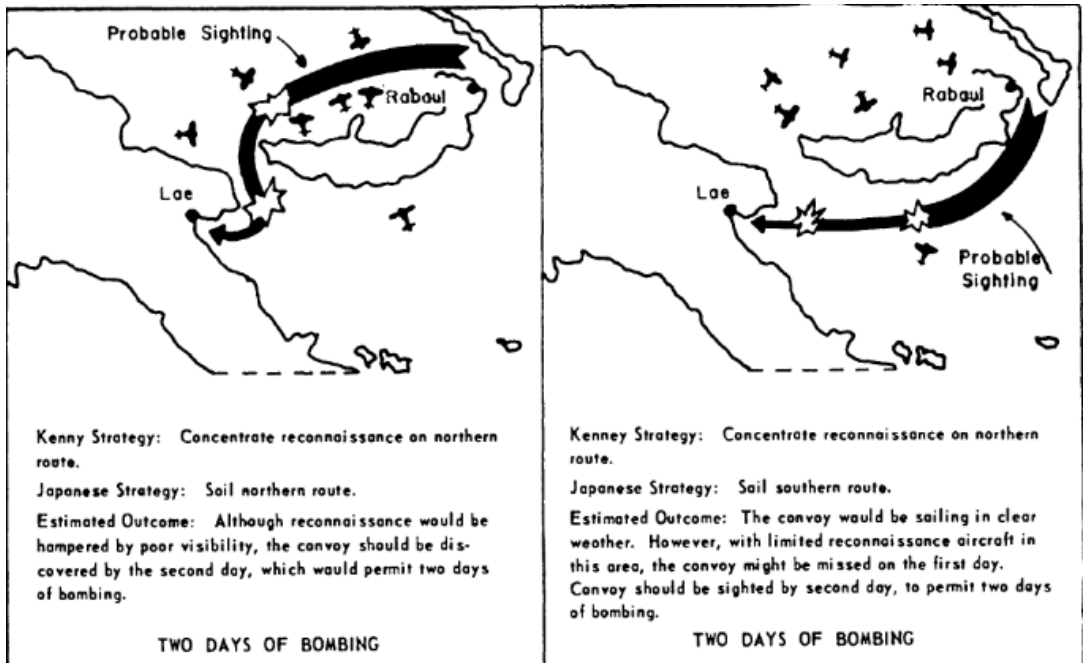
DC GAN – GO GAME

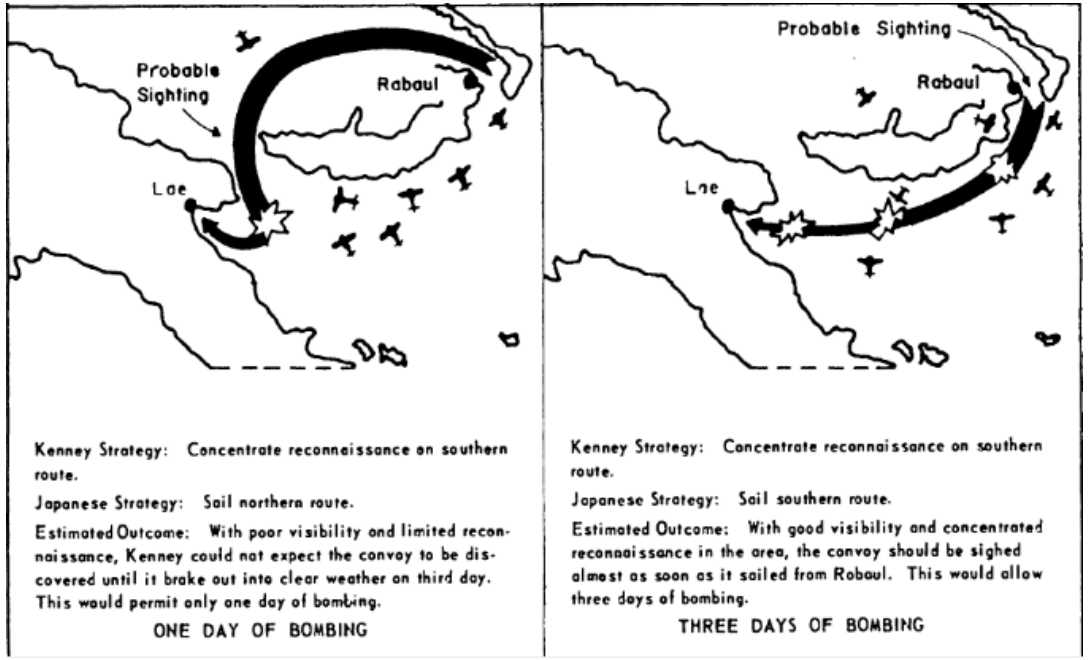
Deep Learning Final Project

SHAOQING TAN

SHIHAN SUN

RUI YUE





INTRODUCTION

Deep Convolutional Generative Adversarial Networks

