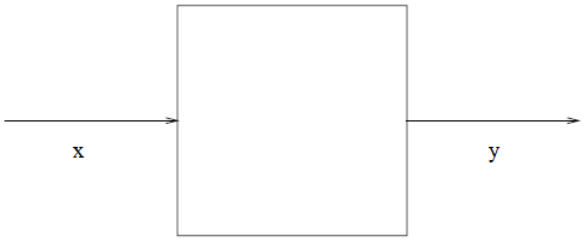
equations [6]:

$$h_{t+1} = \sigma_1(Ah_t + Bx_t)$$
$$x_{t+1} = \sigma_2(Ch_t + Dx_t) + \xi_t$$

where A, B, C, D are transition matrices and σ_1, σ_2-activation functions. The latter equation is written to give the basic notion of dynamical system, but in our work we will consider linear activation functions.

1.3 Input-Output Systems

Dynamical systems are defined by group of variables and its relationship over time. In this work we consider only discrete time. The following scheme illustrates the simplest input-output system:



where input $x_t = (x_t^1, x_t^2, \ldots, x_t^m)^T$ is the vector with m input variables at t and $y_t = (y_t^1, y_t^2, \ldots, y_t^p)^T$ is the vector with p components. The box in the center is called a plant and stands for the way the outputs depend on the inputs. If $m = p = l$ system is called Single Input Single Output (SISO).

Example 1.3.1 *We consider business cycle model which describes the relation between national income y_t and government spending g_t,*

$$y_t - \gamma(1+\alpha)y_{t-1} + \gamma\alpha y_{t-2} = g_t. \tag{1.1}$$

We denote our dynamical system by the symbol Σ for convenience and assume that system Σ starts at time $t = 0$ and can be defined by (A, B, C, D). The values of

variables are zeros when $t < 0$. Since we consider only discrete time, input sequence is $x = (x_0, x_1, \ldots)$, where each component is a vector in $\mathbb{R}^m$. The output sequence is $y = (y_0, y_1, \ldots)$ with elements from $\mathbb{R}^p$. Let us denote *input-output map* from input x onto y by G_Σ. Then output y_t is defined by $y_t = G_\Sigma x_t$.

The system Σ is called *causal* if y does not depend on future inputs and can be obtained by past and present input sequences.

The system is called *linear* if the following two conditions hold:

$$G_\Sigma(x + z) = G_\Sigma x + G_\Sigma z, \quad G_\Sigma(\lambda x) = \lambda G_\Sigma x \quad (\lambda \in \mathbb{R}),$$

where z is another input sequence. If the system is causal and linear, then

$$y_t = G_\Sigma x_t = \sum_{i=0}^{t} G(t, i)x_i, \qquad t = 0, 1, 2, \ldots$$

where $G(t, i)$ are $p \times m$ matrices.

The system is called a *time-invariant* if input-output mapping does not depend on t. To make it more understandable, let us consider the shift operator S, defined by

$$S(x_0, x_1, x_2, \ldots) = S(0, x_0, x_1, \ldots). \tag{1.2}$$

The system is time-invariant if the following equation holds:

$$G_\Sigma S = S G_\Sigma. \tag{1.3}$$

Proposition 1.3.1 *Let the system be a causal time-invariant. Then the input-output map of Σ has the following form:*

$$G_\Sigma x_t = \sum_{j=0}^{t} G(t - j)x_j, \qquad t \geq 0. \tag{1.4}$$

Example 1.3.2 *Let's consider again the relation* (1.1) *between spending g_t and income y_t. If we take spending g as input and income y as output, it can be shown that*

it is a causal, linear and time-invariant system with condition $y_t = g_t = 0$ *for* $t < 0$. *From* (1.1) *we can show that the impulse response of this system defined by* $G_0 = 1$, $G_1 = (1 + \alpha)$, *for* $k \geq 2$ *the impulse response satisfies recursive equation*

$$G_k = \gamma(1 + \alpha)G_{k-1} + \gamma\alpha G_{k-2}, \qquad k \geq 2.$$

1.4 State Space Models

The input-output map T has a *state space representation* if $Tx = y$ can be given by the following system of linear equations:

$$\begin{aligned} h_{t+1} &= Ah_t + Bx_t, \quad t = 0, 1, 2, \ldots \\ y_t &= Ch_t + Dx_t, \\ h_0 &= 0. \end{aligned} \tag{1.5}$$

As in previous section, it is assumed that if $t < 0$, then $x_t = 0$, $h_t = 0$ and $y_t = 0$. A is a linear transformation $A : \mathbb{R}^n \to \mathbb{R}^n$ called *state space*, $B : \mathbb{R}^m \to \mathbb{R}^n$, $C : \mathbb{R}^n \to \mathbb{R}^p$ and $D : \mathbb{R}^m \to \mathbb{R}^p$ are linear transformations. Let us take real numbers as a components of matrices A, B, C and D. Then A is called the *state transition matrix*, B is the *input matrix*, C is the *output matrix* and D is called the *external matrix*.

