THE APPLICATION OF DEEP LEARNING TECHNIQUES TOWARDS PARTICLE IDENTIFICATION AND HIGH ENERGY PHYSICS EVENT SIMULATIONS

CG VILJOEN – MSC DATA SCIENCE

16 AUGUST 2019

AIMS

- 1. Particle Identification
  - $e^- \; vs \; \pi$
  - Based on Raw Data From TRD
- 2. Distinguishing simulated data from real data
  - $\pi_{geant} \; vs \; \pi_{real}$
- 3. Deep Generative Models Towards Event Simulation
  - Variational Autoencoders
  - Generative Adversarial Networks

MOTIVATION: QUARK GLUON PLASMA

- $10^{-43}s$ after initial expansion of the universe → prevailing temperature ≈ $10^{19}GeV$
- Due to this High Energy Density, Quarks and Gluons (normally confined within bound states of colour neutral hadrons) propagated freely in the Early Universe
- This Deconfined Space-Time QGP Expansion Phase existed down to a temperature $150\ MeV$
- For the first time, we can reproduce this on a small scale in Pb-Pb Collisions at ALICE, where a miniscule Space-Time Domain of QGP is formed
- QGP is studied indirectly by analysing the traces left by decay products from the subsequent hadronization process

- Accurate PID is essential in this process
- HEP analyses are Informed by Accurate Event Simulations
- Deep Generative Models are a lot faster than Monte Carlo

PRESENTATION OUTLINE

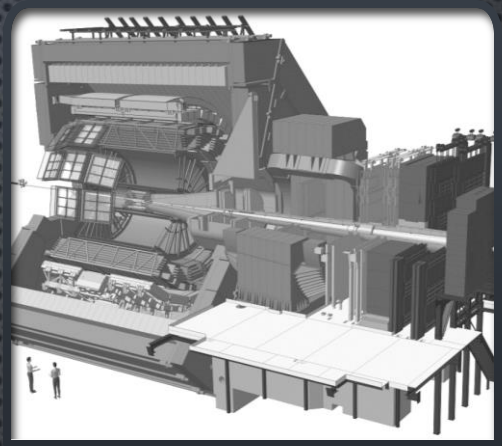
The ALICE Detector System & the TRD

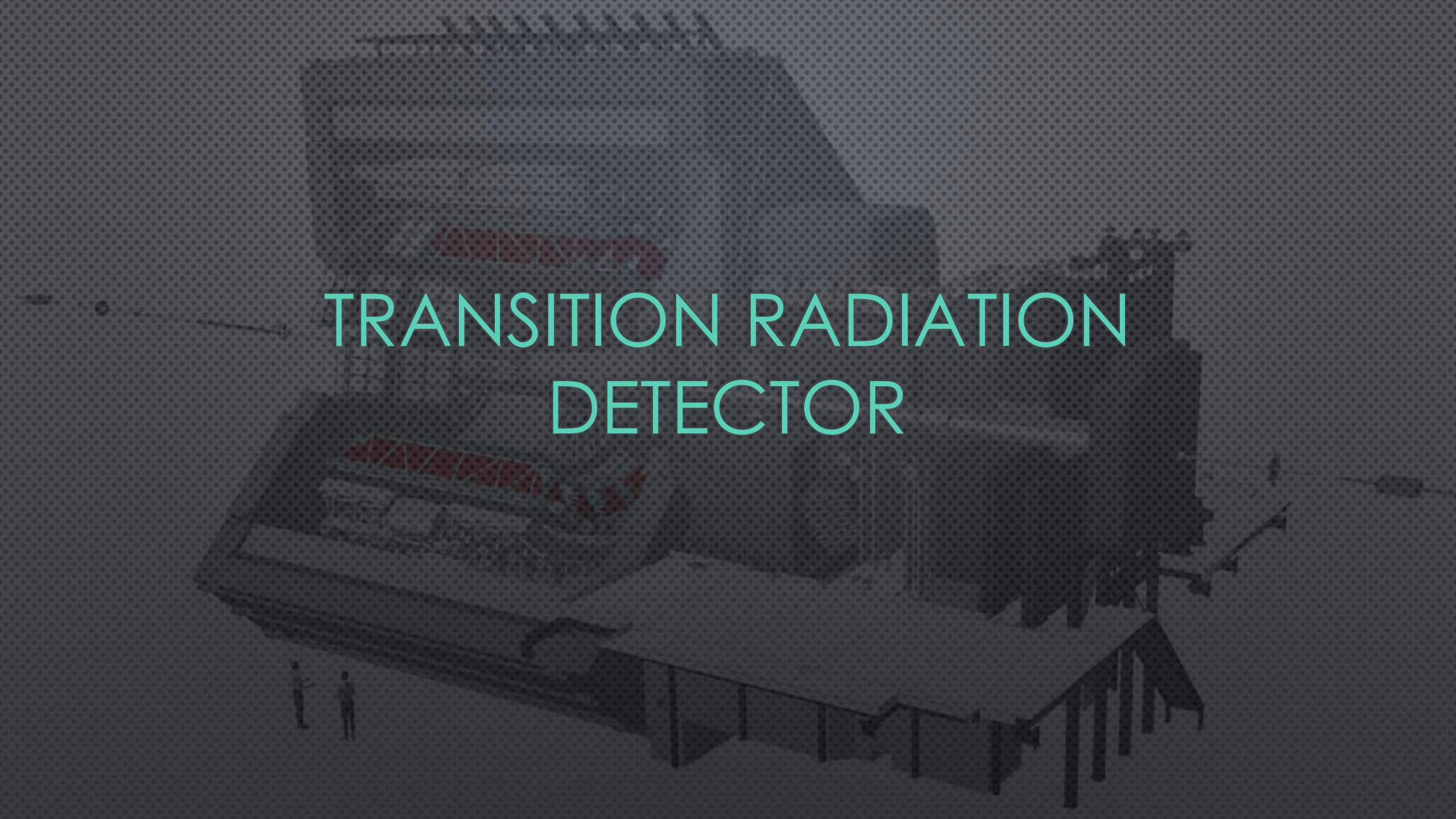
Deep Learning

Latent Variable/ Deep Generative Models

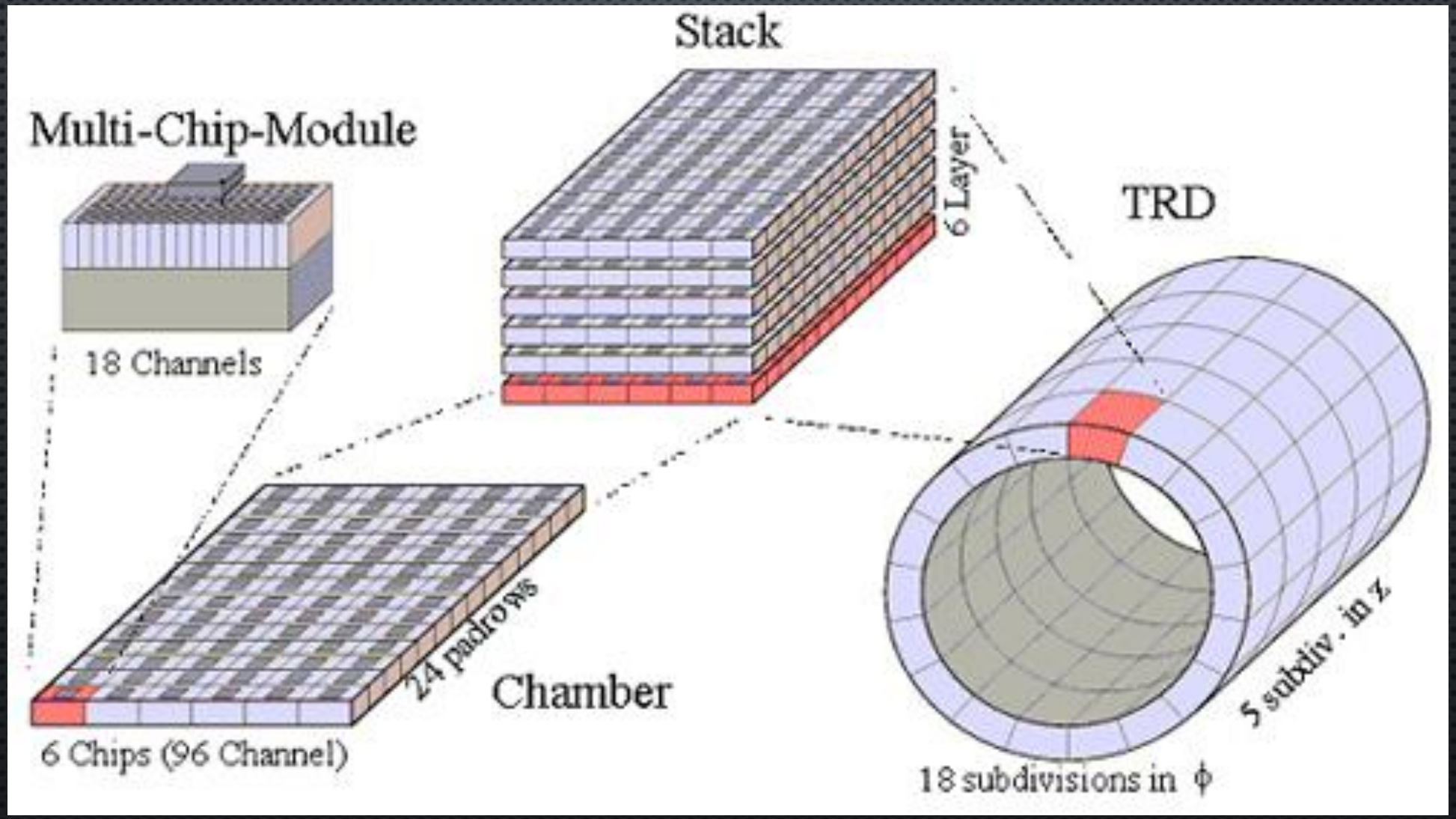
THE ALICE DETECTOR SYSTEM

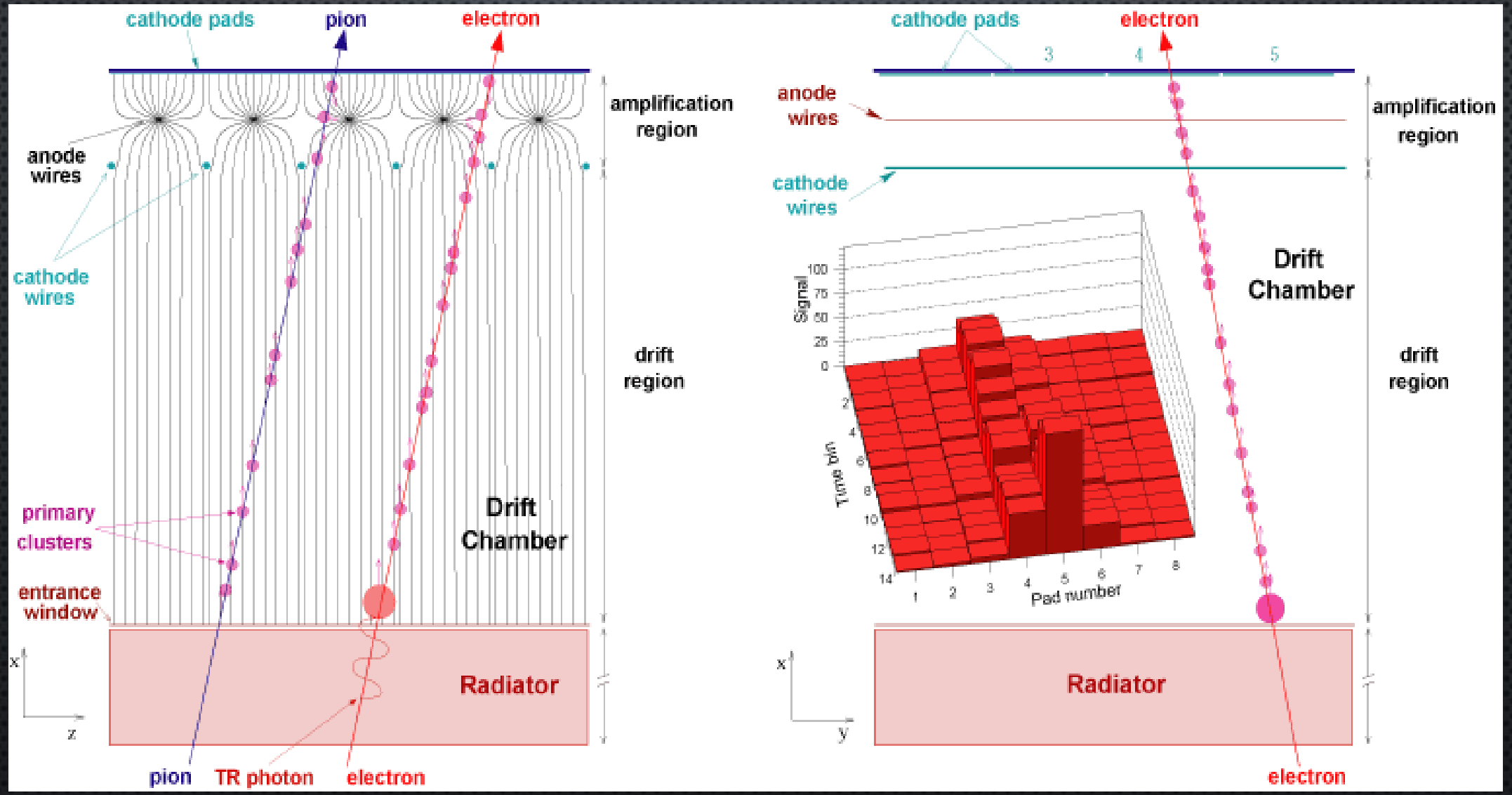
- 10 000 tonnes
- 26m x 16m x 16m
- 18 stacked subdetectors
  - Tracking Systems
  - Electromagnetic and Hadronic Calorimeters
  - Outermost Muon system
- Important Functions
  - Pion Rejection at $P > 1Gev/c$
  - Electron identification, by supplementing its own data with that from the TPC and ITS
  - Event triggering (Based on Collision centrality, etc.)
  - Informs central Barrel's Calibration