“The underlying idea behind GAN is that it contains two neural networks that compete against each other in a zero-sum game framework, i.e. generator and a discriminator.”

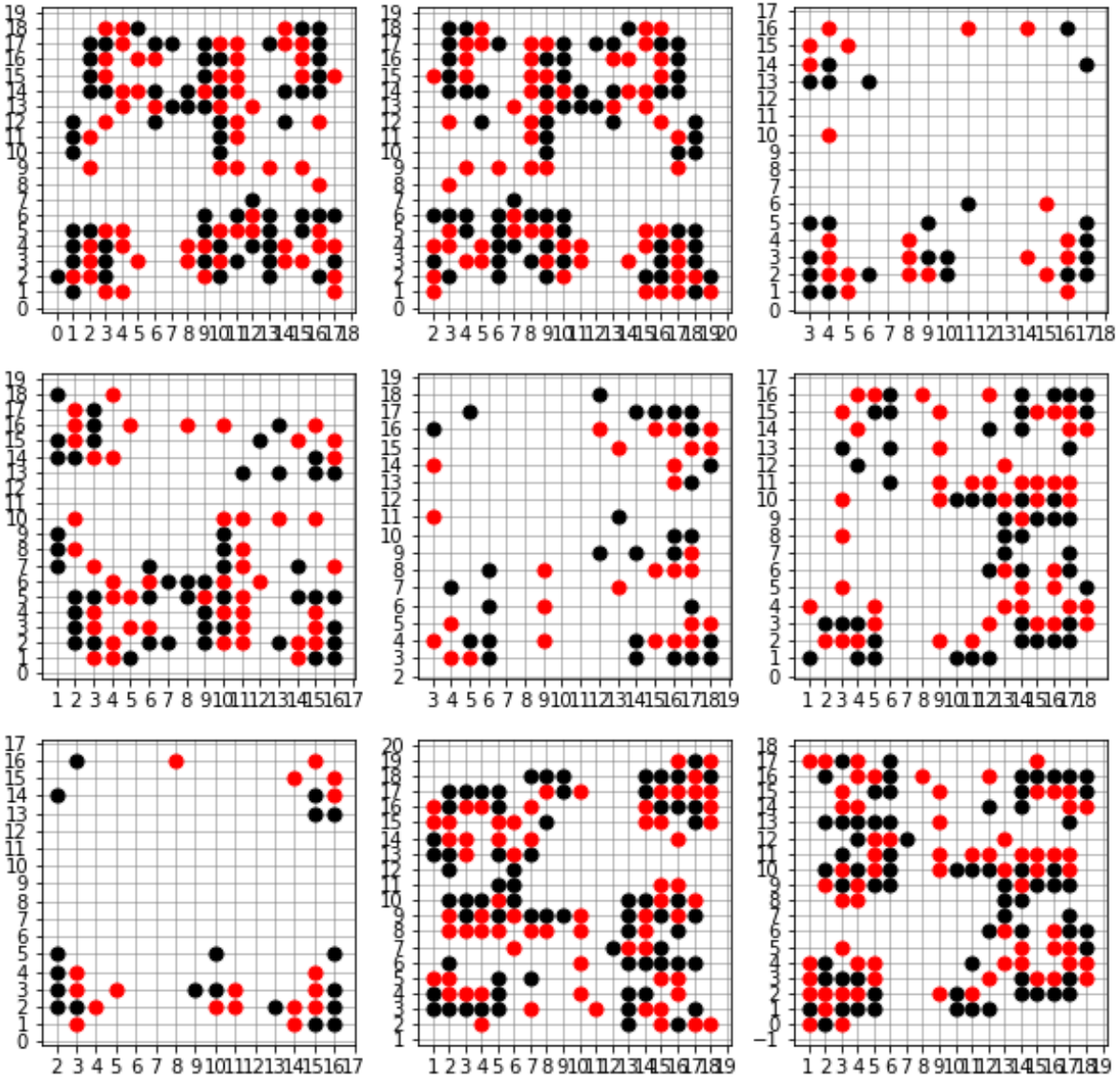
Game Theory (Minimax)