Example 1.4.3 *Consider the equation* (1.1) *of Example 1.3.1 for the relation between* y_t *and* g_t. *Assume that* y_{t_0-1} *and* y_{t_0-2} *are known, then for* $t \geq t_0$ y_t *is uniquely determined by* g_t *for* $t \geq t_0$. *Then the following vector*

$$h_t = \begin{pmatrix} y_{t-1} \\ y_{t-2} \end{pmatrix}$$

defines the "state" of the economy at year t *and state space equation of* (1.1) *is*

described by

$$\begin{cases} h_{t+1} = \begin{pmatrix} \gamma(1+\alpha) & -\gamma\alpha \\ 1 & 0 \end{pmatrix} h_t + \begin{pmatrix} 1 \\ 0 \end{pmatrix} g_t, \quad t \geq t_0, \\ y_t = \begin{pmatrix} \gamma(1+\alpha) & -\gamma\alpha \end{pmatrix} h_t + g_t. \end{cases} \tag{1.6}$$

Suppose that in (1.5) h_{t_0} is known, then for $t \geq h_{t_0} \geq 0$ h_t can be described as the following:

$$h_t = A^{t-t_0} h_{t_0} + \sum_{i=t_0}^{t-1} A^{t-i-1} B x_i, \qquad t > t_0. \tag{1.7}$$

Then, the output y_t of the system is defined by

$$y_t = Dx_t + CA^{t-t_0} h_{t_0} + \sum_{i=t_0}^{t-1} CA^{t-i-1} B x_i. \tag{1.8}$$

1.5 Equivalent and Minimal Realizations

The parameters (A, B, C, D) are called a *realization* of Σ if

$$\begin{cases} h_{t+1} = Ah_t + Bx_t, \quad t = 0, 1, \ldots, \\ y_t = Ch_t + Dx_t \end{cases} \tag{1.9}$$

(1.9) is a state representation of Σ which maps input sequences to output.

Realization is not unique. For example, if we replace a state variable h_t in model by $\tilde{h}_t = Th_t$, where T is invertible matrix. Then we have

$$\tilde{h}_{t+1} = Th_{t+1} = TAh_t + TBx_t = TAT^{-1}\tilde{h}_t + TBx_t,$$

and $y_t = Ch_t + Dx_t = CT^{-1}\tilde{h}_t + Dx_t$. It means that if (A, B, C, D) is a realization of Σ, then $(TAT^{-1}, TB, CT^{-1}, D)$ is also a realization of Σ.

Suppose that (A_0, B_0, C_0, D_0) is a realization, and

$$A = \begin{pmatrix} A_1 & A_3 & A_4 \\ 0 & A_0 & A_5 \\ 0 & 0 & A_2 \end{pmatrix}, \quad B = \begin{pmatrix} B_1 \\ B_0 \\ 0 \end{pmatrix}, \quad C = \begin{pmatrix} 0 & C_0 & C_2 \end{pmatrix}, \tag{1.10}$$

where components $A_1, A_2, A_3, A_4, A_5, B_1$ and C_2 are free to choose. Then

$$A^k = \begin{pmatrix} \cdot & \cdot & \cdot \\ 0 & A_0^k & \cdot \\ 0 & 0 & \cdot \end{pmatrix},$$

where $\cdot$'s denote entries which is not important. Then for $k \geq 0$, $CA^kB = C_0A_0^kB_0$. Thus, if (A_0, B_0, C_0, D_0) is a realization of dynamical system Σ, then (A, B, C, D_0) is also a realization of Σ.

A realization (A, B, C, D) of Σ is called *minimal* if it has a smallest space dimension.

1.6 What is controllable and observable system?

A realization $\Theta = (A, B, C, D)$ of the system Σ is called *controllable* if, from any initial state h_0, any other state h_t can be obtained in finite time by choosing appropriate input sequence x. To make it more accurately, let $h_t(h_0, x)$ be the hidden layer at time t which satisfies the recursion equation of latter model $h_{t+1} = Ah_t + Bx_t$, $t = 0, 1, 2, \ldots$.

$$h_t(h_0, x) = A^t h_0 + \sum_{i=0}^{t-1} A^{t-1-i} B x_i, \quad t \geq 1. \tag{1.11}$$

Thereby Θ is controllable if and only if for $h_0, h_t \in \mathbb{R}^n$ $\exists t > 0$ and x such that $h_t = h_t(h_0, x)$.

Here we need to give a notion of reachability. A realization Θ is said to be *reachable* if from $h_0 = 0$ every other states of hidden layer can be reached by using input sequence x in finite time. Roughly, for all $h \in \mathbb{R}$ $\exists t > 0$ and an input sequence x

such that $h = h_t(h_0, x) = h_t(0, x)$. This implies that a realization which is controllable is also reachable.

Theorem 1.6.1 *Let (A, B, C, D) be a realization of Σ with space dimension n. Then,*

- *Controllability of Θ ;*
- *Reachability of Θ;*
- *$rank(B, AB, \ldots, A^{t-1}B) = n$;*
- *Non-singularity of the matrix $\sum_{i=0}^{n-1} A^i BB^T (A^T)^i$*

are equivalent.