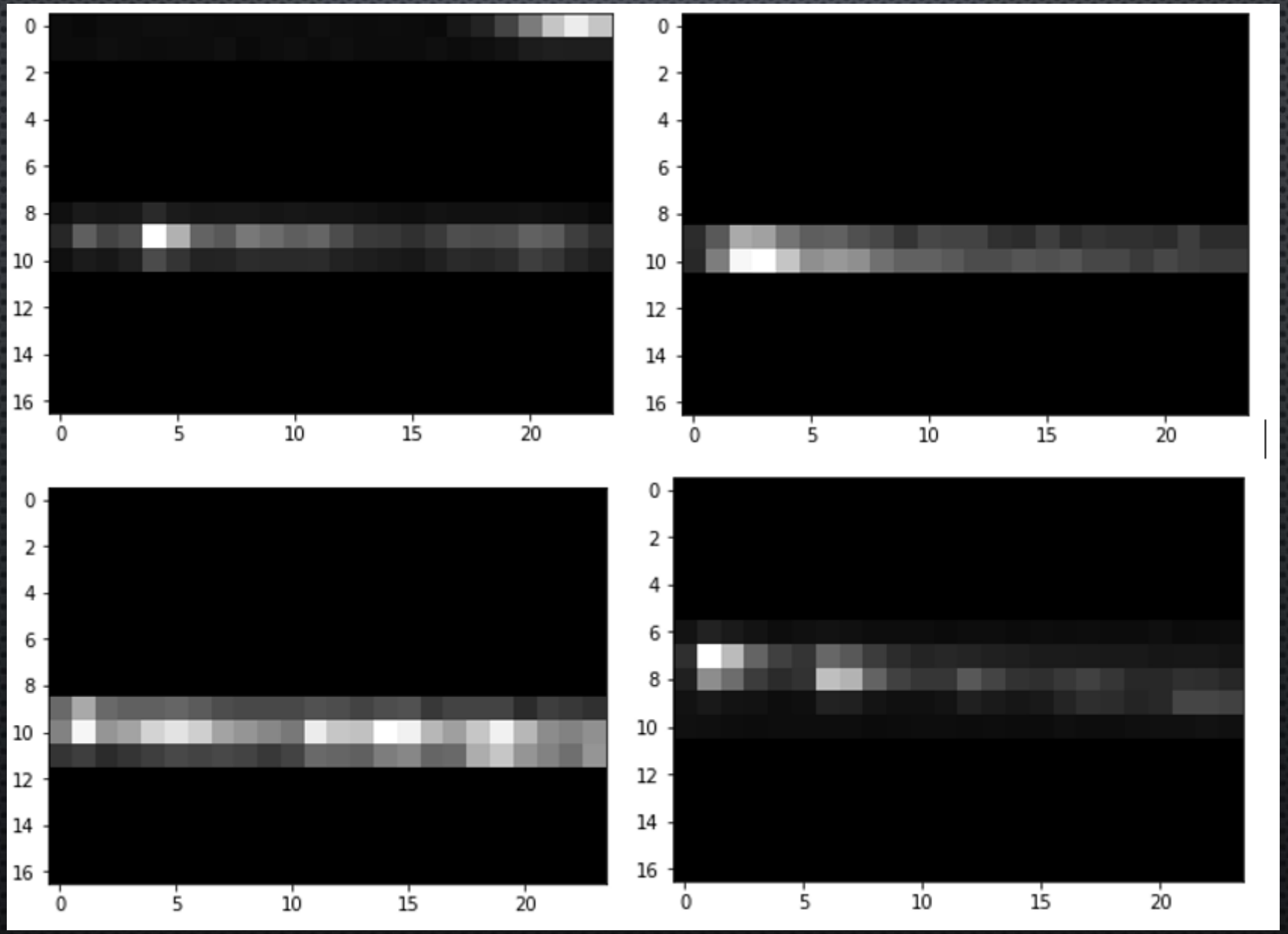




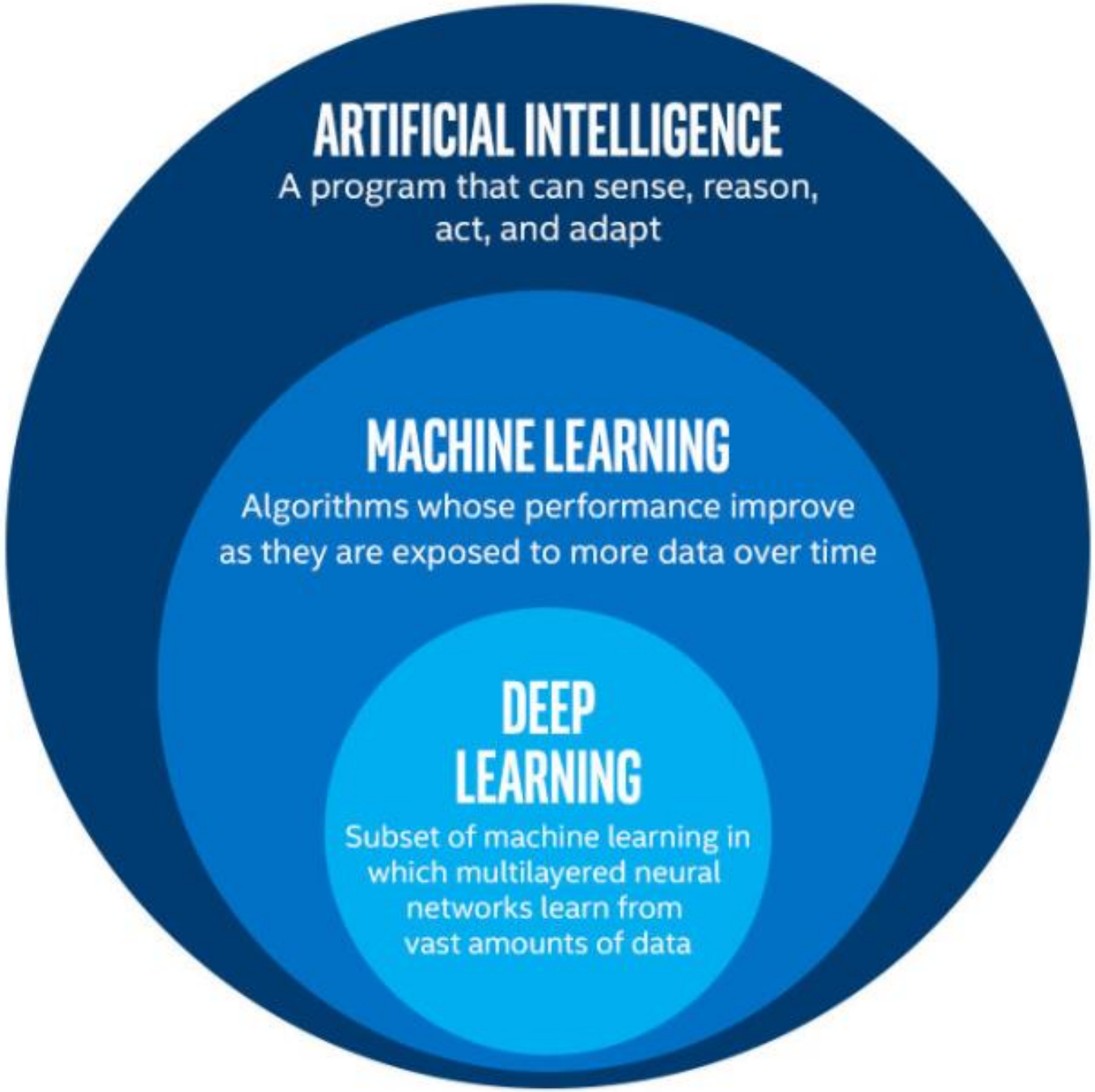
TRANSITION RADIATION DETECTOR


Stack
Multi-Chip-Module
6 Layer
TRD
18 Channels
24 padrows
Chamber
6 Chips (96 Channel)
5 subdiv. in z
18 subdivisions in ϕ


cathode pads
pion
electron
amplification region
anode wires
cathode wires
drift region
primary clusters
Drift Chamber
entrance window
Radiator
x
z
pion
TR photon
electron
cathode pads
electron
3
4
5
anode wires
cathode wires
amplification region
Drift Chamber
drift region
Signal
Time bin
Pad number
Radiator
x
y
electron


0
2
4
6
8
10
12
14
16
0
5
10
15
20

DEEP LEARNING


ARTIFICIAL INTELLIGENCE
A program that can sense, reason, act, and adapt
MACHINE LEARNING
Algorithms whose performance improve as they are exposed to more data over time
DEEP LEARNING
Subset of machine learning in which multilayered neural networks learn from vast amounts of data

DEEP FEEDFORWARD NETWORKS

- NESTED APPROXIMATING MAPPING FUNCTIONS
- MAPS x_i ; $i = \{1,2, \dots, n\} \rightarrow y$
- $f(x_{i,..,n}) = f_a^m(f_a^{\cdots}(f_a^2(f_a^1(x_{i,\dots,n}))))$
- EACH LAYER TAKES THE FORM: $w^T x + b$
- NON-LINEAR ACTIVATION FUNCTIONS $\phi(f_a(x))$ CAPTURES NON-LINEARITIES IN THE MULTIDIMENSIONAL FEATURE SPACE
- GOAL → TO ARRIVE AT A THEORETICALLY OPTIMAL MAPPING FUNCTION $f^*(x) \approx y$

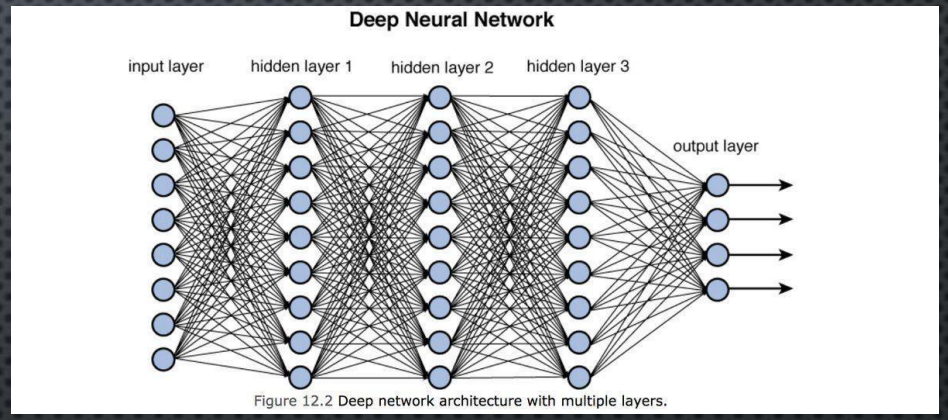
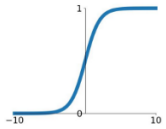


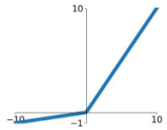
Figure 12.2 Deep network architecture with multiple layers.

Activation Functions

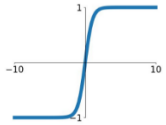
Sigmoid
$\sigma(x) = \frac{1}{1+e^{-x}}$



Leaky ReLU
$\max(0.1x, x)$

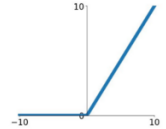


tanh
$\tanh(x)$

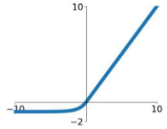


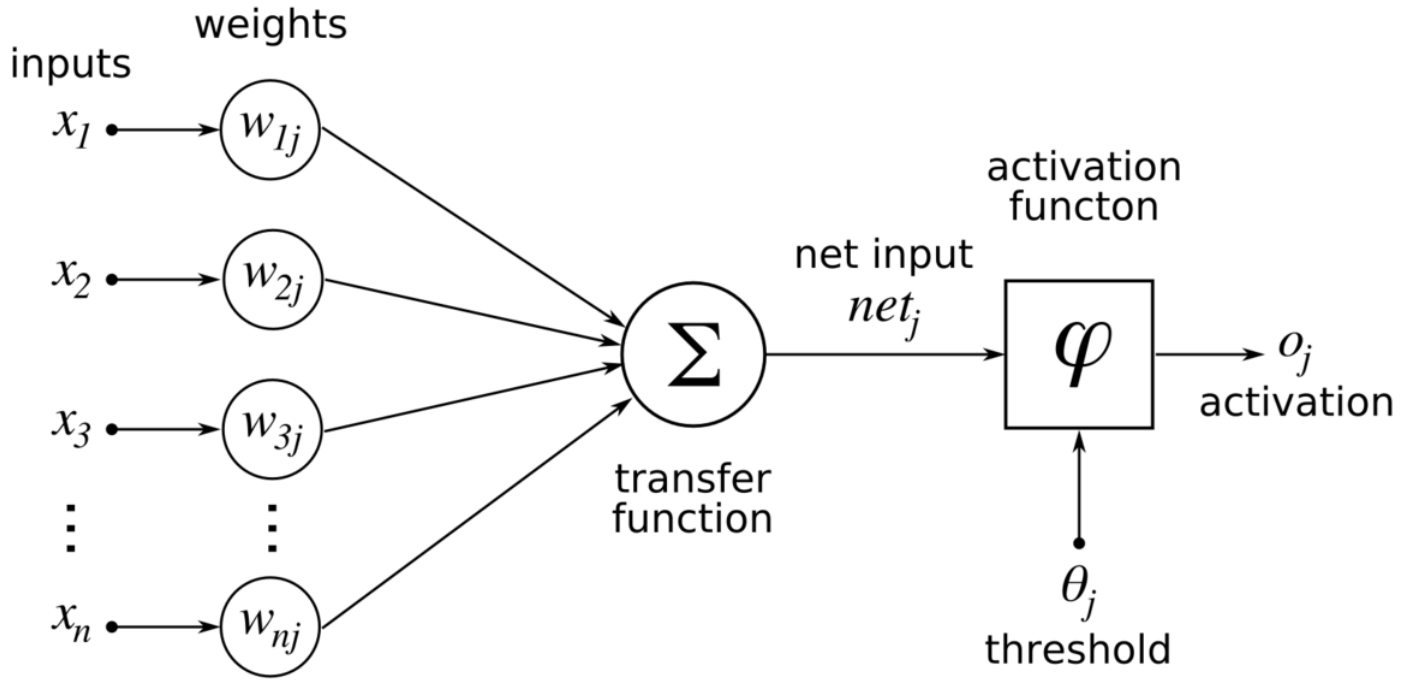
Maxout
$\max(w_1^T x + b_1, w_2^T x + b_2)$

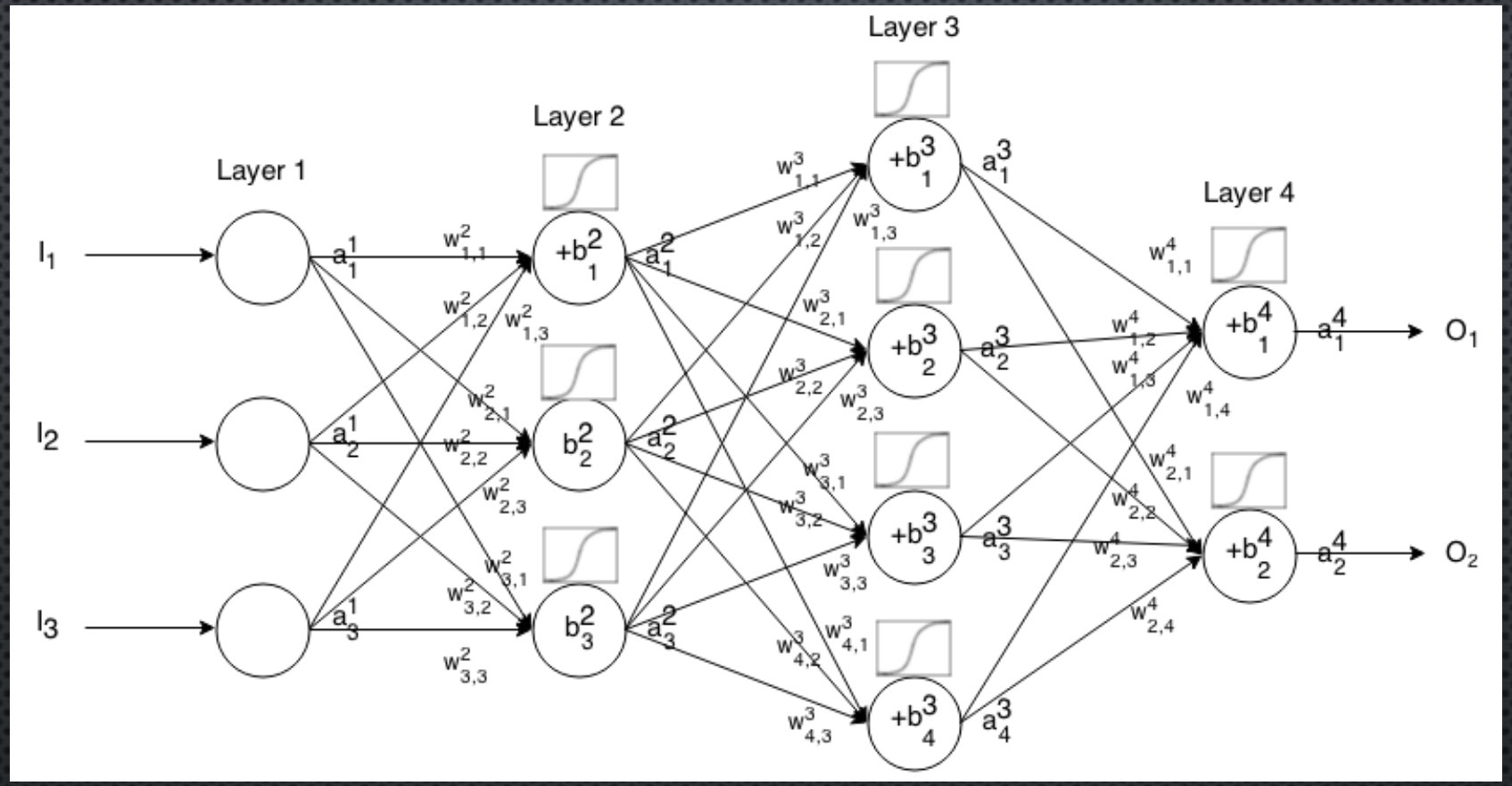
ReLU
$\max(0, x)$



ELU
$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$




weights
inputs
x_1
x_2
x_3
x_n
w_{1j}
w_{2j}
w_{3j}
w_{nj}
Σ
transfer
function
net input
net_j
activation
functon
φ
o_j
activation
θ_j
threshold