		Japanese strategies		Minimum of row
		#1	#2	
Kenney strategies	#1	2 days	2 days	2 days (maximin)
	#2	1 day	3 days	1 day
Maximum of column		2 days (minimax)	3 days	

How to Play GO

Data preparation

- Source
  - github https://github.com/timlyrics/gangengo
  - Download with git clone
- Format
  - Kifu formatted zipped text files
    `B[pd];W[dd];B[pp];W[dp];`
  - From old professional and AI games
- Extraction
  - Over 80,000 games
  - Remove bad characters
- Unrolling
  - Replay each step and record snapshot
  - Bitmap of 20x20 (padded from 19x19)
  - Two binary channels
  - Transpose and mirror

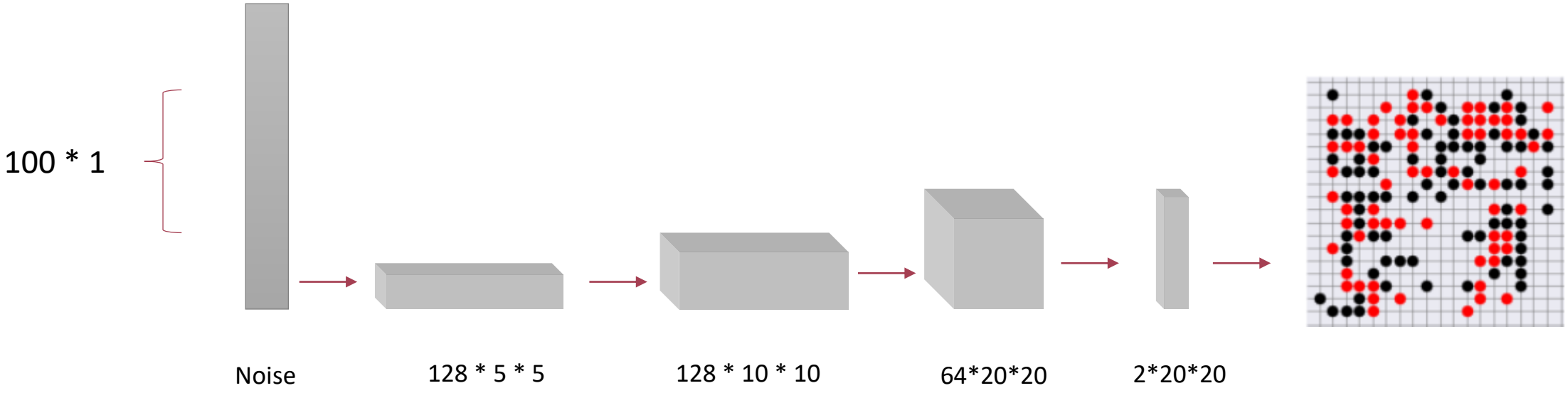


Data preparation

- Packaging
  - Over 512 GB of raw data
  - That’s too much! Even after zipping
  - Dynamically unroll kifu while training
- Unsupervised training
  - All data is considered real (1)
  - Random noise is considered fake (0)
- Testing & validation
  - No objective way to either test or validate
  - Observe loss function to have confidence
- Example snapshot