Controllability of dynamical system is independent of the matrices C and D. Then it is enough to say that the duplex (A, B) is controllable and we ignore matrices C and D.

Example 1.6.4 *Consider the state space representation* (1.6) *of Example 1.3.1. In this example the dimension of state transition matrix is 2, and*

$$A = \begin{pmatrix} \gamma(1+\alpha) & -\gamma\alpha \\ 1 & 0 \end{pmatrix}, \qquad B = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Then

$$\begin{pmatrix} B & AB \end{pmatrix} = \begin{pmatrix} 1 & \gamma(1+\alpha) \\ 0 & 1 \end{pmatrix},$$

and its rank is 2. So this realization is controllable.

The system is said to be *observable* if the state vector can be defined from the inputs and outputs. By $y_t(h_0, x)$ we denote output at time t obtained by the input x and initial state h_0 in the system

$$\begin{cases} h_{t+1} = Ah_t + Bx_t, \quad t \geq 0 \\ y_t = Ch_t + Dx_t \end{cases} \tag{1.12}$$

Instead of second line of above state space system we can write $y_t(h_0, x) = Ch_t(h_0, x) + Dx_t$ where $h_t(h_0, x)$ is defined by (1.11). The realization Θ is *observable* if for some x the following implication holds:

$$y_t(h_0, x) = y_t(\tilde{h}_0, x), \quad t \geq 0 \implies h_0 = \tilde{h}_0. \tag{1.13}$$

It means that initial state h_0, $\tilde{h}_0$ at time $t = 0$ is uniquely determined. Then, $h_t(h_0, x) = h_t(h_0, 0) + \sum_{j=0}^{t-1} A^{t-1-j} B x_j$, and thus

$$y_t(h_0, x) = y_t(h_0, 0) + \sum_{j=0}^{t-1} CA^{t-1-j} B x_j + Dx_t,$$

so (1.13) holds if and only if

$$y_t(h_0, 0) = y_t(\tilde{h}_0, 0), \quad t \geq 0 \implies h_0 = \tilde{h}_0. \tag{1.14}$$

A state $h \in \mathbb{R}^n$ is called *unobservable* over the time $t = 0, \ldots, k-1$ if $y_t(h, 0) = CA^t h = 0$ for $t = 0, \ldots, k-1$. The set of unobservable states over the time interval $t = 0, \ldots, k-1$ is denoted by

$$\mathcal{U}_k(\Theta) = \{h \in \mathbb{R}^n | y_t(h, 0) = 0 \text{ for } t = 0, \ldots, k-1\}.$$

Then

$$\mathcal{U}_k(\Theta) = \ker \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{k-1} \end{pmatrix}. \tag{1.15}$$

Theorem 1.6.2 *Let Θ be a realization of Σ with state space dimension n. Then,*

- *Observability of Θ,*

- $rank \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{k-1} \end{pmatrix} = n,$
- *Non-singularity of the matrix* $\sum_{j=0}^{n-1} (A^T)^j C^T C A^j$

are equivalent.

There is a duality between notions controllability and observability, as (A^T, C^T) is controllable if and only if (A, C) is observable.

Example 1.6.5 *Consider the realization* (1.6) *of Example* (1.4.3), $n = 2$ *and*

$$A = \begin{pmatrix} \gamma(1+\alpha) & -\gamma\alpha \\ 1 & 0 \end{pmatrix}, \qquad C = \begin{pmatrix} \gamma(1+\alpha) & -\gamma\alpha \end{pmatrix},$$

then

$$\begin{pmatrix} C \\ CA \end{pmatrix} = \begin{pmatrix} \gamma(1+\alpha) & -\gamma\alpha \\ \gamma^2(1+\alpha)^2 - \gamma\alpha & -\gamma(1+\alpha)\gamma\alpha \end{pmatrix}.$$

Latter matrix has rank 2 if $\gamma\alpha \neq 0$. *In other words, the realization is said to be observable if* $\alpha \neq 0$ *and* $\gamma \neq 0$. *Then if we add the condition* $\alpha, \gamma > 0$ *to the model, it will be observable.*

And now we consider three-dimensional case of this system with

$$A = \begin{pmatrix} 0 & 0 & \gamma \\ -\alpha & 0 & \gamma\alpha \\ \alpha & 0 & \gamma(1+\alpha) \end{pmatrix}, \qquad C = \begin{pmatrix} -\alpha & 0 & \gamma(1+\alpha) \end{pmatrix}.$$

Then we obtain

$$\begin{pmatrix} C \\ CA \\ CA^2 \end{pmatrix} = \begin{pmatrix} \cdot & 0 & \cdot \\ \cdot & 0 & \cdot \\ \cdot & 0 & \cdot \end{pmatrix},$$

$\cdot$'s denote entries with α and γ. It is obvious that this matrix does not have rank 3, then this realization is not observable.