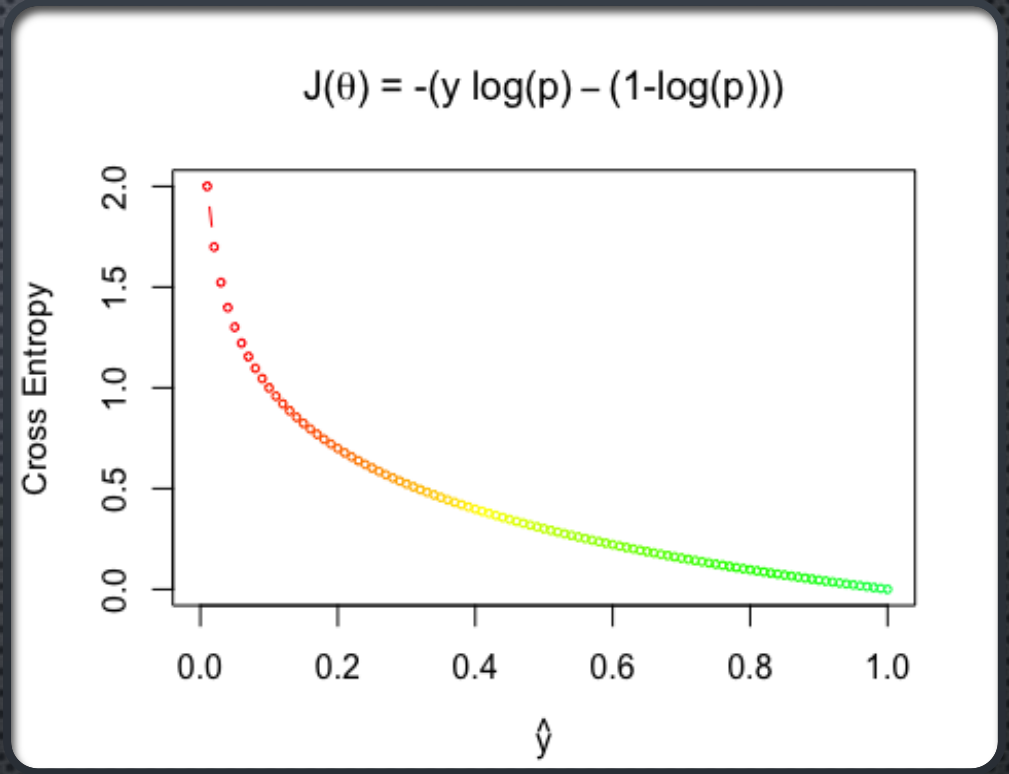
$$h^{(1)} = \phi^{(1)}(W^{(1)T}x + b^{(1)})$$

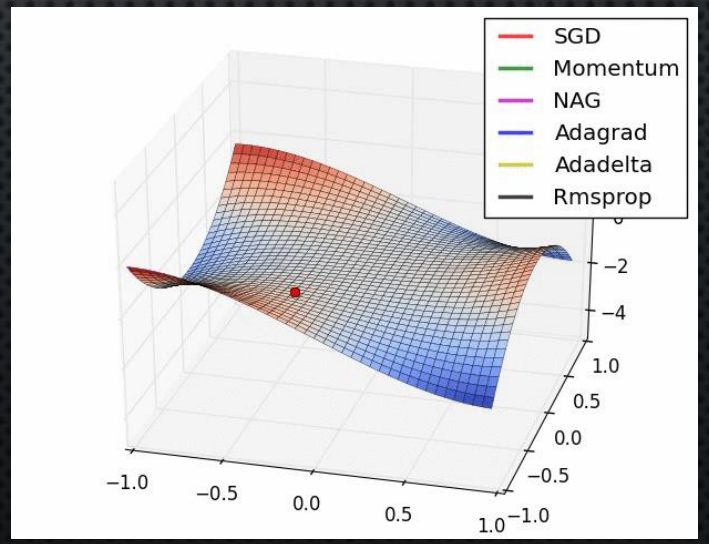
$$h^{(2)} = \phi^{(2)}(W^{(2)T}h^{(1)} + b^{(2)})$$

$$h^{(3)} = \phi^{(3)}(W^{(3)T}h^{(2)} + b^{(3)})$$

LOSS FUNCTION

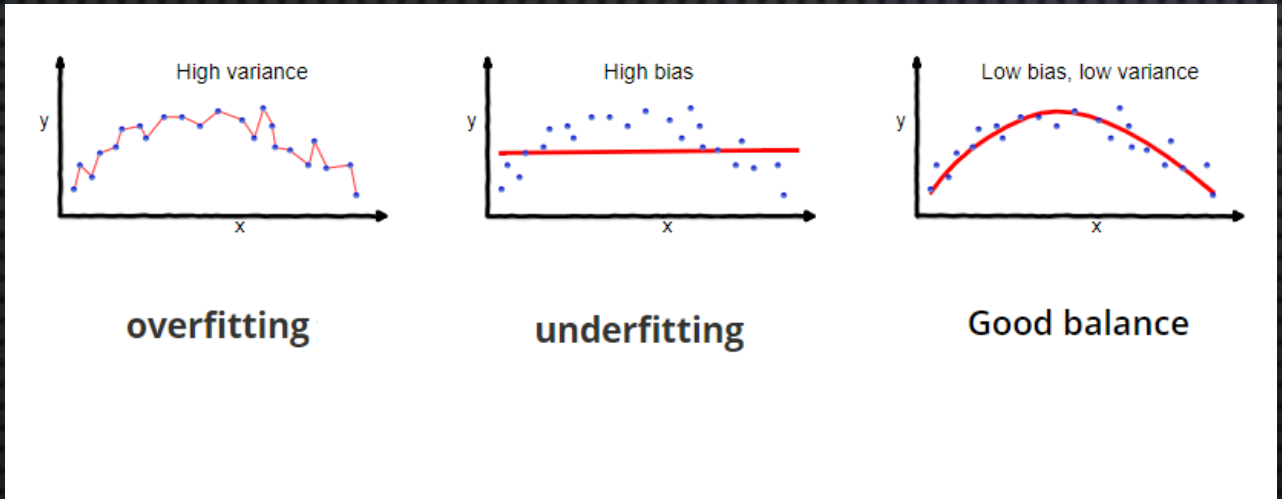
- INFORMS BACK-PROPAGATION
- PARAMETER SET θ OF EACH MAPPING FUNCTION $f_a^{j,\ldots,m}$ IS ADJUSTED ITERATIVELY ACCORDING TO ITS CONTRIBUTION TO THE DIFFERENTIAL OF THE LOSS FUNCTION AT EACH TRAINING STEP k
- $\frac{\partial J_k}{\partial f_a^j}$





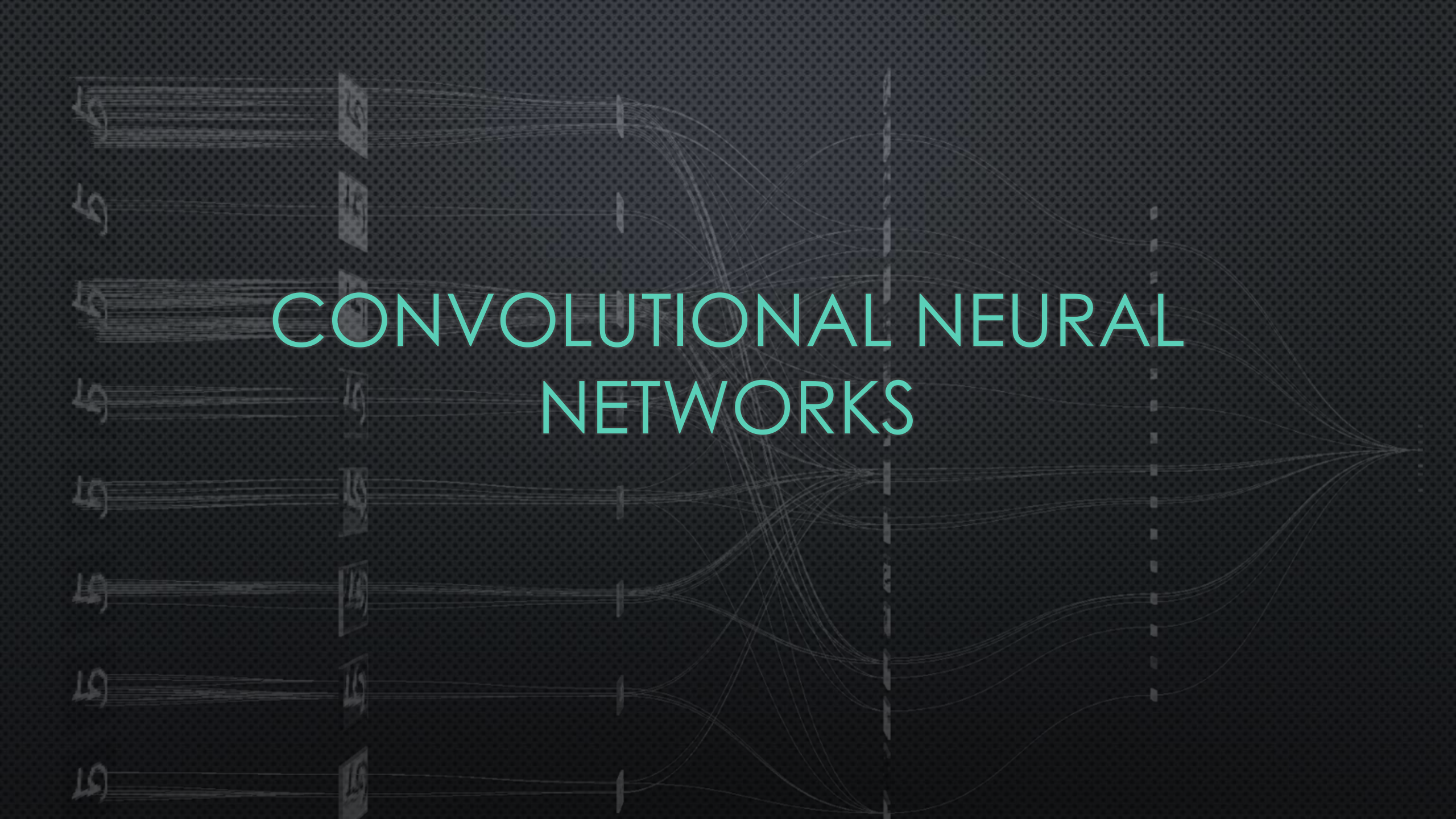
NOT **TOO** PERFECT

BIAS-VARIANCE TRADE-OFF



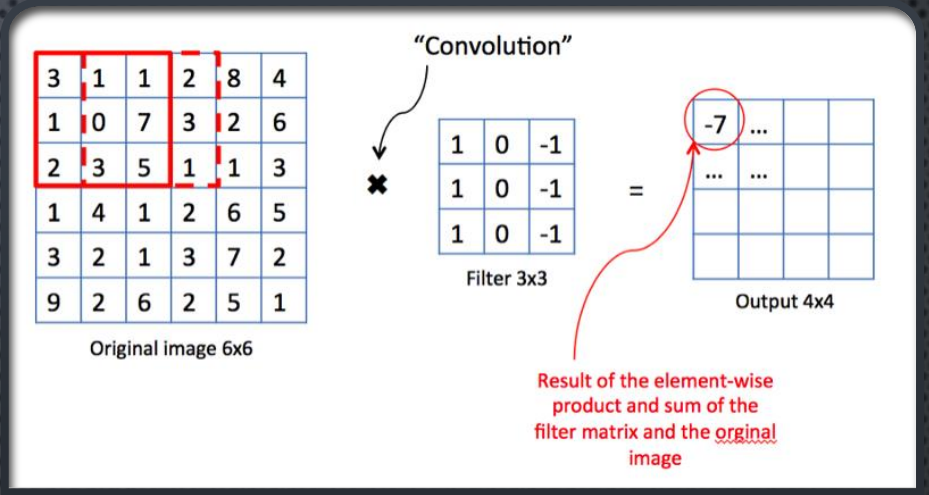
REGULARIZATION

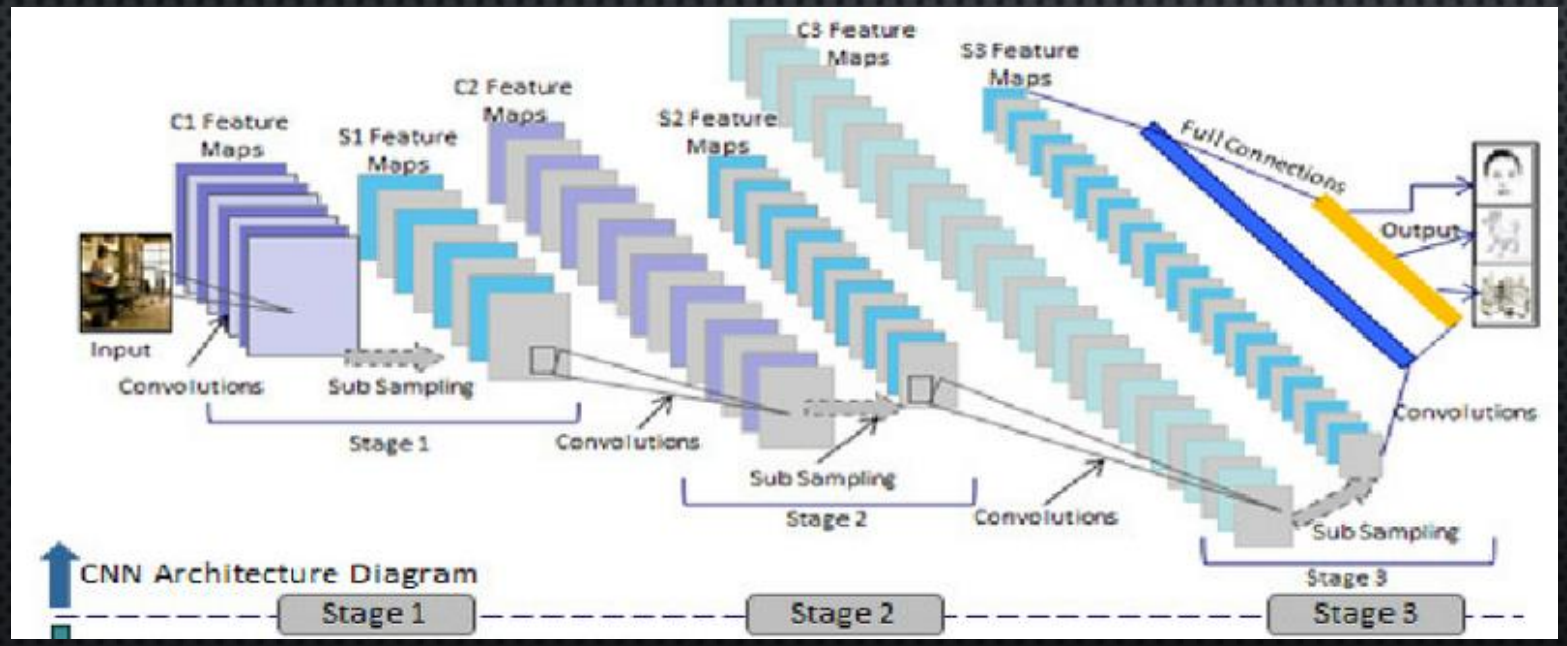
- L2-NORM PENALTY
- DROP-OUT
- LIMITING CAPACITY OF NETWORK
- GAUSSIAN NOISE LAYERS
- EARLY STOPPING