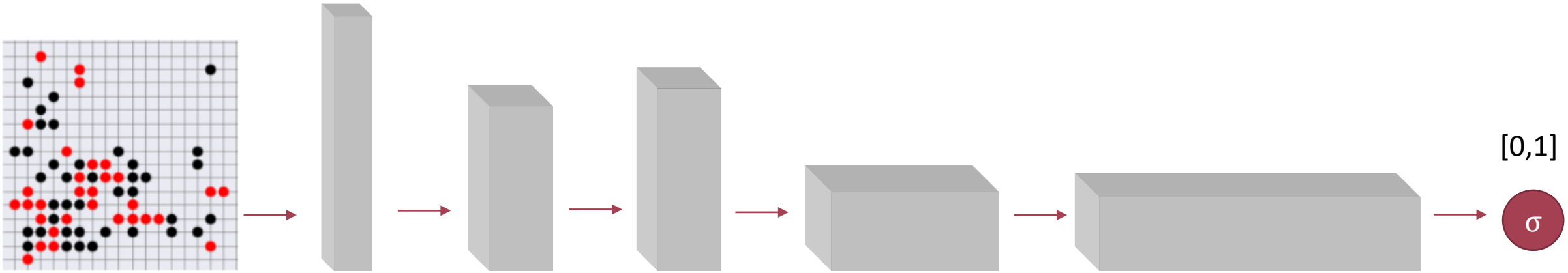
Generator

- Generative Model
- Selection of layers
- Regularization Method: Dropout
- Parameter
- Loss function