1.7 Structure of Realizations

Let $\Theta = (A, B, C, D)$ and $\Theta_0 = (A_0, B_0, C_0, D_0)$ be two realizations, then Θ and Θ_0 are called *similar* if

- $D = D_0$,
- Θ and Θ_0 have the same state space,
- there exists linear invertible transformation $S : \mathbb{R}^n \to \mathbb{R}^n$ and $A = SA_0S^{-1}$, $B = SB_0$, $C = C_0S^{-1}$.

The realization Θ_0 is called a *reduction* of Θ if

- $D = D_0$,
- for appropriate choice of $A_1, A_2, A_3, A_4, A_5, B_1$ and C_2

$$A = \begin{pmatrix} A_1 & A_3 & A_4 \\ 0 & A_0 & A_5 \\ 0 & 0 & A_2 \end{pmatrix}, \quad B = \begin{pmatrix} B_1 \\ B_0 \\ 0 \end{pmatrix}, \quad C = \begin{pmatrix} 0 & C_0 & C_2 \end{pmatrix}, \tag{1.16}$$

hold true.

Theorem 1.7.1 *A realization of the system Σ is minimal if and only if it is controllable and observable.*

Theorem 1.7.2

- *Two minimal realizations of the system are similar and similarity transformation is unique.*
- *Each realization of the system is similar to dilation of a minimal realization.*

1.8 Internal Stability

In general, the system is stable if perturbation does not have effect over time. It means that if system is out of equilibrium, then the dynamics brings it to original equilibrium position.

$$h_{t+1} = Ah_t \tag{1.17}$$

where A is $n \times n$ matrix with real components. Obviously, h_t is trivial solution of the system, and the aim is to determine whether the state vector h_t approaches to zero starting from $h_0 \neq 0$.

Definition 1.8.1 *The system* (1.17) *is said to be asymptotically stable if for* $t \to \infty$, $h_t \to 0$ *for any* $h_0 \in \mathbb{R}^n$.

Theorem 1.8.1 *The system* (1.17) *is called asymptotically stable if and only if all eigenvalues of matrix* A *lie in the open unit disc.*

A is *stable* if its eigenvalues are in open unit disc. Then we get the result on the stability of matrix A with respect to positive definite matrices.

Theorem 1.8.2 *The matrix* A *is stable if and only if there exists positive definite matrix* P *such that* $P - A^T PA$ *is also positive definite matrix.*

Theorem 1.8.3 *Let the duplex* (A, C) *be observable, then the matrix* A *is called stable if and only if the following equation has a unique positive definite solution*

$$P - A^T PA = C^T C, \tag{1.18}$$

where P *is defined by*

$$P = \sum_{i=0}^{\infty} \left(A^T\right)^i C^T C A^i. \tag{1.19}$$

Theorem 1.8.4 *Let the duplex* (A, B) *be controllable, then the matrix* A *is said to be stable if and only if the following equation has a unique positive definite solution*

$$Q - AQA^T = BB^T, \tag{1.20}$$

where Q is defined by

$$Q = \sum_{i=0}^{\infty} A^i B B^T \left(A^T\right)^i . \tag{1.21}$$

1.9 Input-Output Stability

In previous section we considered internal stability. Internal stability is connected with external stability, which is given by the following state space equation:

$$\begin{cases} h_{t+1} = Ah_t + Bx_t, & h_0 = 0, \\ y_t = Ch_t + Dx_t. \end{cases} \tag{1.22}$$

Definition 1.9.1 *The system 1.22 is said to be externally stable if there exist $M > 0$ and $N > 0$ such that $\|x_t\| \leq M$ implies $\|y_t\| \leq N$ where $t \geq 0$.*

Theorem 1.9.1

- *The system 1.22 is called externally stable if and only if*

$$\sum_{i=0}^{\infty} \|G_i\| < \infty,$$

where G_i is given by $G_i = CA^{i-1}B$ is the impulse response of the system 1.22 with $G_0 = D$.

- *The system is said to be externally stable if the matrix A is stable.*

Theorem 1.9.2 *Let the system 1.22 be a minimal realization, thus* (1.22) *is called externally stable if and only if the matrix A is stable.*

Chapter 2

MAIN

2.1 Problem statement

Recent works prove the efficiency of gradient descent method for minimizing MLE given noisy observations of the time-invariant linear system with $x_t \in \mathbb{R}^k$, $y_t \in \mathbb{R}^l$ and $h_t \in \mathbb{R}^n$:

$$\begin{aligned} h_{t+1} &= Ah_t + Bx_t \\ y_t &= Ch_t + Dx_t + \xi_t \qquad\qquad \xi_t \overset{iid}{\sim} N(0,1) \end{aligned} \tag{2.1}$$

where A, B, C and D are parameters which are not given directly to us. h_t is the vector of dimension n, representing order of the system, measures the hidden state at given time t. The output is disconcerted with the stochastic noise variables ξ_t. The existence of these variables makes an output error model. The systems of interest for current work are *controllable* and *externally stable* systems.

We need to take N couples of sequences (x, y) in the training set,

$$S = \Big\{(x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)})\Big\}.$$

which is sampled by (2.1) with unknown initial state h, which is allowed to be different

for every pair of sequence.

