1 Introduction

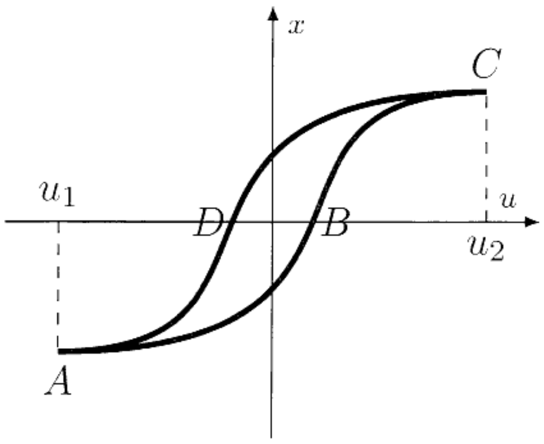
Hysteresis (see Figures 1.1a to 1.1d) is defined as a *rate independent* process with *memory effect* [Visintin, 2013, p. 13-14]. It's ubiquitous in various fields, including microelectronics [Bondurant and Gnadinger, 1989, Jiles and Atherton, 1983], materials [Kaltenbacher and Krejcí, 2014, Krejčí and Sprekels, 2007, Al Janaideh et al., 2009, Ge and Jouaneh, 1995], mechanics [Truesdell and Noll, 2004, Krejčí et al., 2014, Kunze and Marques, 2000], economics [Belke et al., 2013, Göcke, 2002, Belke et al., 2014, Blanchard and Summers, 1986], etc. The nonlinear operators ~~like~~ such as *non-ideal relay operator* (see Figure 1.1b), *stop operator* ~~[Krejci, 1996]~~ (see Figure 1.1c), *play operator* ~~[Krejci, 1996]~~ (see Figure 1.1d) and their generalizations are used as essential blocks to model dynamical systems with hysteresis. [Krejci, 1996] Preisach-type model, constructed as a superposition of non-ideal relays, is quite general in applications and it's equivalent to a linear combination of *generalized plays* [Visintin, 2013, p. 110-111, Theorem 2.7].

In this thesis, we want to approximate a wide class of hysteretic processes, Preisach-type model, by recurrent neural network. Intuitively, recurrent neural network (RNN) is a optional approach to approximate systems with *memory* since it allows previous output to be used as input while having hidden states. Wang et al. [2018] applied internal time-delay RNN to describe the hysteresis and showed promising performance that their networks described both the major and minor hysteresis loops well. However, they assumed hysteretic operator is obtained in advanced so that they didn't learn this *nonsmooth* operator by neural networks in their model. We also checked long short-term memory (LSTM) networks [Hochreiter and Schmidhuber, 1997], the state-of-the-art architecture of RNN, and found that it didn't perform well enough to reveal the relations between output and original input in hysteretic systems. ~~In order to achieve better performance,~~ Hence, in this master thesis **we develop a new neural network architecture, namely hysteretic neural network (HNN) (see Figure 1.3)**. It is a realization of a linear combination of generalized plays and hence it's able to approximate any *Preisach operator* [Visintin, 2013].

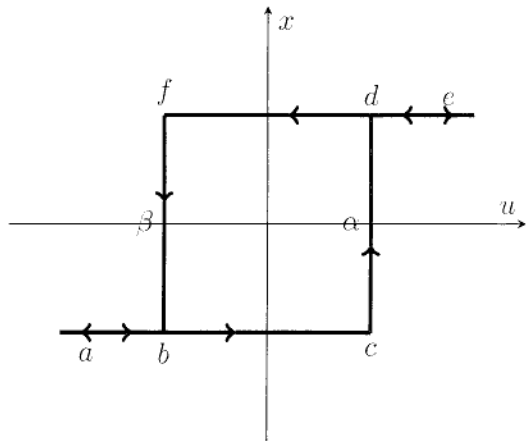
what does this mean?