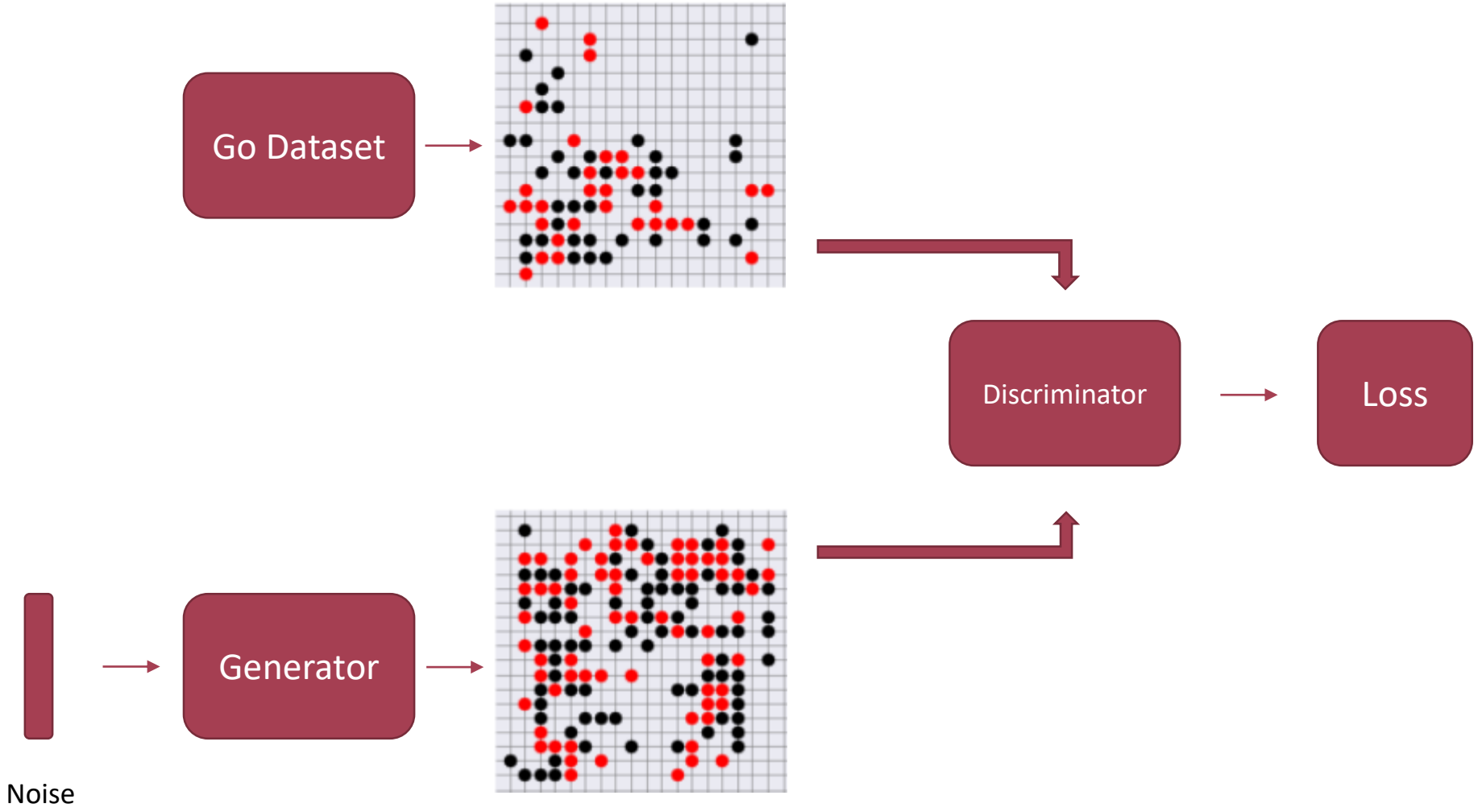


Discriminator

- Discriminator Model
- Selection of layers
- Regularization Method: Dropout
- Parameter
- Loss function