Our goal is to fit given linear system as accurately as possible to the training model constructed as the following:

$$\begin{aligned} \hat{h}_{t+1} &= \hat{A}\hat{h}_t + \hat{B}x_t \\ \hat{y}_{t+1} &= \hat{C}\hat{h}_t + \hat{D}x_t \end{aligned} \tag{2.2}$$

We use given initial states and inputs x_t and errors ξ_t, obtained results are matched with given outputs. $\hat{y}_t$ is referred to the t-th result of training model. $\hat{\Theta}$ combines all the parameters $\hat{A}$, $\hat{B}$, $\hat{C}$ and $\hat{D}$, and we get the (population) risk:

$$f(\hat{\Theta}) = \mathop{\mathbb{E}}_{\{x_t\},\{\xi_t\}} \left[\frac{1}{T} \sum_{t=1}^{T} \|\hat{y}_t - y_t\|^2 \right] \tag{2.3}$$

Even though $\hat{y}_t$ is obtained using given initial data (state, input sequence), training sequence does not converge to values of initial state.

Non-convexity obtained by squaring the loss function adds more properties. The inputs x_t and errors ξ_t are assumed to be independent of Gaussian distribution.

2.2 Methodology

When $k = l = 1$ our model is called SISO (Single-input single-output). A SISO of order n is in *controllable canonical form* if matrices A and B are in the following form:

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \dots & -a_1 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$
$$C = \begin{bmatrix} c_1 & c_2 & c_3 & \dots & c_n \end{bmatrix} \qquad D = [d] \tag{2.4}$$

Here, our goal is to parameterize $\hat{A}$, $\hat{B}$, $\hat{C}$, $\hat{D}$. We will use notation $A = CC(a)$ which is shorthand notation of latter matrix A, where a is the last unknown row $[-a_n, -a_{n-1}, \ldots, -a_1]$ of matrix. Since B is known, $\hat{B}$ is not a trainable parameter anymore, and it must be equal to B.

Any controllable system can be written in controllable canonical form [11]. For vector $a = [a_n, \ldots, a_1]$ which is the last row of matrix A, let $p_a(z)$ denote characteristic polynomial of A

$$p_a(z) = z^n + a_1 z^{n-1} + \ldots + a_n. \tag{2.5}$$

That is, $p_a(z) = \det(zI - A)$.

We know that under some weak conditions gradient descent converges even on non-convex functions to minimum [12,13]. In this work we will introduce a condition same with the quasi-convexity conception in [14], which guarantees that any point with vanishing gradient is the optimal solution. In other words, latter condition says that at point θ the negative gradient $-\nabla f(\theta)$ must be positively correlated with direction $\theta^* - \theta$ which goes to the optimal solution. Condition in our work is a little bit weaker than in [14], because we only need quasi-convexity and smoothness with respect to the optimum, and we will use it for analysis.

Definition 2.2.1 (Hardt, Tengyu Ma) *(Weak quasi-convexity). We say that the objective function f is τ-weakly-quasi-convex (τ-WQC) over a domain β with respect to global minimum if there exists a positive constant $\tau > 0$ s.t. $\forall \theta \in \beta$,*

$$\nabla f(\theta)^T(\theta - \theta^*) \geq \tau(f(\theta) - f(\theta^*)). \tag{2.6}$$

We also say that f is Γ-weakly-smooth if for any θ, we will have $\|\nabla f(\theta)\|^2 \leq \Gamma(f(\theta) - f(\theta^*))$.

Pac-Man condition: A linear dynamical system in controllable canonical form

satisfies the Pac-Man condition if the coefficient vector a defining the state transition matrix satisfies $|Re(q_a(z))| > |Im(q_a(z))|$ for all complex numbers z of modulus $|z| = 1$, where $q_a(z) = p_a(z)/z^n = 1 + a_1 z^{-1} + \ldots + a_n z^{-n}$.

It was proven in [15] that Pac-Man condition is satisfied by vectors a with $|a|_1 \leq \frac{\sqrt{2}}{2}$.

The Pac-Man condition has three important implications:

- Rouche's theorem can be used to show that the spectral radius of A is smaller than 1 and therefore ensures stability of the system.
- The vectors satisfying it form a convex set in $\mathbb{R}^n$.
- Finally, it ensures that the objective function is *quasi-convex*.

Theorem 2.2.1 *Under the Pac-Man condition, projected gradient descent algorithm, given N sample sequences of length T, returns parameters $\hat{\Theta}$ with population risk $f(\hat{\Theta}) \leq f(\Theta) + poly(n)/NT$.*

2.3 Our experimental goals

Our goal is to apply experimental mathematics to study the possibility for relaxing the conditions under which Pac-Man conditions guarantee the success of gradient descent in identification problem for discrete dynamical system. Moreover, we also hope that experimenting could help to generalize Pac-Man-conditions themselves, relaxing them even in the case $n = l = 1$ and $x_t \overset{iid}{\sim} N(0, \sigma^2)$. We make experiments and answer the following questions:

- What if we consider the cases where x_t is distributed differently from Gaussian (exponential and γ-distributions) or even x_t is not iid?
- Is it possible to find the conditions for general case, when $x_t \in \mathbb{R}^n$ and $y_t \in \mathbb{R}^l$?
- What if we consider a matrix A beyond the canonical form with operator norm $\|A\| < 1$?