Given an observation (x_n, y_n) $(n = 1, \ldots, N)$ underlying hysteretic input-output relations, we trained both LSTM and HNN ~~minimize~~ by minimizing the mean square error (MSE) between predicted target $\hat{y}_n$ and observed target y_n. **It shows HNN outperforms LSTM by comparing root mean square error (RMSE) (see Figure 1.2). In particular, HNN is able to reconstruct minor hysteresis loops well whereas LSTM fails.**

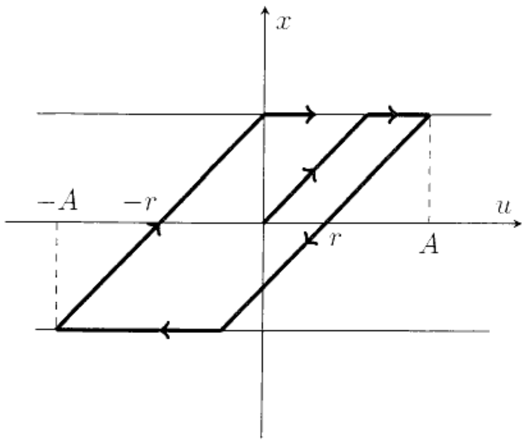
Further, ~~Based on the HNN we obtain,~~ we study a particular application of HNN, namely momentum-based trading strategies in financial market. ~~with hysteretic property in economic.~~ Krejčí et al. [2014] proposed a market model and used Prandtl-Ishlinskii operator, a particular case of the Preisach model, to model trading strategies within their market model. It provides a promising insight to make play-hysteresis economic models compatible with multi-agent