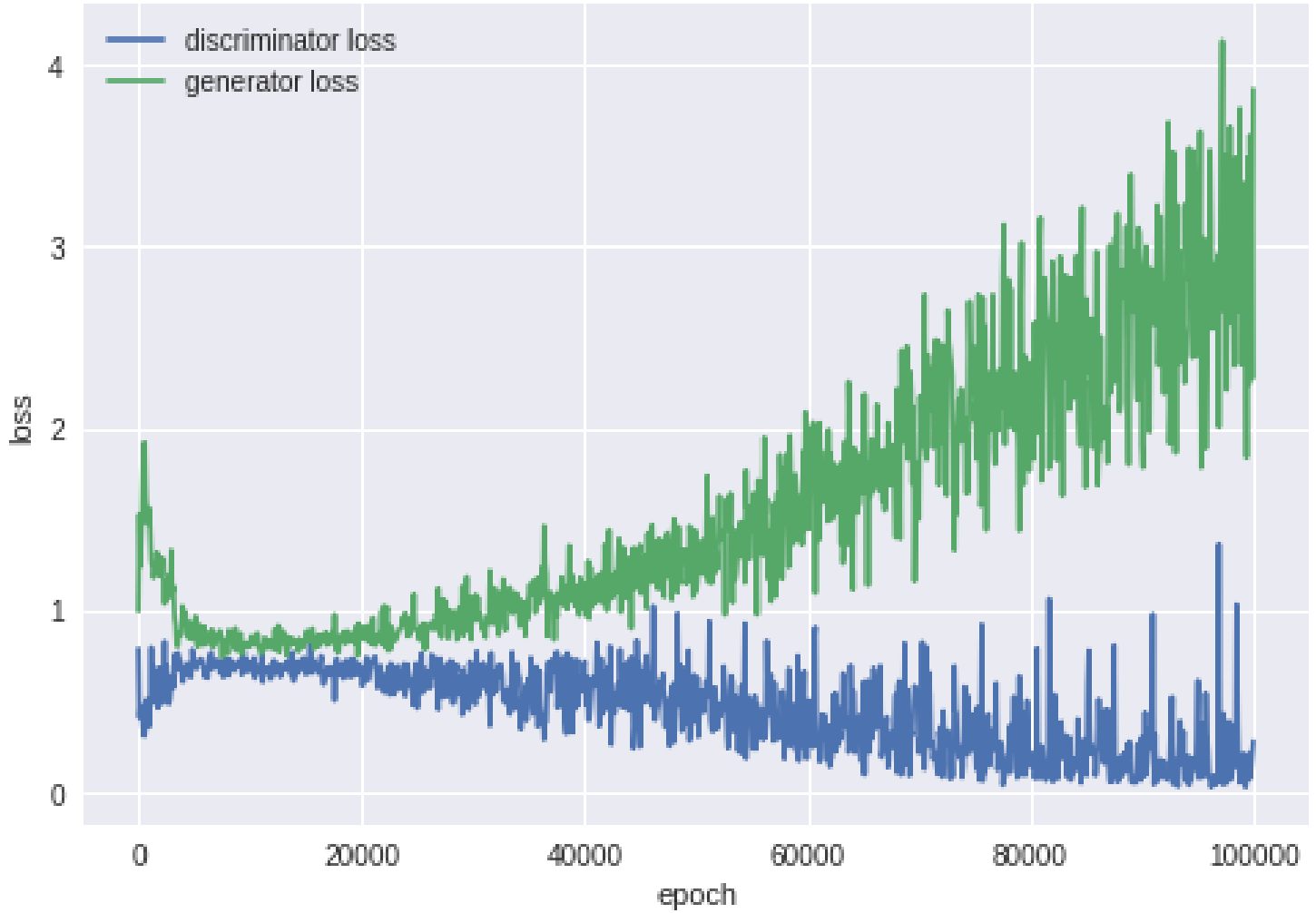
Loss Function



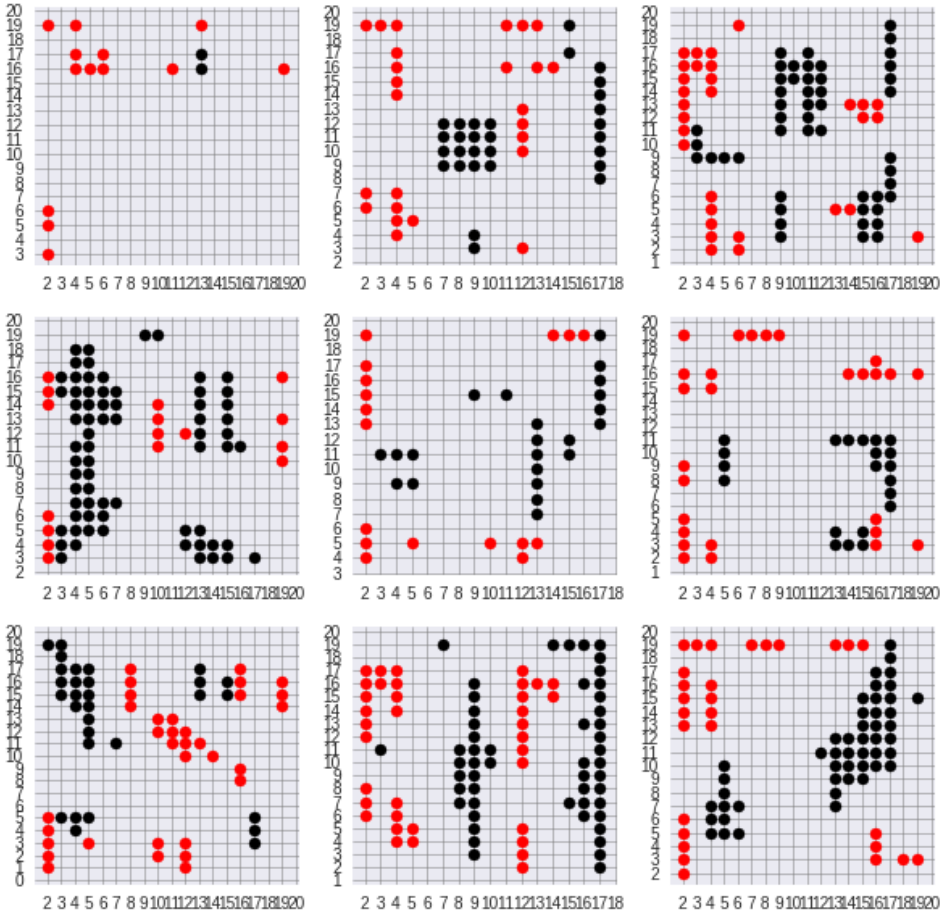
Fine Tune

- Deviated from DC Gan reference design
  - Each pixel carries two channel
  - Two channels do not coincide
  - Each channel is binary
  - Go has global and local logic
- Learning rate
  - Discriminator can easily overpower
  - gen=0.0001 disc=0.00002 (decay = 1e-6)