- What if we consider a matrix A with spectral norm $\rho(A) = \max_i |\lambda_i| < 1$, where λ_i is eigenvalue of A?

2.4 Experimental setup

The code was implemented on TensorFlow. Here we introduced hyperparameters and input sequences are generated as the following:

```
m = 20000
n = 1 #size of input x is (m,n)
n_h = 3 #size of hidden layer is (m,n_h)
l = 1 #size of output y is (m,l)
K = 16 #number of mini-batches
max_grad_norm = 5
lr_decay = 0.5
learning_rate = 1.0
init_scale = 0.05
nm_steps = 35
batch_size = int(m/K)


x = np.zeros([m,n], dtype=np.float32)
h = np.zeros([m,n_h], dtype=np.float32)
y = np.zeros([m,l], dtype=np.float32)


C1 = np.float32(np.random.rand(n_h,l))
B1 = np.zeros([n,n_h], dtype=np.float32)
B1[n-1,n_h-1] = 1
D1 = np.float32(np.abs(np.random.randn(n,n)))
A1 = np.diag(np.ones(n_h-1), -1)
cc = np.random.randn(n_h)
cc = cc/(2*np.sum(np.fabs(cc)))
```

```
A1[:,n_h-1] = cc
A1 = np.float32(A1)
```

Here we consider different distributions of input sequence x.

```
x = np.float32(np.random.normal(0,1,[m,n]))
#x = np.float32(np.random.exponential(1,[m,n]))
#x = np.float32(np.random.gamma(1,1,[m,n]))
#x = np.float32(np.random.uniform(0,1,[m,n]))
```

In the following lines of code we introduced a model (2.1):

```
for i in range(len(x)):
    if(i==0):
        h[i] = np.matmul(x[i],B1) #np.zeros([n_h], dtype=np.float32)
    else:
        h[i] = np.matmul(h[i-1],A1) + np.matmul(x[i],B1)


y = np.matmul(h,C1) + np.matmul(x,D1)
    + np.float32(np.random.normal(0,1,[m,l]))
```

After generating the output sequence y, we are going to reconstruct the quadruple (A, B, C, D). Then we describe our model (2.1) on Python as the following:

```
BB = np.zeros([n,n_h], dtype=np.float32)
BB[n-1,n_h-1] = 1
B = tf.constant(BB,shape=[n,n_h])


AA = np.diag(np.ones(n_h-1), -1)
A2 = tf.constant(AA,shape=[self._n_h,self._n_h-1],dtype=tf.foat32)
A3 = tf.get_variable('A', [n_h, 1],
     initializer=tf.random_uniform_initializer
     (-init_scale, init_scale), dtype=tf.float32)
A = tf.concat([A2, A3],1)
```

```
new_h = tf.matmul(inputs, B) + tf.matmul(h, A)
outputs_s = tf.reshape(outputs, [-1, n_h])
logits = tf.matmul(self.x, D) + tf.matmul(outputs_s, C)
self.cost = tf.reduce_mean(tf.square(self.y-logits))
```

The latter line is equation of loss function which should be minimized. Then we minimize loss function (2.3) to get global minimum. We use Adam Optimizer instead of the classical stochastic gradient descent because it has some benefits:

- Straightforward to implement,
- computationally efficient,
- little memory requirements.

```
self.train_op = tf.train.AdamOptimizer(1e-3).minimize(self.cost)
```

2.5 Results

2.5.1 Canonical matrix A

- If matrix A is in canonical form and $\|a\|_1 \leq \frac{\sqrt{2}}{2}$, where a is the last column of matrix A and $x \overset{iid}{\sim} N(0,1)$,then loss function (2.3) of model (2.1) approaches to global minimum. It means that we can rebuild initially given quadruple (A, B, C, D). This result was obtained in [15]. But what if we consider cases out of these condition? Let us consider input sequence x in different distributions (exponential,gamma,uniform).
- For matrix A in canonical form and x in different distributions, loss function can be computed and it also goes to global minimum. Figure 2-1 shows that the objective function (2.3) of model (2.1) converges to global minimum regardless of distribution of input sequence x.