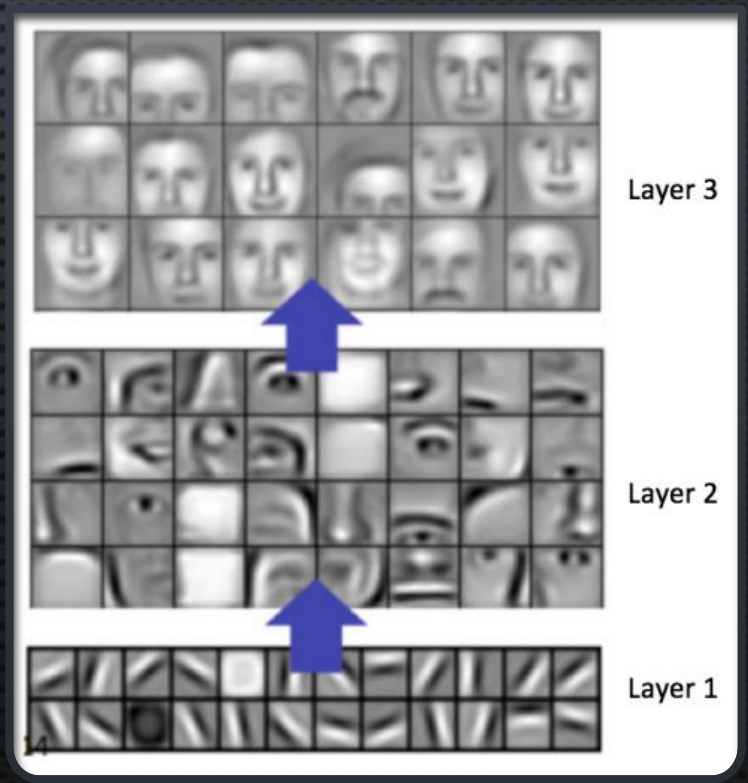


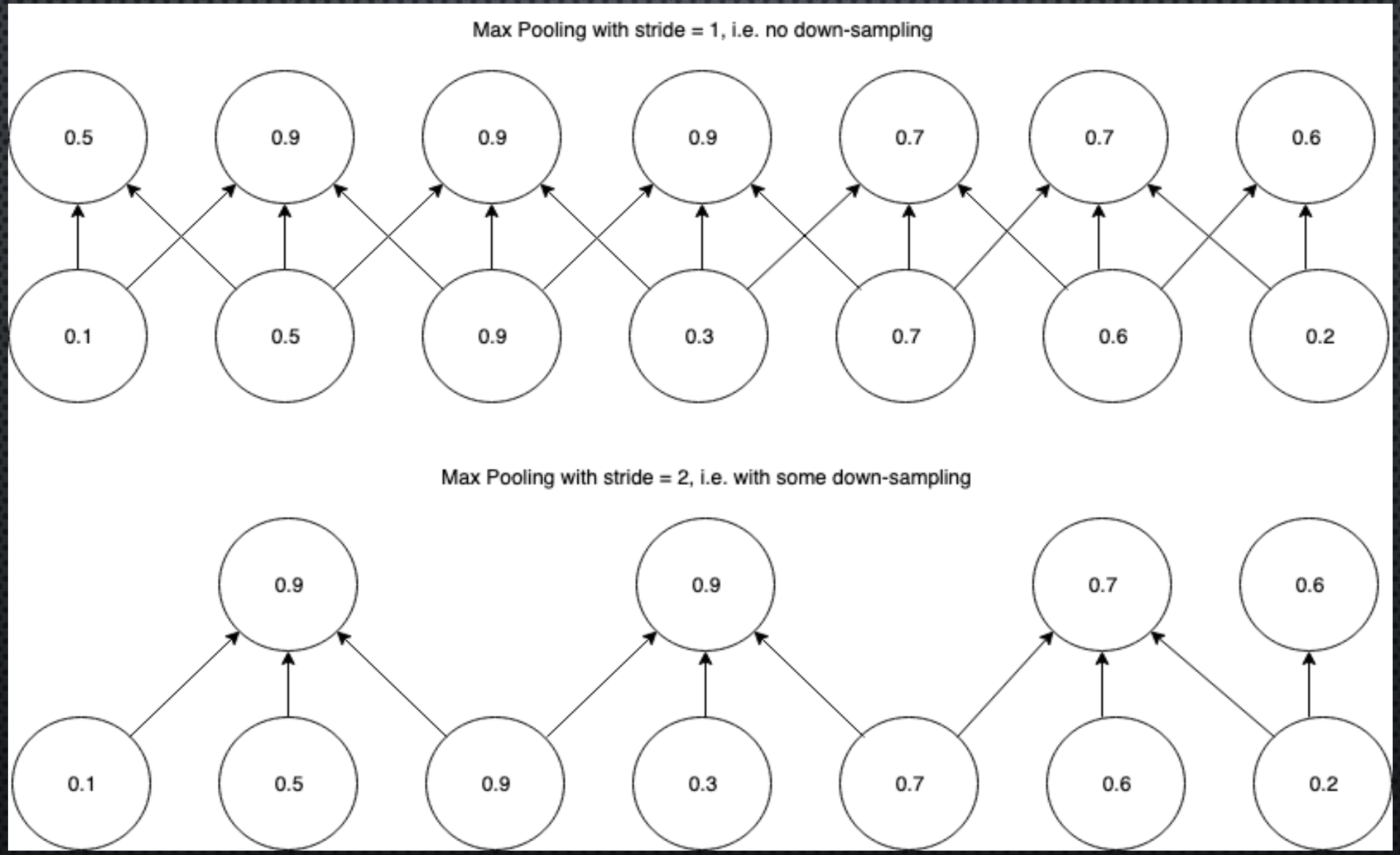
CONVOLUTIONAL NEURAL NETWORKS

CONVOLUTIONAL FILTERS







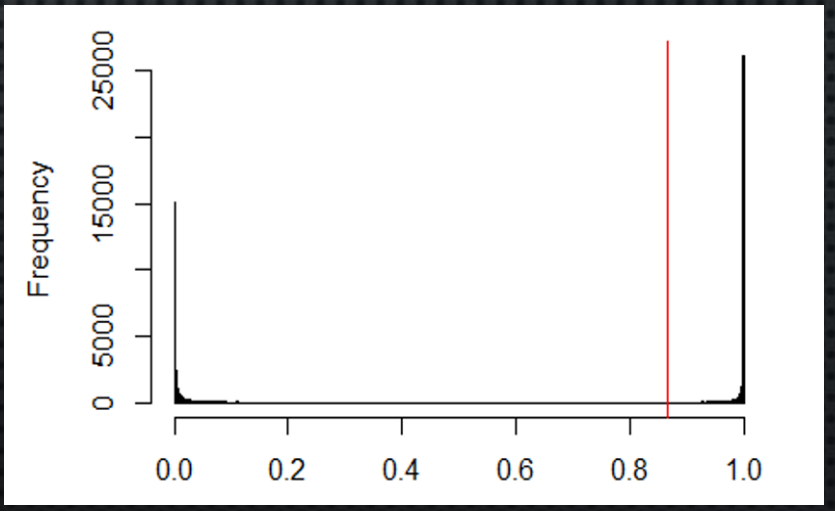

Max Pooling with stride = 1, i.e. no down-sampling
0.5
0.9
0.9
0.9
0.7
0.7
0.6
0.1
0.5
0.9
0.3
0.7
0.6
0.2
Max Pooling with stride = 2, i.e. with some down-sampling
0.9
0.9
0.7
0.6
0.1
0.5
0.9
0.3
0.7
0.6
0.2

PARTICLE IDENTIFICATION RESULTS

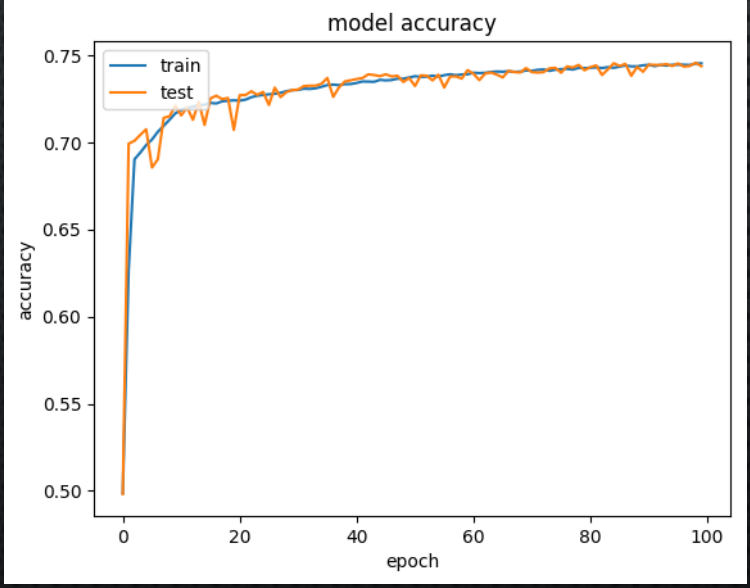
- PION EFFICIENCY $\varepsilon_\pi = 2.2\%$ AT ELECTRON EFFICIENCY $\varepsilon_{e^-} = 90\%$

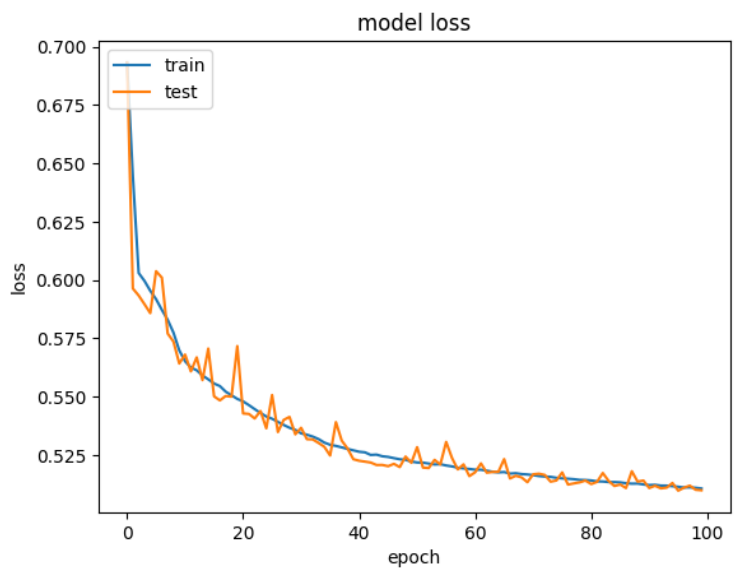
$$P(elec) = \frac{\prod_{j=1}^{6} P_j(elec)}{\sum_{k \in e,\pi} \prod_{j=1}^{6} P_j(k)}$$

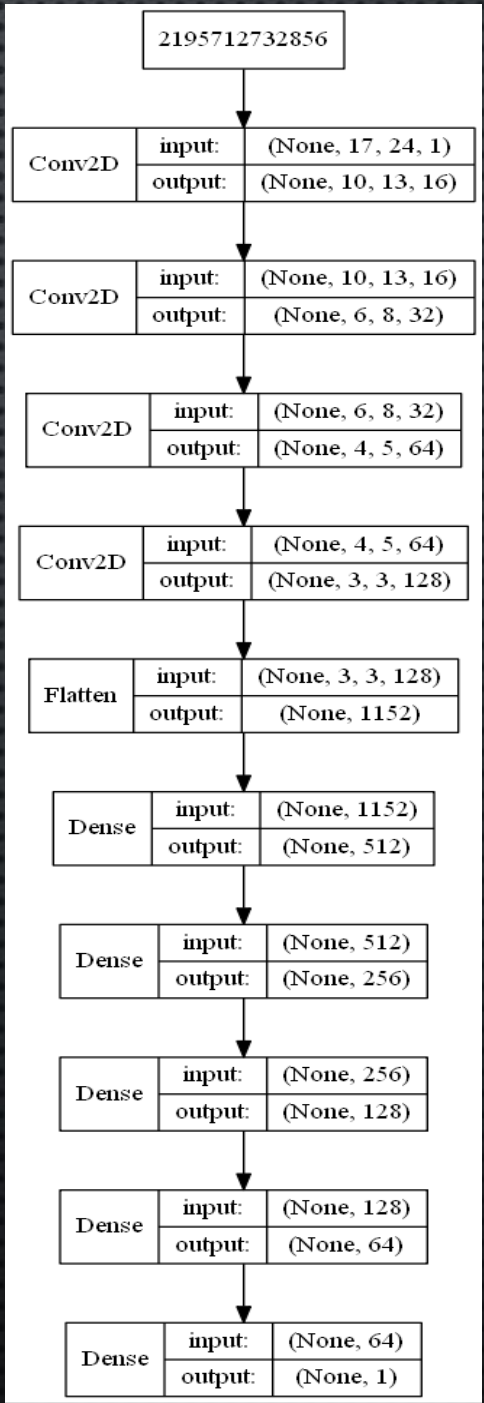
$$\varepsilon_e = \int_{-\infty}^{t_{cut}} g(t|e)dt$$