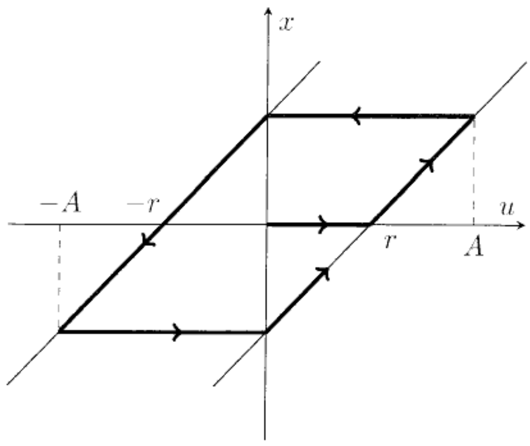
(a) hysteresis loops



(b) non-ideal relay



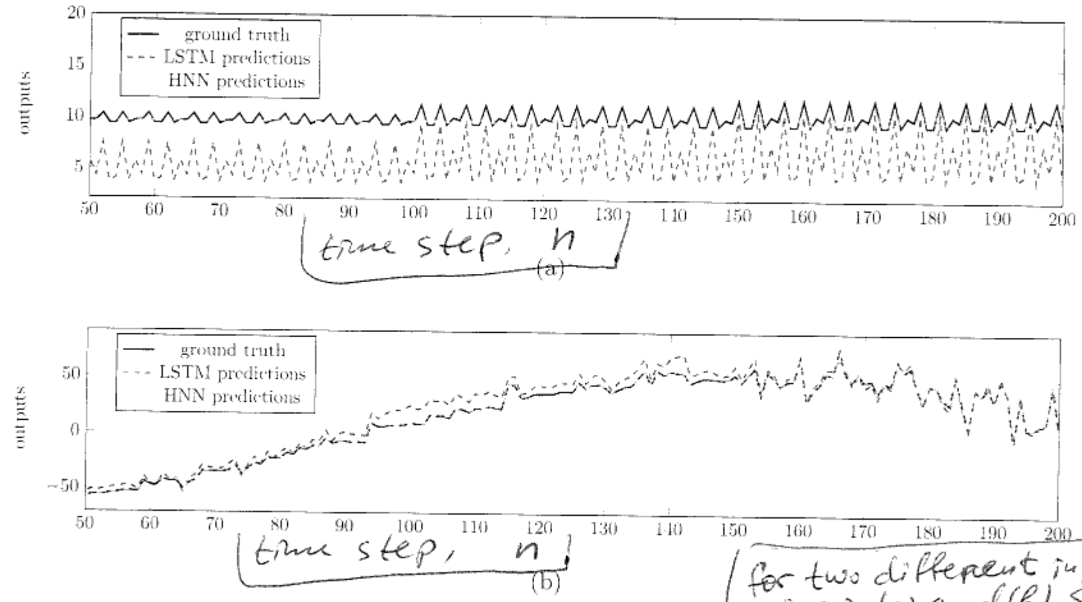
(c) stop



(d) play

Figure 1.1: Interpretation of simplest *hysteresis loop*, *non-ideal relay*, *stop* and *play*. (Figure 1.1a) If u monotonically increases from u_1 to u_2, then the coordinate (u, x) moves along the path $A \to B \to C$; conversely, if u monotonically decreases from u_2 to u_1, then (u, x) moves along the path $C \to D \to A$ [Visintin, 2013, p. 12-13]. (Figure 1.1b) Hyper-parameters α and β correspond to *on* and *off* switching values of input, respectively. As the input u monotonically increased, the ascending branch $a \to b \to c \to d \to e$ is followed. When the input is monotonically decreased, the descending branch $e \to d \to f \to b \to a$ is traced [Mayergoyz, 1986, p. 2]. (Figure 1.1c, Figure 1.1d) Input-output diagram for *stop* and *play* in the case $\dim X = 1, Z = [-r, r], u(t) = A\sin(\omega t)\, for\, A > r > 0$ [Krejci, 1996, p. 9]

modeling framework [Cross et al., 2007, Cartwright et al., 1999, Lamba and Seaman, 2008]. However, they only considered single trading strategy that agents take reaction



for two different input sequencies in (a) and (b). See details in Sec...

Figure 1.2: The overall predictive outputs against ground-truth outputs. TODO: detail interpretation of input and output sequence or not ?

who is called "agent D"?

to the change of *price trend*, called *agents D* (cf. ??) in this thesis, to simulate market movements. Another trading strategy, which is common in financial trading pattern as well, is ~~also~~ threshold-based where agents in markets are sensitive to the fluctuation of some *fixed price value* instead of *price trend*, called *agents N* (cf. ??) in this thesis. **We generalize Krejčí et al. [2014]'s market model by introducing two different agents, agents D and agents N based on ~~generalized~~ play operator and non-ideal relay operator respectively.**

reformulate

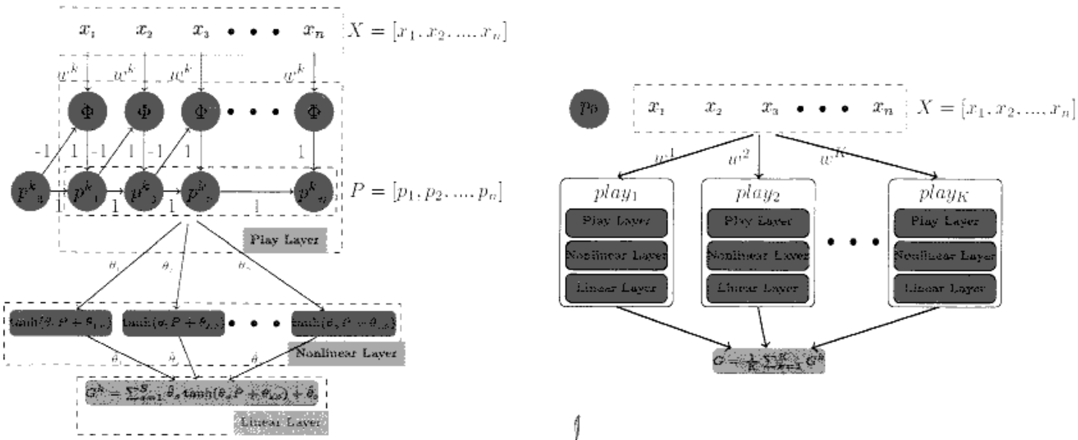
~~Again,~~ We learn this financial market model using HNN and LSTM, by maximizing log-likelihood of price distribution to train both networks. It reveals HNN models this kind of market model better than LSTM. ~~Even HNN is possible to reconstruct the *unobserved state changes underlying the market*, which is useful to interpret the *avalanche of market movements*, since it inherently contains agents that use different trading strategies in micro level.~~

see Sec ? for details

reformulate, bad style

and is able to predict avalanches of prices.

The thesis is organized as follows. In the next chapter, a detail methodology for HNN is discussed, including how to formulate hysteretic systems by generalized play operator and learn the neural networks. In chapter 3, we presents detail discussions about our financial market model and provide a practical approach to generate synthetic data for further evaluation. In chapter 4, we formulate the methods to learn our market model. In chapter 5, we evaluate performance between LSTM and HNN in different degrees, including the complexity of data sets, network complexity and the accuracy of predicted results. The last chapter contains our conclusions and future works.



(a) generalized play operator

(b) linear combination of multiple generalized play operator to approximate Preisach operator

Figure 1.3