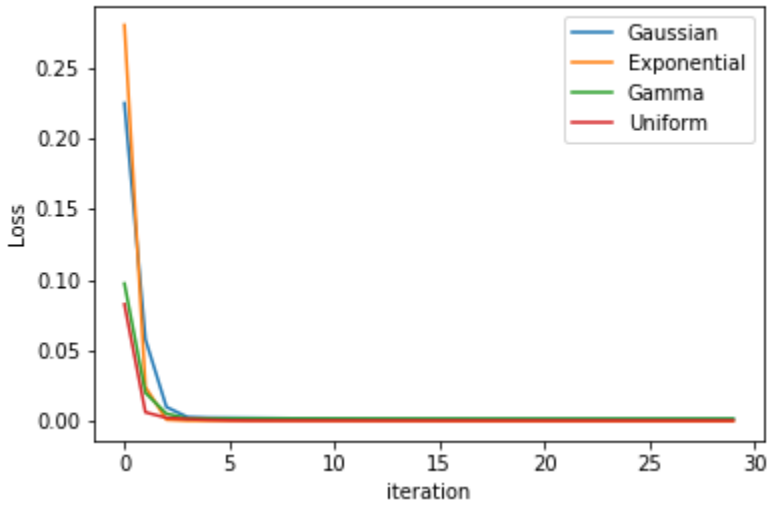


Figure 2-1: Loss function for different distributions of x

- What if we consider a matrix A with $\|a\|_1 > \frac{\sqrt{2}}{2}$? To check this hypothesis we sketched a graph for different values of $\|a\|_1$. In Figure 2-2 we can see that final value of loss function increases for $\|a\|_1 > \frac{\sqrt{2}}{2}$. Since objective function does not converge to global minimum we can not rebuild initial matrices (A, B, C, D).

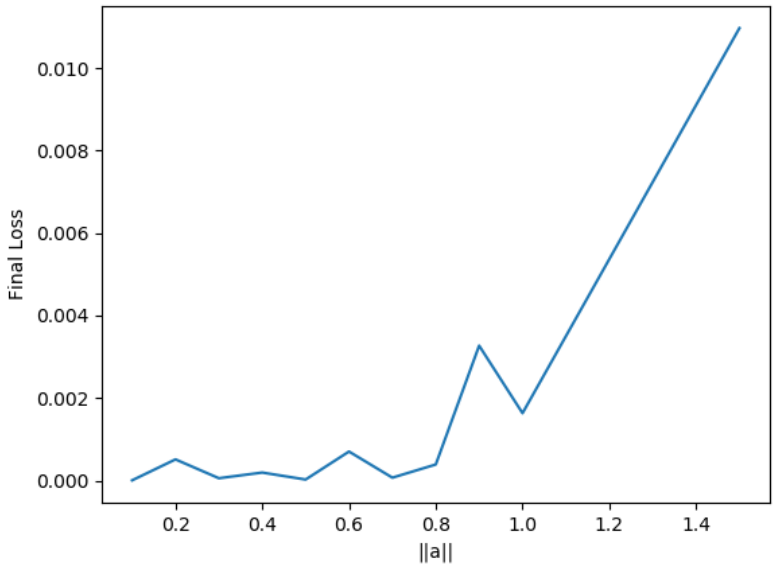


Figure 2-2: Final value of Loss function for different values of $\|a\|_1$

2.5.2 Multidimensional case

In previous experiments we considered input sequence $x = (x_1, x_2, \ldots, x_T)$, where $x_t \in \mathbb{R}$. What if we take input sequence $x = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_T)$ with $\bar{x}_t \in \mathbb{R}^n$? Then we need to change size of B and D. We use a particular state space representation for learning the system in time domain with example sequences and take a matrix B in the following form:

$$B = \begin{pmatrix} 0 \\ 0 \\ I \end{pmatrix}.$$

We made numerical experiments and got the result that objective function approaches to global minimum. At the end we compared (A, B, C, D) with learned matrices (A', B', C', D').

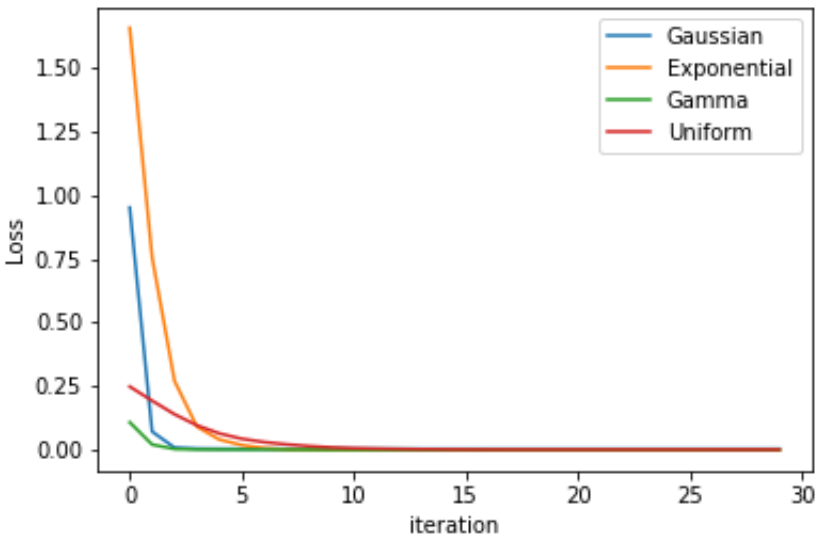


Figure 2-3: Loss function for different distributions of x

2.5.3 Beyond the canonical form

Since

$$h_{t+1} = Ah_t + Bx_t$$

and

$$h_t = A^t h_0 + \sum_{i=0}^{t-1} A^{t-i-1} B x_i,$$

$A^t h_0$ approaches to zero, then h_0 does not affect the output y_t over time t. Thus, one question appears: is it enough to consider a matrix with $\|A\| < 1$?