ARCHITECTURE AND TRAINING CURVES





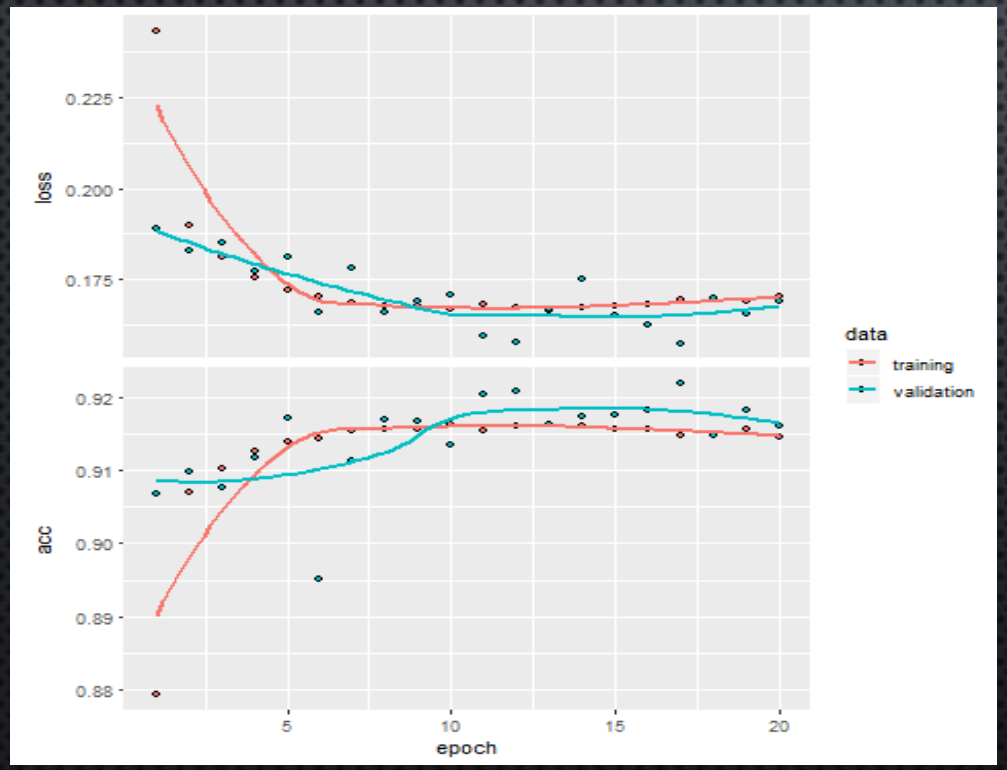


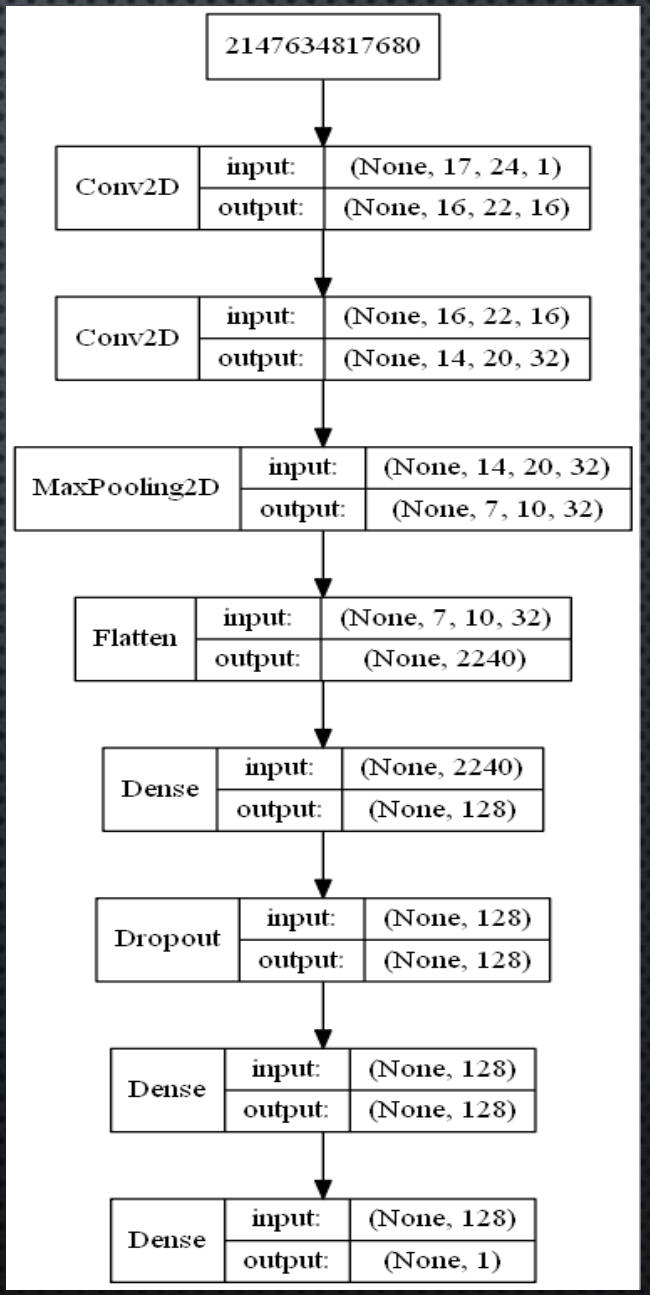
DISTINGUISHING GEANT FROM REAL DATA RESULTS

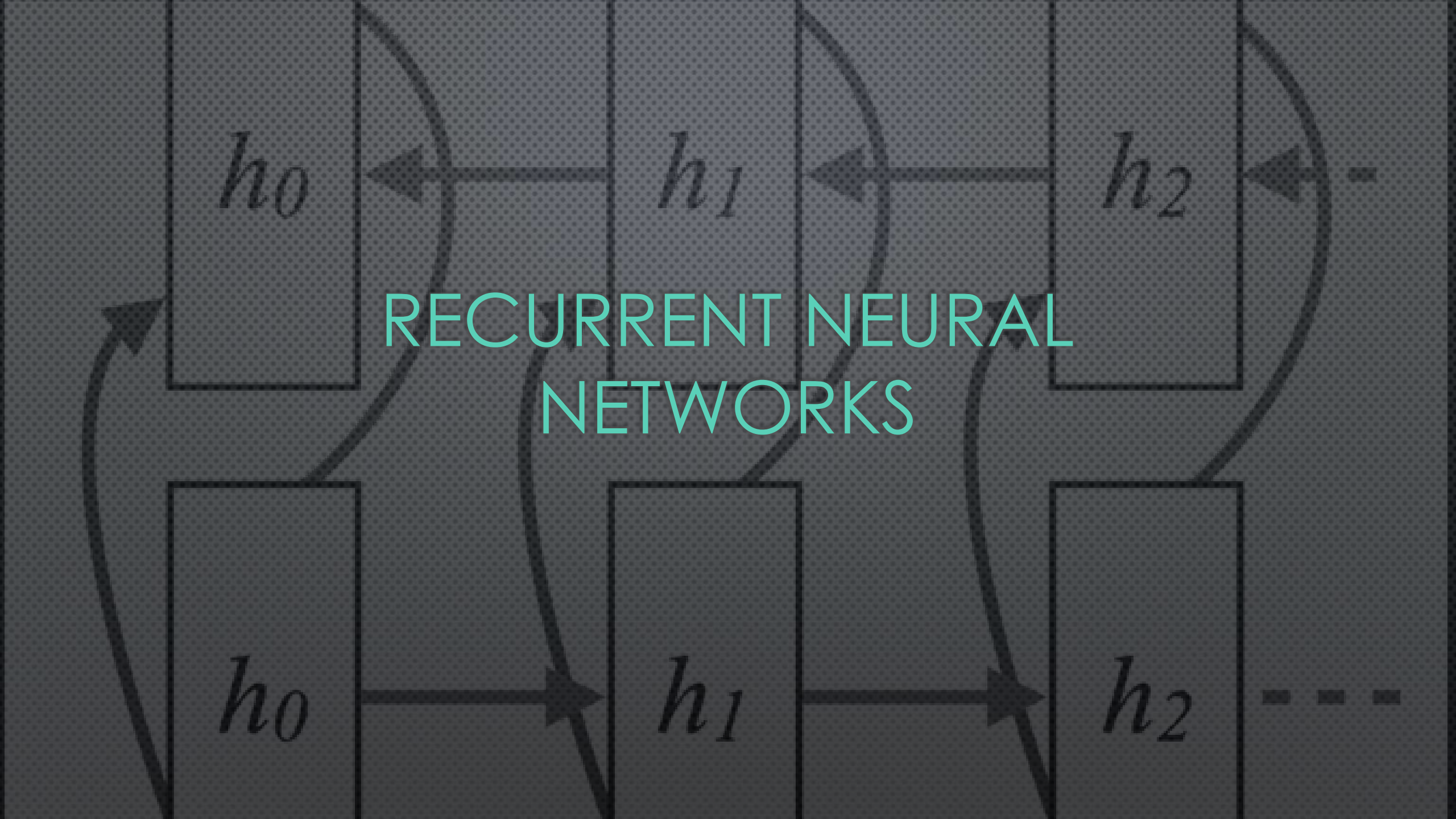
BALANCED ACCURACY = 91.05 %

Prediction/Actual	0	1
0	42 553	681
1	7 069	24 058

ARCHITECTURE AND TRAINING CURVES



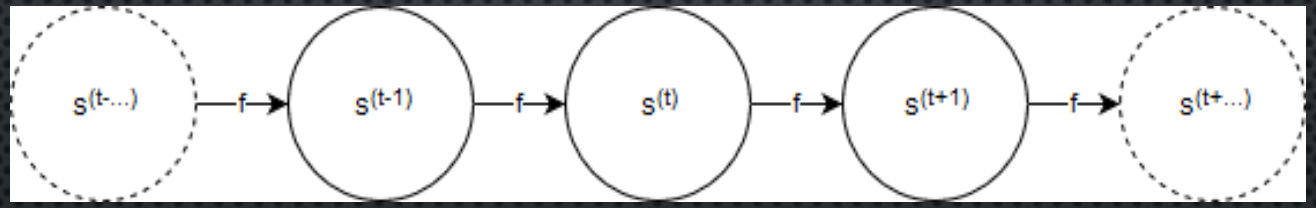




RECURRENT NEURAL NETWORKS

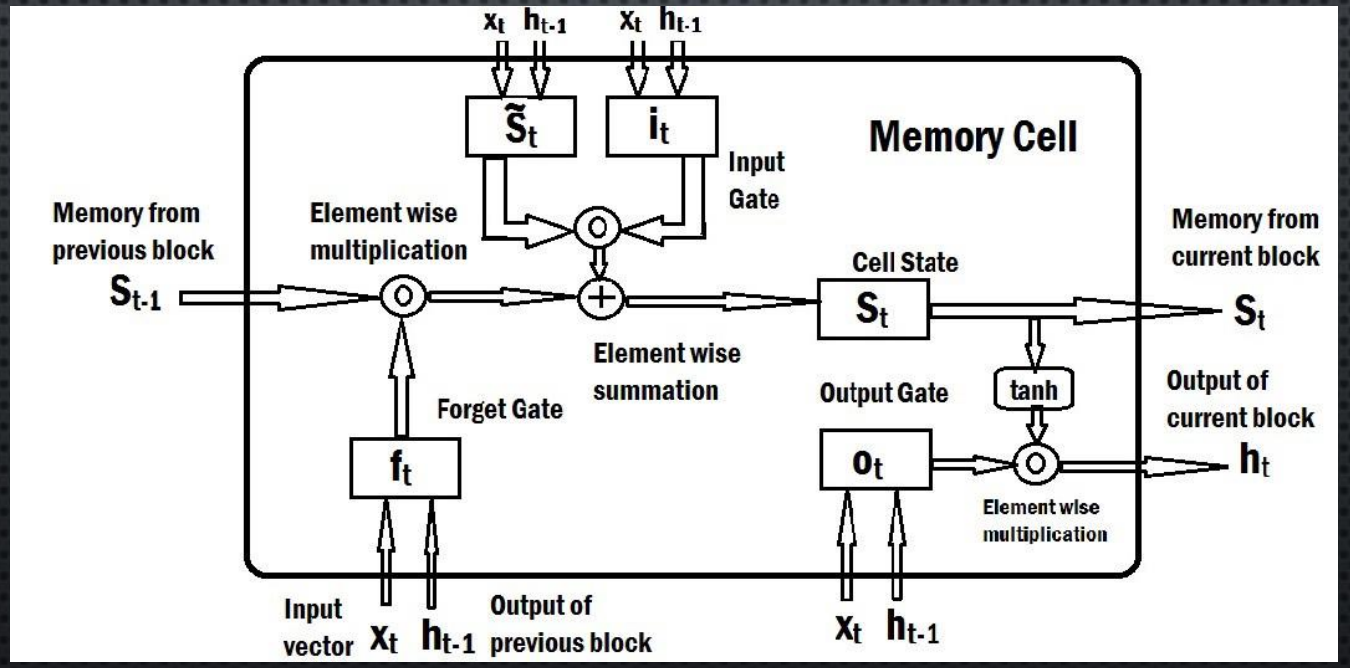
RECURRENT NEURAL NETWORKS

$$s^{(t)} = f(s^{(t-1)}; \theta)$$

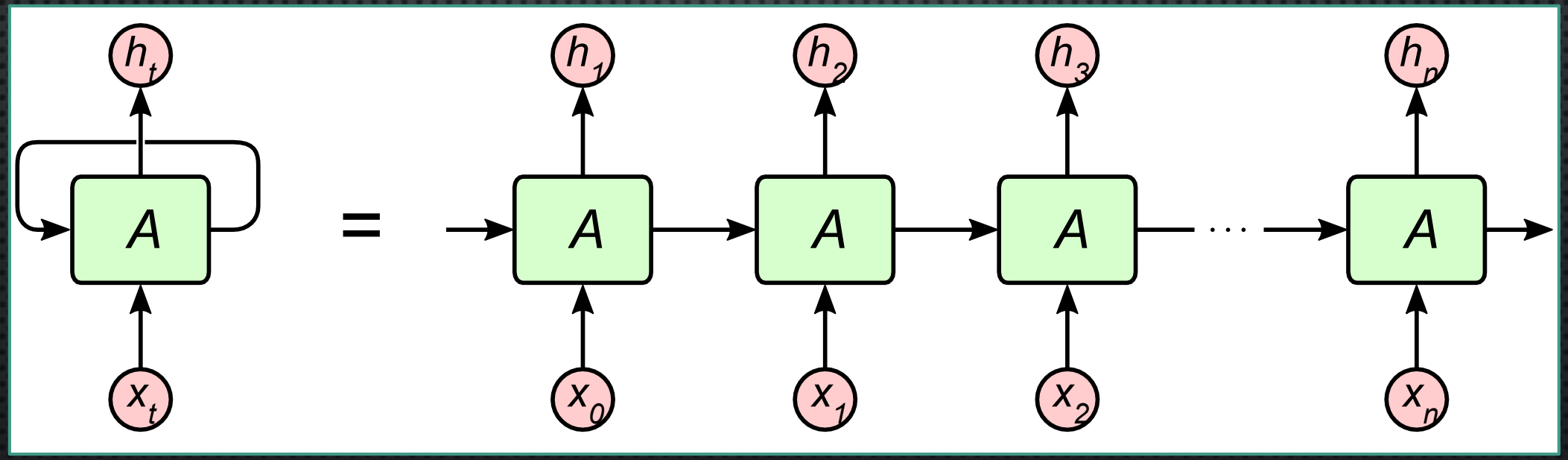


$$h^{(t)} = f(h^{(t-1)}, x^{(t)}; \theta)$$

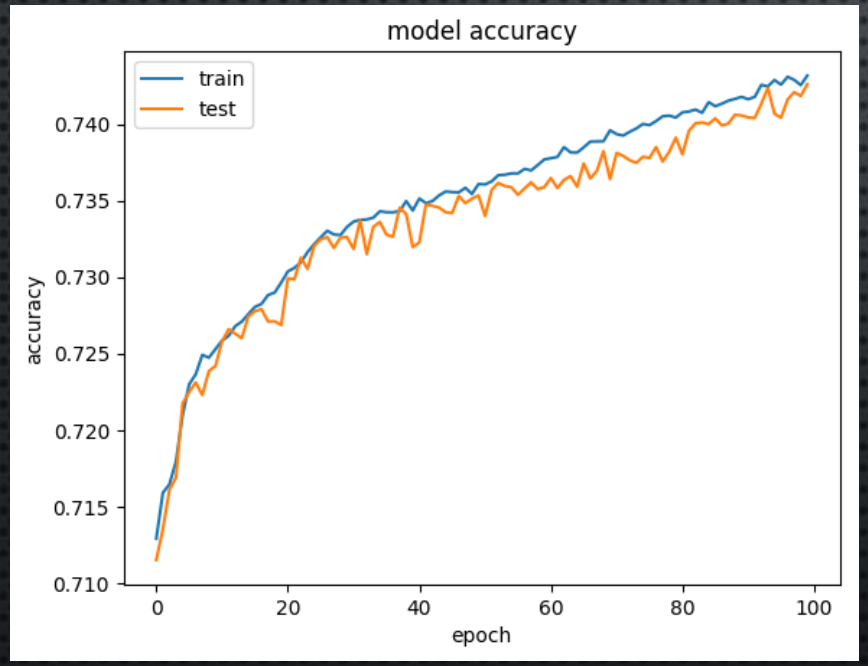
LSTM CELL


x_t h_{t-1} x_t h_{t-1}
$\tilde{S}_t$
i_t
Input Gate
Memory Cell
Memory from previous block
S_{t-1}
Element wise multiplication
Cell State
S_t
Memory from current block
S_t
Element wise summation
Output Gate
tanh
Output of current block
h_t
Forget Gate
f_t
o_t
Element wise multiplication
Input vector x_t
h_{t-1} Output of previous block
x_t h_{t-1}