- Batch size
  - Must feed the beast slowly
  - Batch size = 128
- Generator
  - Output activation is sigmoid
  - Latent dimension is reduced to 32
- Discriminator
  - First CNN layer has no zero padding

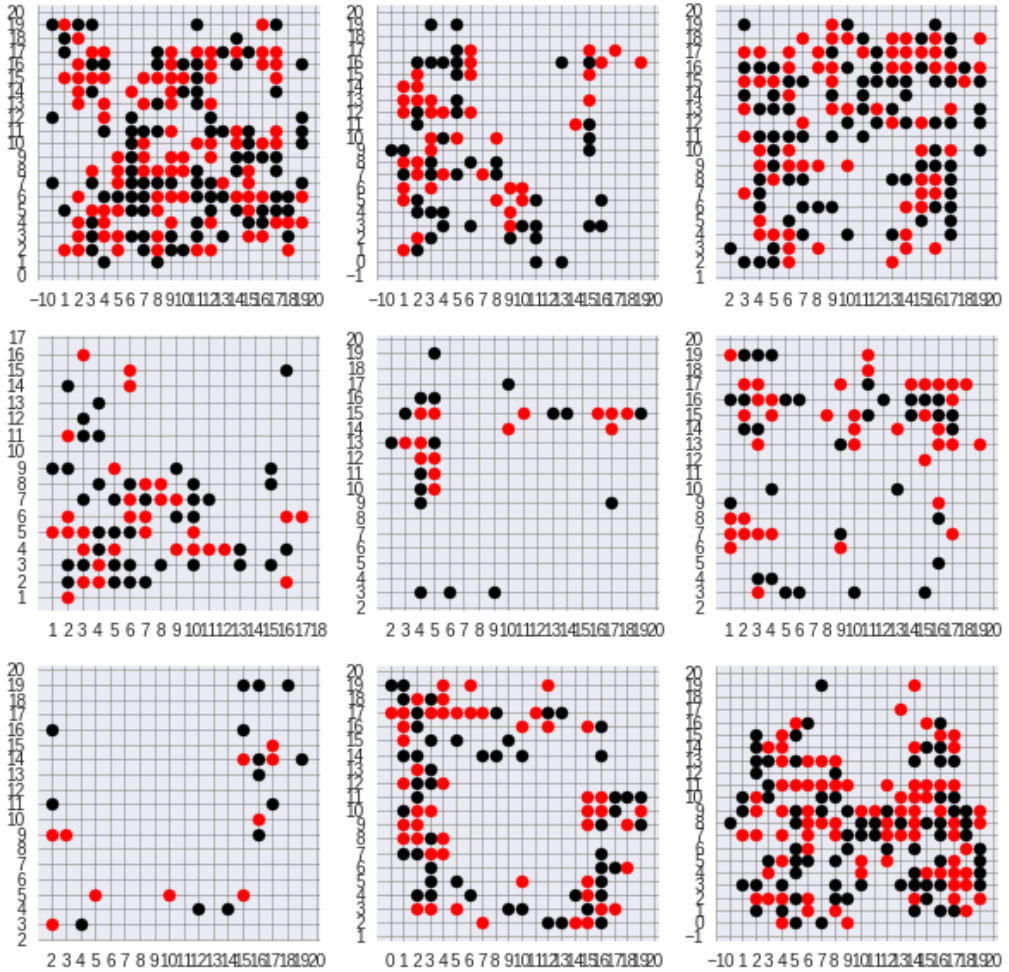
Result – Loss Function



Result - Output



500 Epochs



100,000 Epochs

Question


THANK YOU