Hypothesis 2.5.1 *A realization* (A, B, C, D) *of the system* (2.1) *can be reconstructed for any matrix* A *such that* $\|A\| = \max_i \sqrt{\lambda_i} < 1$*, where* λ_i *is an eigenvalue of matrix* $A^T A$*.*

- In previous results we considered matrix A in canonical form, and now we consider any matrix A such that its norm $\|A\| < 1$.

  It is difficult to learn arbitrary linear dynamical systems. According to Definition 1.8.1 and Theorem 1.8.1 a natural bound is *stability*, which requires that all eigenvalues of matrix A are less than 1. Equivalently, the roots of the characteristic polynomial should all be contained in the complex unit disc. Without stability, the state of the system could blow up exponentially which makes learning difficult. But the set of all stable systems forms a non-convex domain. It seems daunting to guarantee that stochastic gradient descent would converge from an arbitrary starting point in this domain without ever leaving the domain.

  After implementation the code we got different matrices, but in previous section we gave a notion of equivalency of matrices. Since the quadruple (A, B, C, D) is equivalent to $(TAT^{-1}, TB, CT^{-1}, D)$ for some invertible matrix T, we compared eigenvalues of initial matrix A and learned matrix A', we also compared D with estimated D'. The only one of eigenvalues are same and the rest diverges. It shows that our hypothesis does not work.

Hypothesis 2.5.2 *A realization* (A, B, C, D) *of the system* (2.1) *can be reconstructed for any matrix* A *such that* $\rho(A) = \max_i |\lambda_i|$*, where* λ_i *is an eigenvalue of matrix* A*.*

- Now we consider a matrix A in any form such that its eigenvalues are less than one. As in the previous, after implementation the code we got different matrices

and then we compared eigenvalues of matrices A and A'. Again, the only one of eigenvalues are same and the rest diverges. Our hypotheses do not work.

2.6 Research schedule

No.	Tasks, actions to implement thesis tasks	Work period	Months of thesis implementation, expectations for thesis			
			January	February	March	April
1	Experimental verification of Pacman criterion on linear models	1 month	Conditions for the applicability of Pacman criterion			
2	Software implementation of an algorithm for the identification problem	1 month		Python-based software		
3	Making synthetic data: selecting a class of dynamical systems, sampling their dynamics	1 month			Data bank containing processes	
4	Numerical identification of systems parameters for data bank processes, testing of Pacman criterion for optima. Experiments with the Gershgorin criterion.	1 month				Conditions for the applicability of Pacman criterion, preliminary report

Table 1: Research schedule

Bibliography

[1] J. Schmidhuber. Deep Learning in Neural Networks: An Overview. Neural Networks, Volume 61, January 2015, Pages 85-117.

[2] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. Neural Computation, 9(8):1735âĂŞ1780, 1997.

[3] Tomas Mikolov, Armand Joulin, Sumit Chopra, Michael Mathieu and Marco Aurelio Ranzato, Learning Longer Memory in Recurrent Neural Networks, https://arxiv.org/pdf/1412.7753.pdf

[4] J. G. Zilly, R. K. Srivastava, J. Koutnik and J. Schmidhuber. Recurrent Highway Networks. International Conference on Machine Learning (ICML 2017)

[5] Rumelhart D.E., Hinton G.E., Williams R.J., Learning Internal Representations by Error Propagation. In: Parallel Distributed Processing, vol. 1, pp. 318âĂŤ362. Cambridge, MA, MIT Press. 1986

[6] Christoph H. Lampert, Predicting the Future Behavior of a Time-Varying Probability Distribution. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 942-950

[7] Lennart Ljung. *System Identification. Theory for the user.* Prentice Hall, Upper Saddle River, NJ, 2nd edition, 1998.

[8] M. Vidyasagar and Rajeeva L. Karandikar. A learning theory approach to system identification and stochastic adaptive control. *Journal of Process Control,* 18(3):421-430, 2008.

[9] M. C. Campi and Erik Weyer. Finite sample properties of system identification methods. *IEEE Transactions on Automatic Control*, 47(8):1329-1334, 2002.

[10] Erik Weyer and M. C. Campi. Finite sample properties of system identification methods. In *Proceedings of the 38th Conference on Decision and Control*, 1999.

[11] Christiaan Heij, Andre Ran, and Freek van Schagen. *Introduction to mathematical systems theory : linear systems, identification and control*. Basel, Boston, Berlin, 2007.

[12] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points-online stochastic gradient for tensor decomposition. In *Proc. 28th COLT*, pages 797-842, 2015.

[13] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht. Gradient Descent Converges to Minimizers. *ArXiv e-prints*, February 2016.

[14] E. Hazan, K. Y. Levy, and S. Shalev-Shwartz. Beyond Convexity: Stochastic Quasi-Convex Optimization. *ArXiv e-prints*, July 2015.

[15] Moritz Hardt, Tengyu Ma, Benjamin Recht. Gradient Descent Learns Linear Dynamical Systems. https://arxiv.org/abs/1609.05191