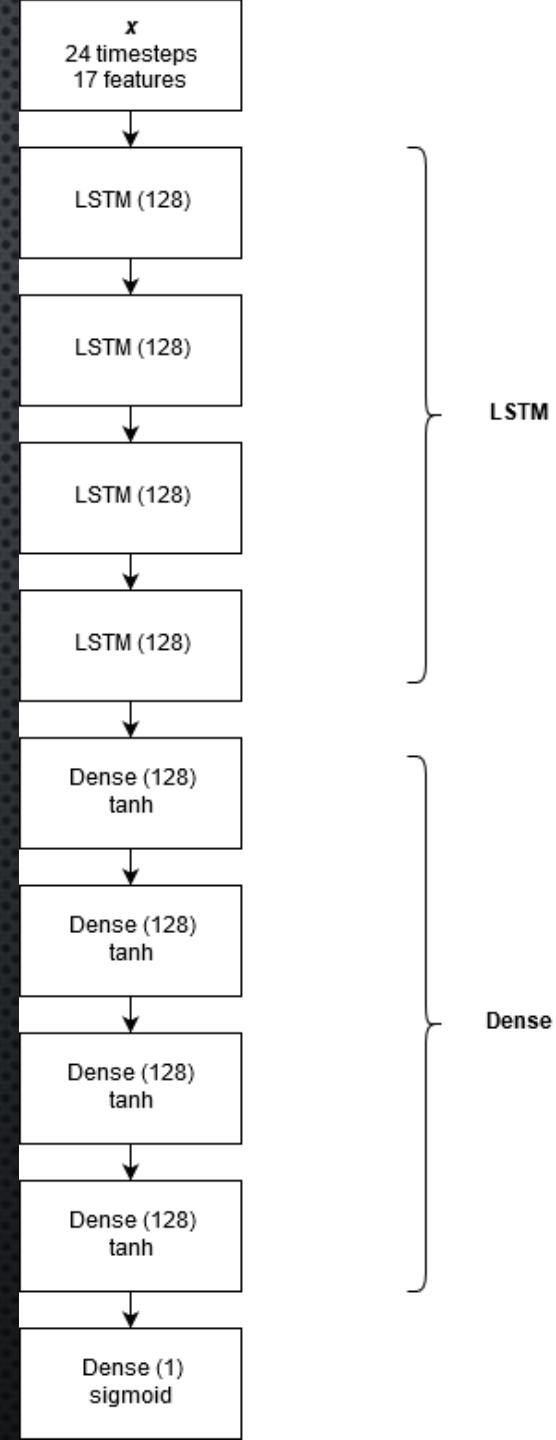
CHAIN OF LSTM CELLS

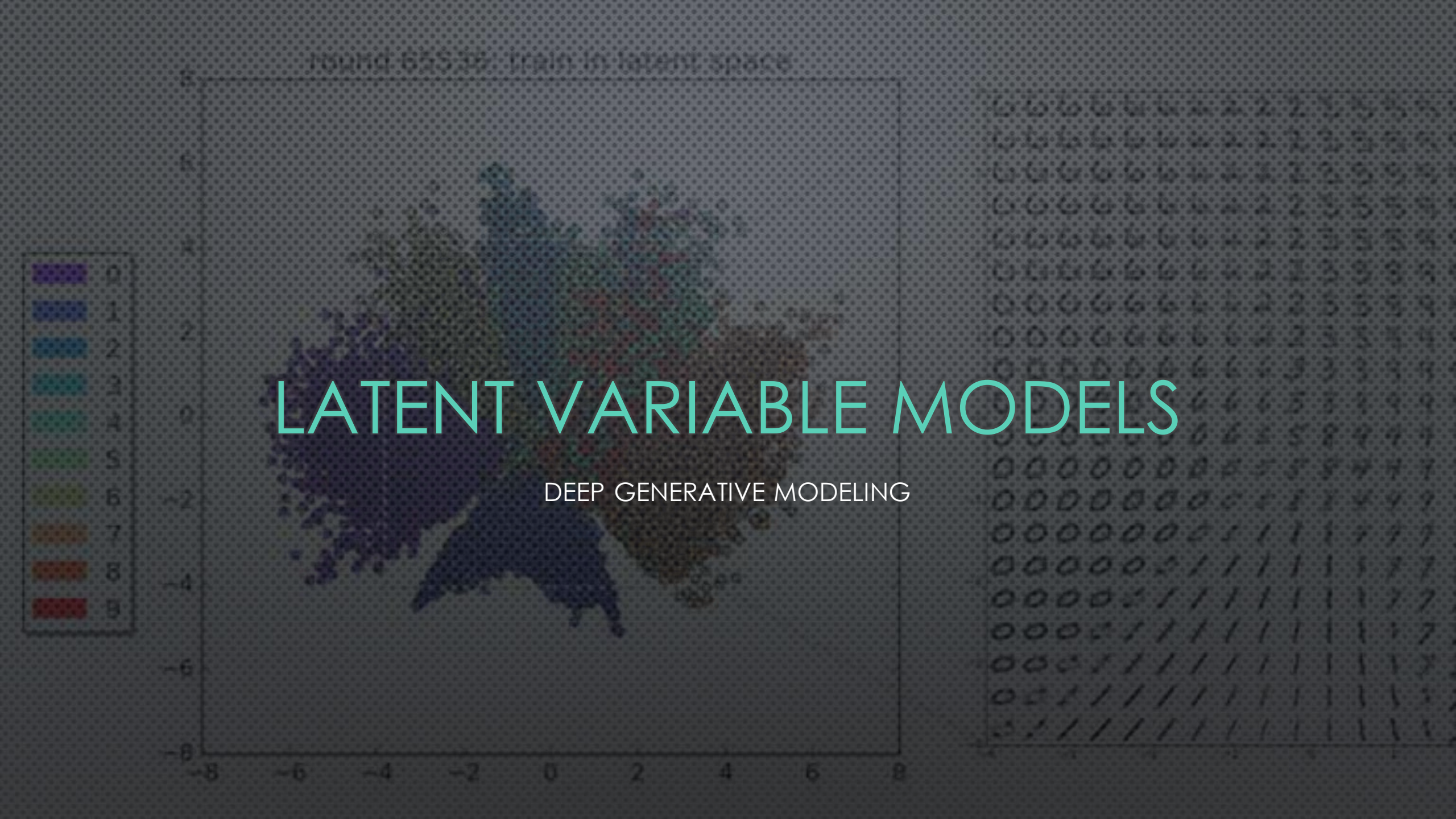


PARTICLE IDENTIFICATION RESULTS: LSTM



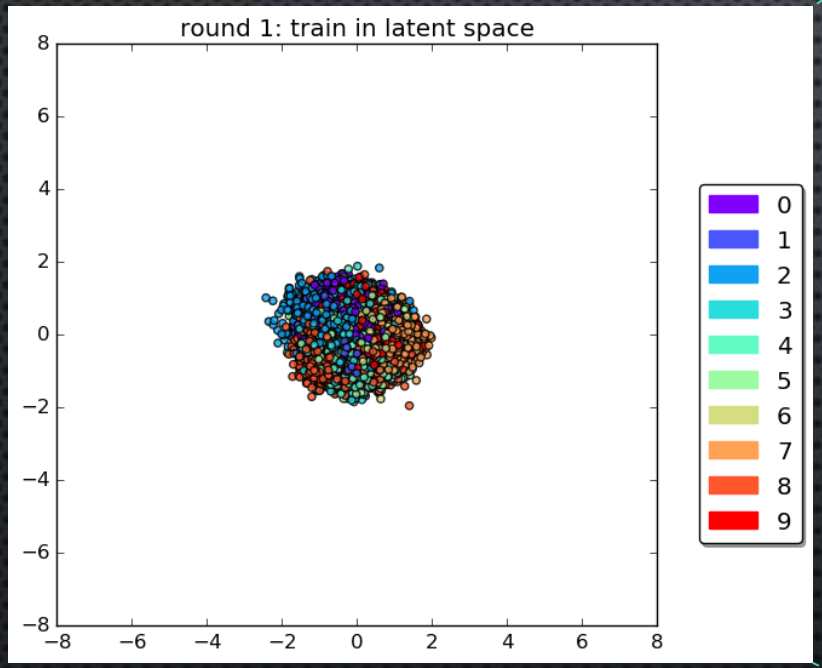
- $\varepsilon_\pi = 8.5\%$ AT $\varepsilon_{e^-} = 90\%$

