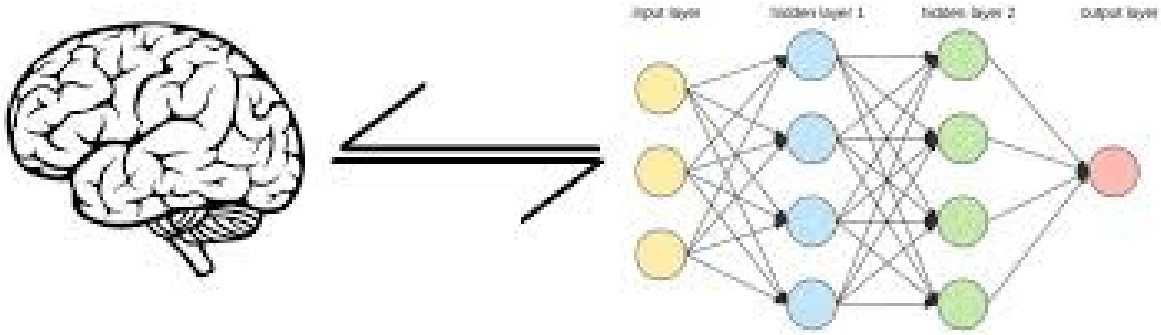
Learning Bijections

Can Duyar

Advisor: Dr. Zafeirakis ZAFEIRAKOPOULOS

20 October 2021

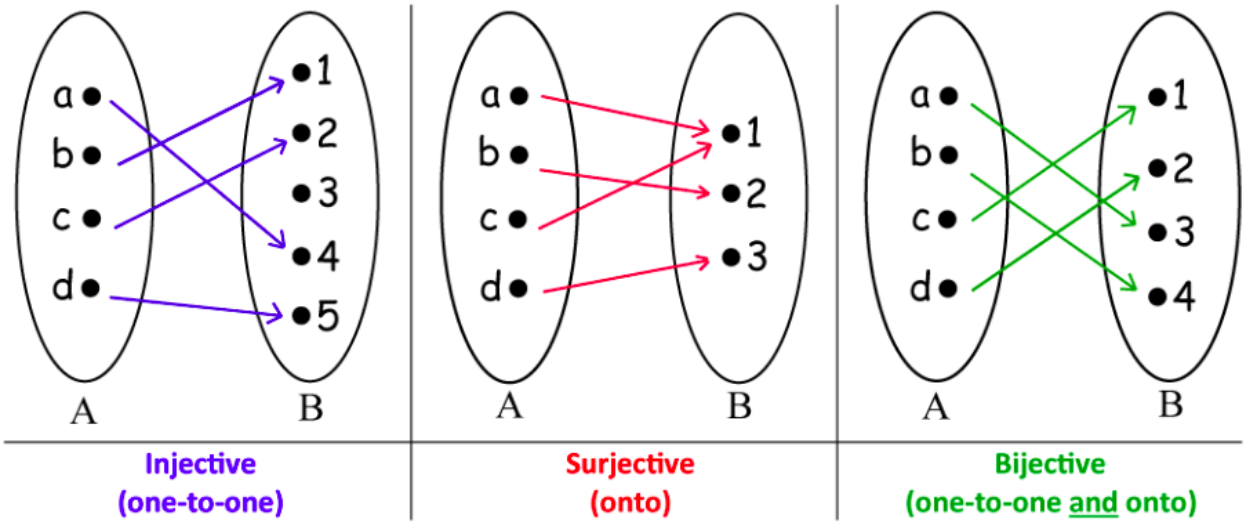
Project Definition



My project is related with training a neural network to act as a bijection. I am going to pick two sets of combinatorial objects that are known to have the same cardinality and train a network to act as a bijection. In this case, I am going to have different output for each input.

The goal is to investigate best practices to develop a general framework in the direction of learning bijections.

WHAT IS A BIJECTION?



Bijection is a function between the elements of two sets, where each element of one set is paired with exactly one element of the other set, and each element of the other set is paired with exactly one element of the first set. There are no unpaired elements.

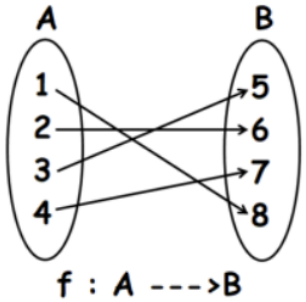
Example of bijection:

Let f: A $\rightarrow$ B. A, B and f are defined as

A = {1, 2, 3, 4}

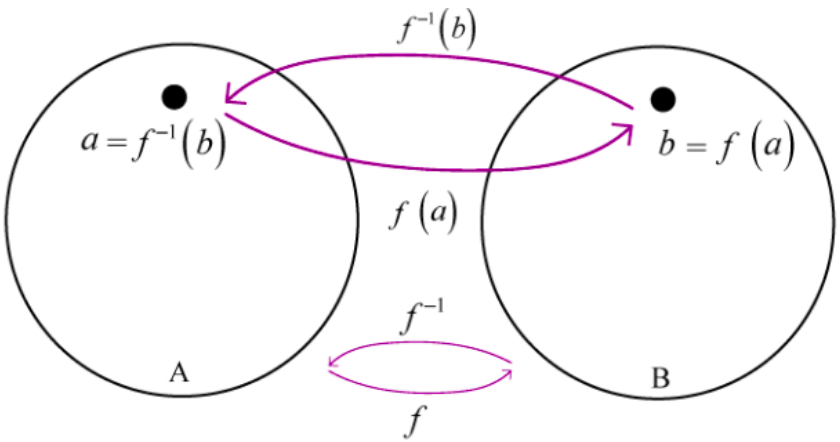
B = {5, 6, 7, 8}

f = {(1, 8), (2, 6), (3, 5), (4, 7)}



- Every element of B has a preimage in A. So f is onto function.
- Every element of A has a different image in B. That is, no two or more elements of A have the same image in B. So f is one to one function.Therefore, f is one to one and onto or bijective function.

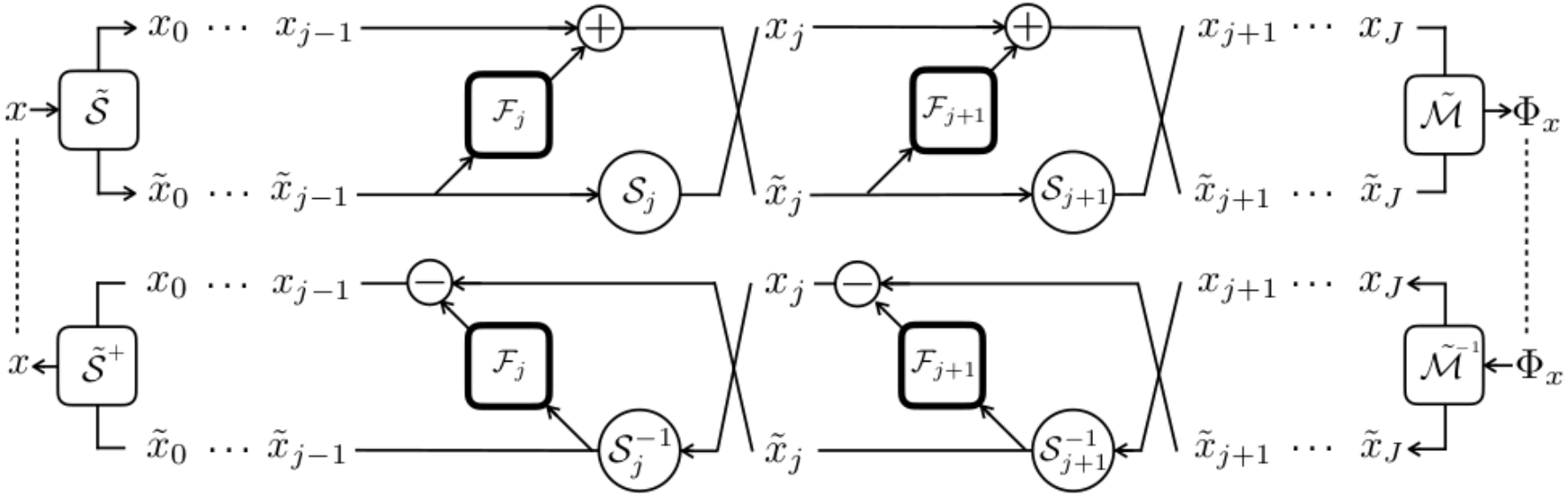
My first approach to solving the problem:



We know that if a function is both injective(one-to-one) and surjective(onto) then this function is a bijective function and is invertible.Therefore, I consider using invertible neural networks to learn behavior of bijections during the implementation of the project.

One of the possible architectures for solving the problem:

i-RevNet(Deep Invertible Networks): i-RevNets define a family of fully invertible deep networks, built from a succession of homeomorphic layers.



Project Requirements

- The programming language used in the project will be "Julia".
- I will use K-net framework. It is the Koç University deep learning framework implemented in Julia.
- I have to use LaTeX for documentation.

Success Criteria

- Minimum %95 accuracy value during test.
- Speeding up the training process with an efficient architecture
- Creating the most suitable dataset for the project

References I

- Lynton Ardizzone, Jakob Kruse, Sebastian Wirkert, Daniel Rahner, Eric W. Pellegrini, Ralf S. Klessen, Lena Maier-Hein, Carsten Rother, and Ullrich Köthe, *Analyzing inverse problems with invertible neural networks*, 2019.
- Jens Behrmann, Will Grathwohl, Ricky T. Q. Chen, David Duvenaud, and Jörn-Henrik Jacobsen, *Invertible residual networks*, 2019.
- Jörn-Henrik Jacobsen, Arnold Smeulders, and Edouard Oyallon, *i-revnet: Deep invertible networks*, 2018.