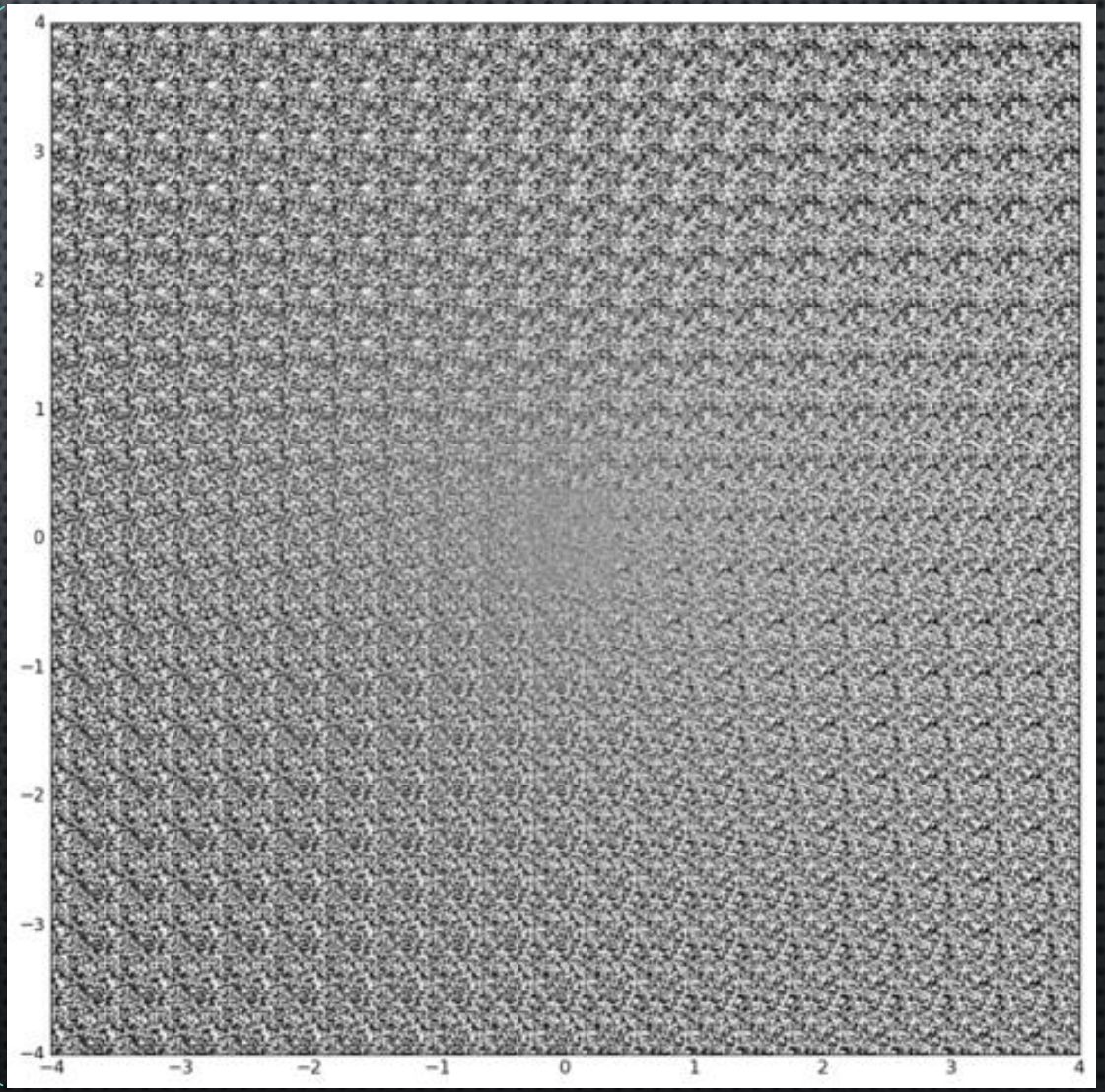




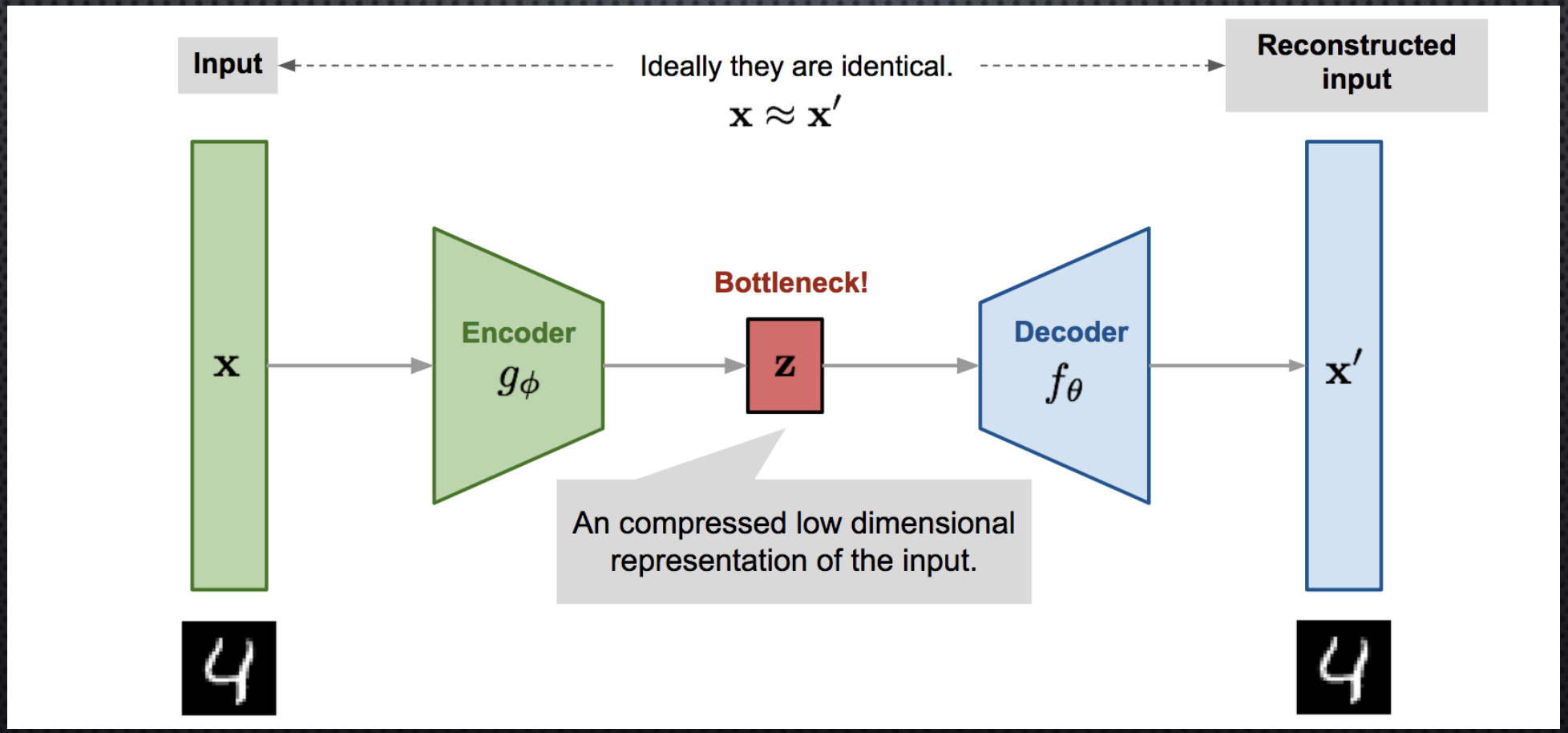
LATENT VARIABLE MODELS

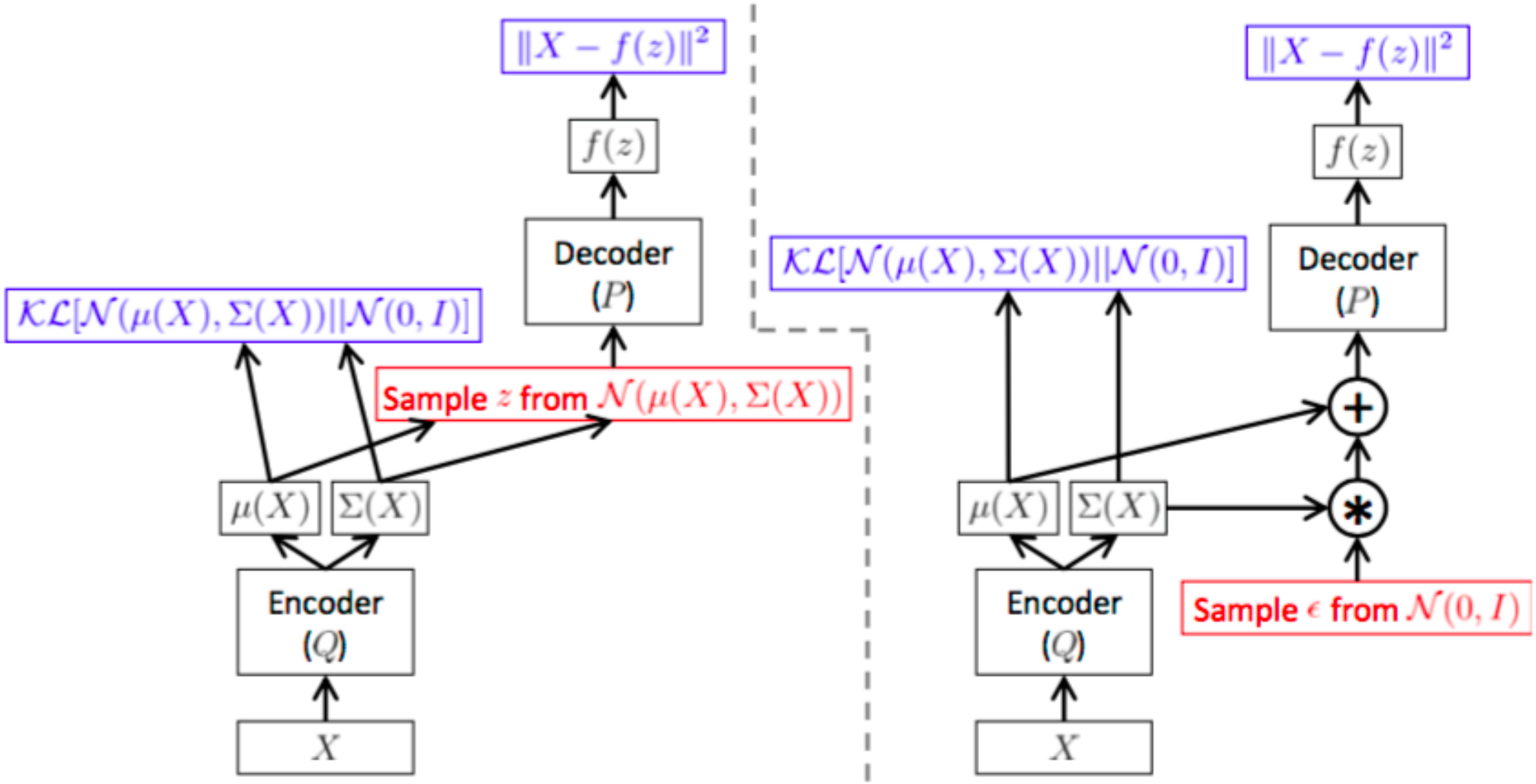
DEEP GENERATIVE MODELING


round 1: train in latent space
8
6
4
2
0
−2
−4
−6
−8
−8
−6
−4
−2
0
2
4
6
8
0
1
2
3
4
5
6
7
8
9


4
3
2
1
0
−1
−2
−3
−4
−4
−3
−2
−1
0
1
2
3
4

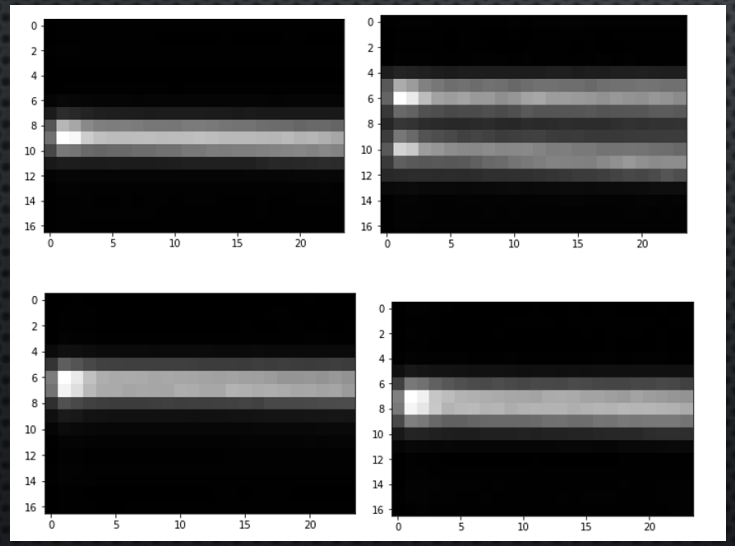
VARIATIONAL AUTOENCODERS



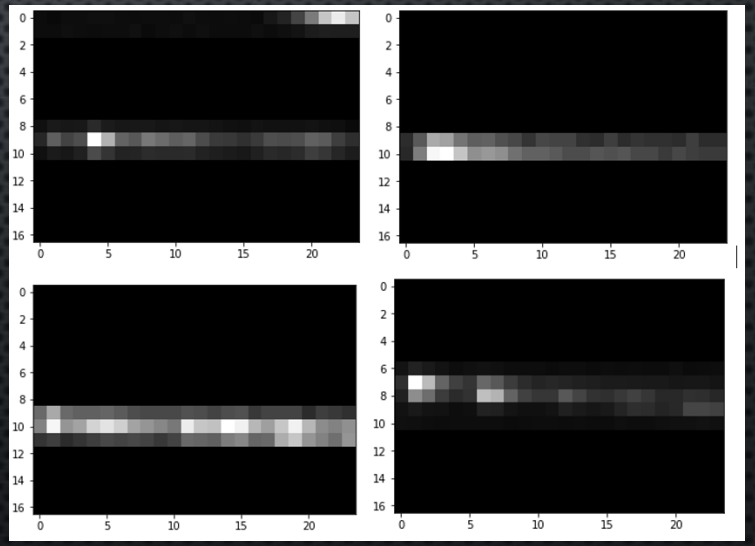

$\|X - f(z)\|^2$
$f(z)$
Decoder
(P)
$\mathcal{KL}[\mathcal{N}(\mu(X), \Sigma(X))||\mathcal{N}(0, I)]$
Sample z from $\mathcal{N}(\mu(X), \Sigma(X))$
$\mu(X)$
$\Sigma(X)$
Encoder
(Q)
X
$\|X - f(z)\|^2$
$f(z)$
Decoder
(P)
$\mathcal{KL}[\mathcal{N}(\mu(X), \Sigma(X))||\mathcal{N}(0, I)]$
+
*
$\mu(X)$
$\Sigma(X)$
Encoder
(Q)
Sample ϵ from $\mathcal{N}(0, I)$
X

VAE RESULTS

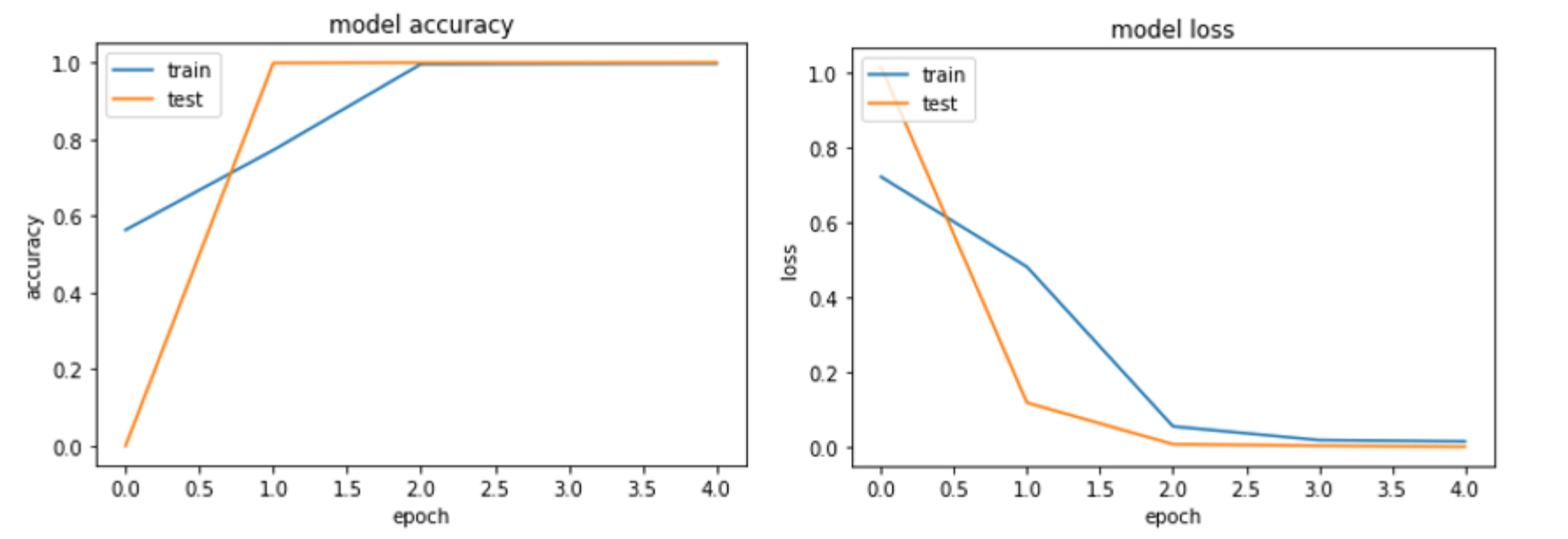
Variational Autoencoder



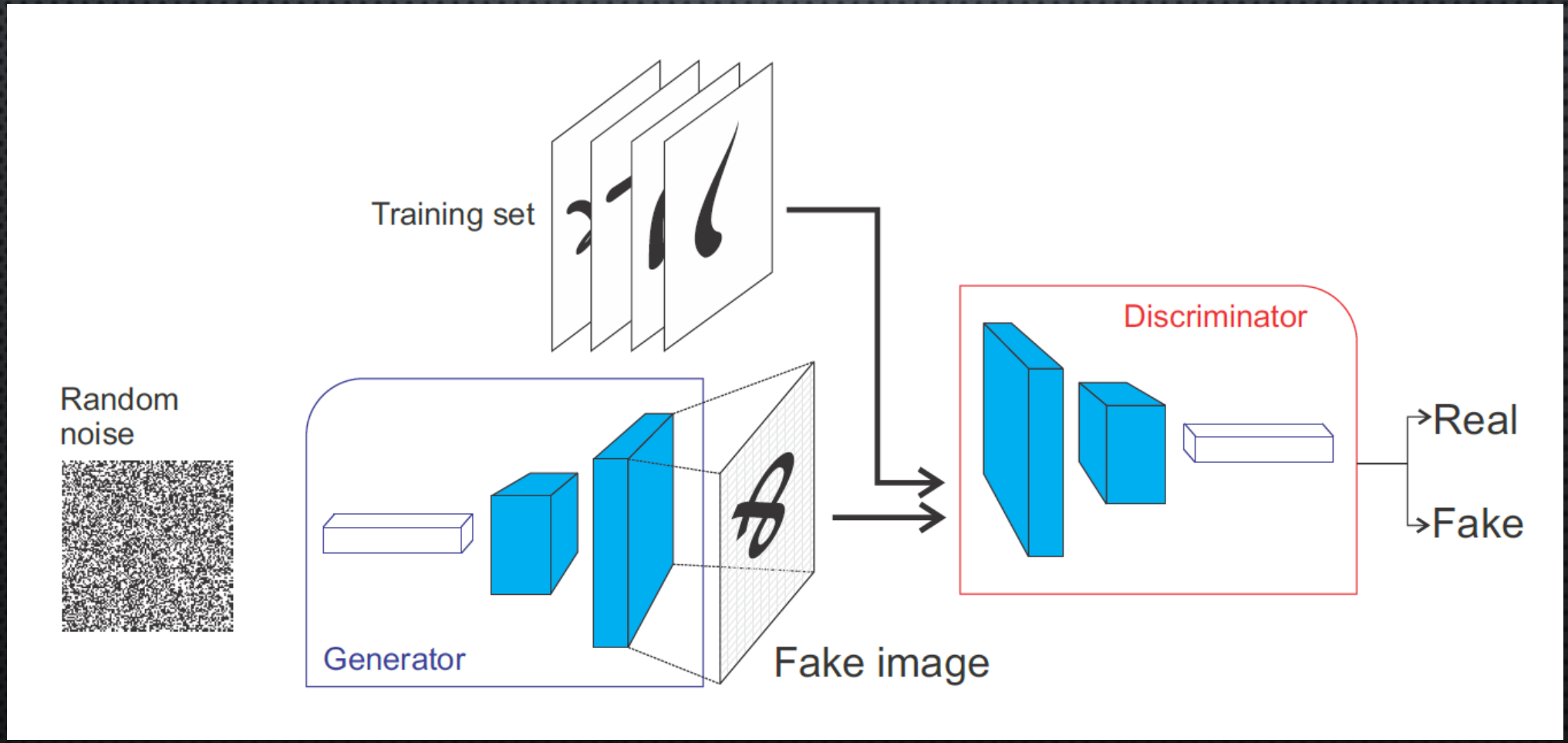
Real Data



DISTINGUISHING VARIATIONAL AUTOENCODER DATA FROM REAL DATA

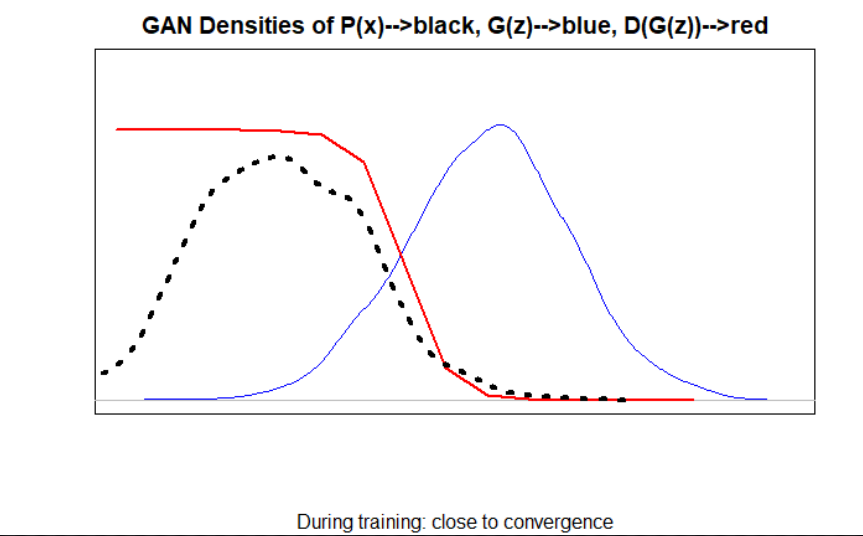


GENERATIVE ADVERSARIAL NETWORKS



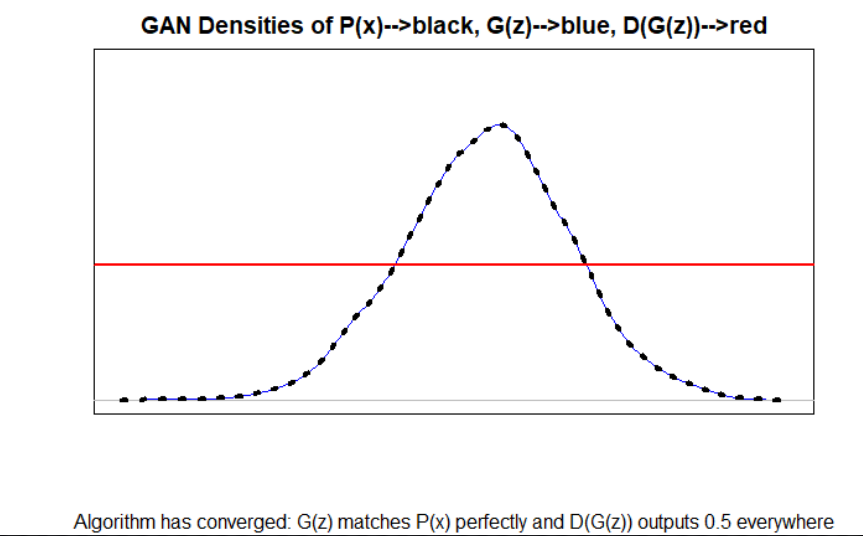
GAN DENSITIES DURING TRAINING

CLOSE TO CONVERGENCE



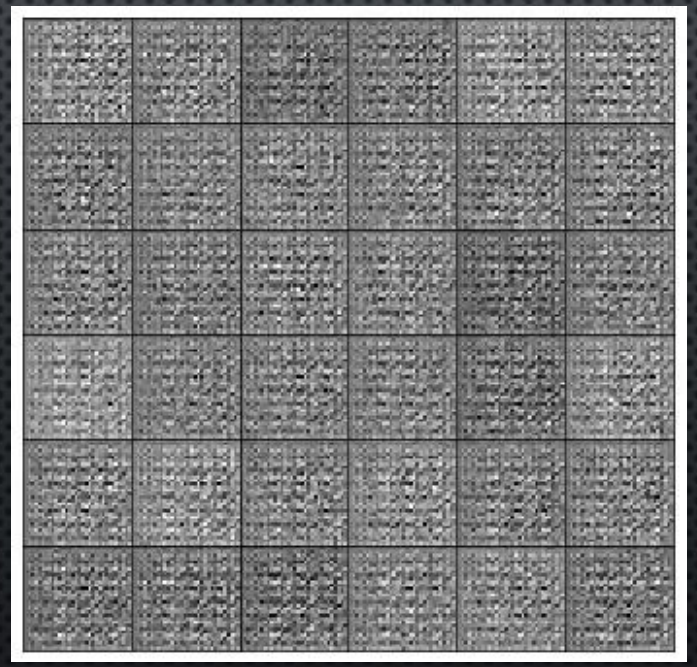
During training: close to convergence

ALGORITHM HAS CONVERGED

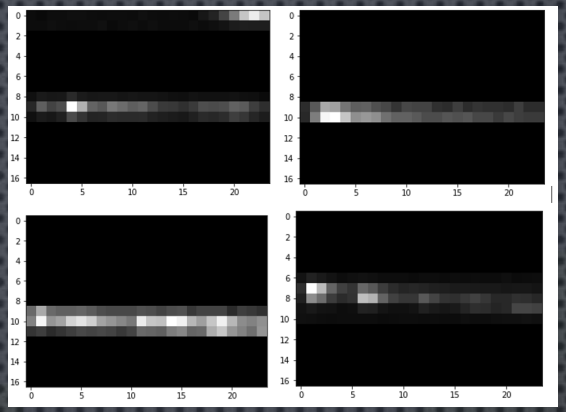


Algorithm has converged: G(z) matches P(x) perfectly and D(G(z)) outputs 0.5 everywhere

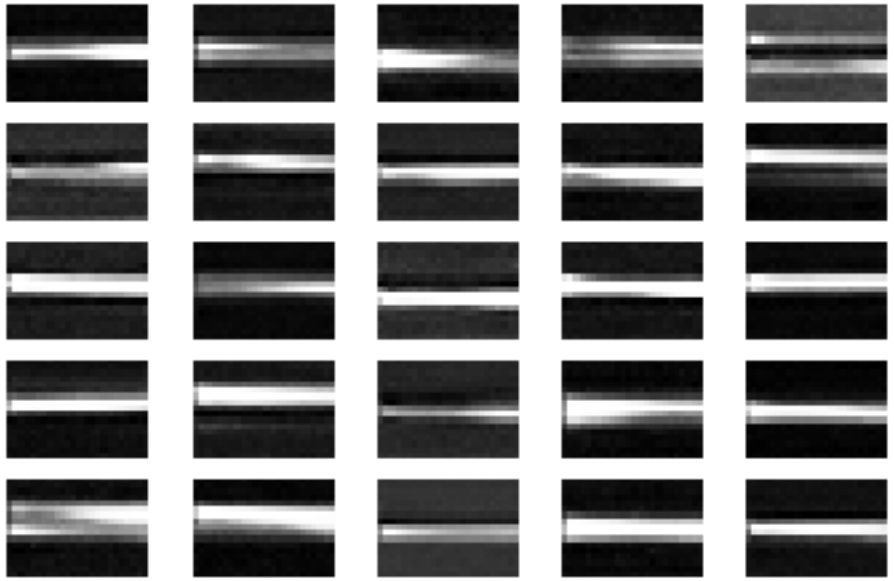
GAN TRAINING ON MNIST



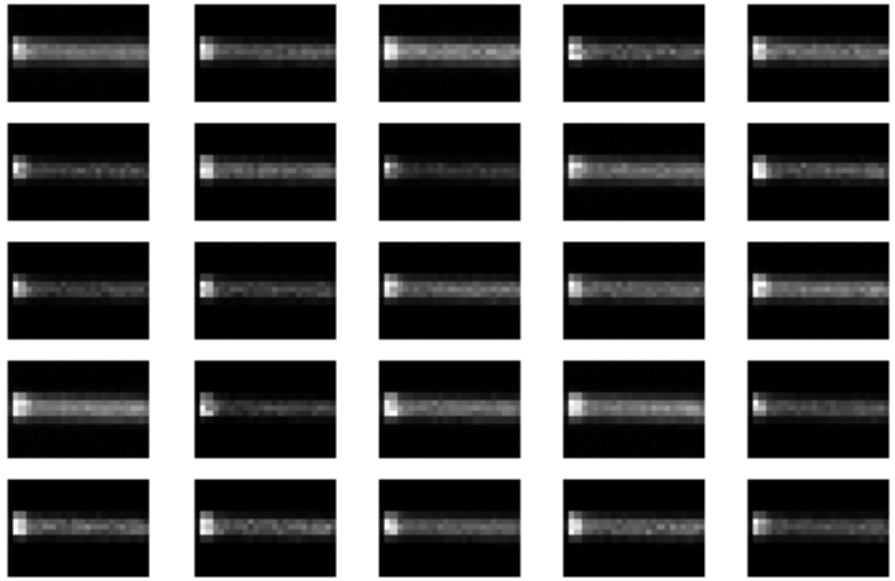
GAN RESULTS



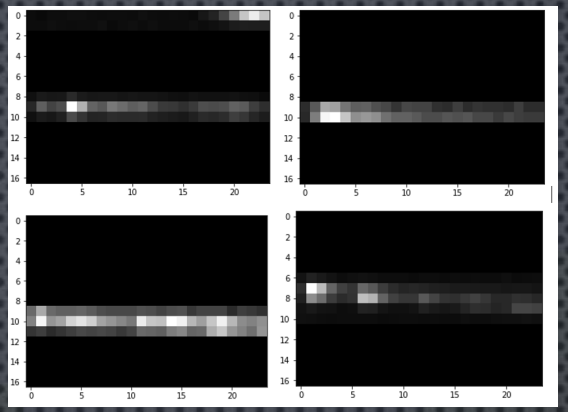
ADVERSARIAL AUTOENCODER 1



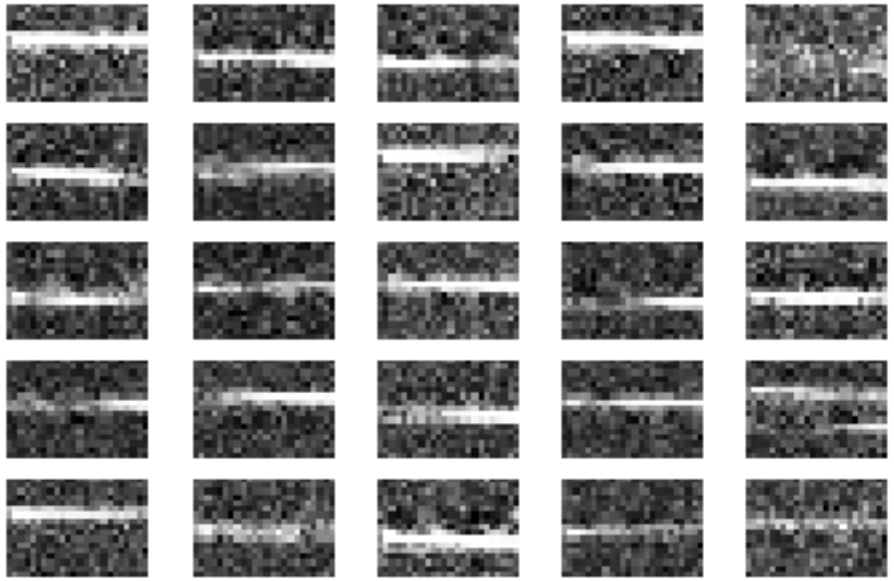
ADVERSARIAL AUTOENCODER 2



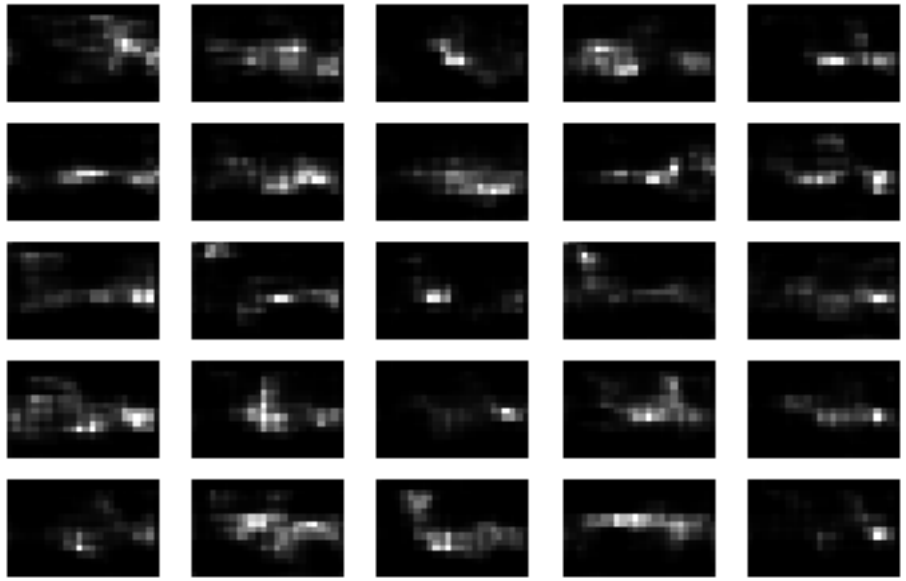
GAN RESULTS



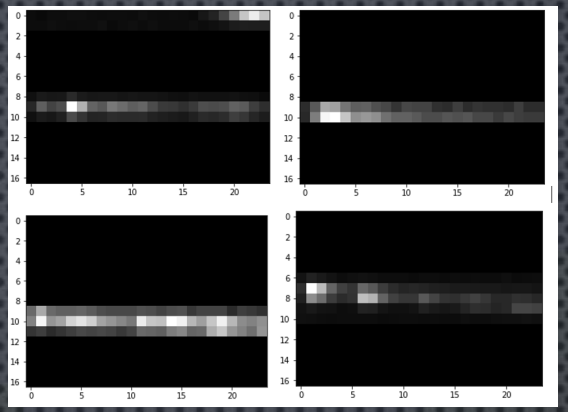
BIDIRECTIONAL GAN



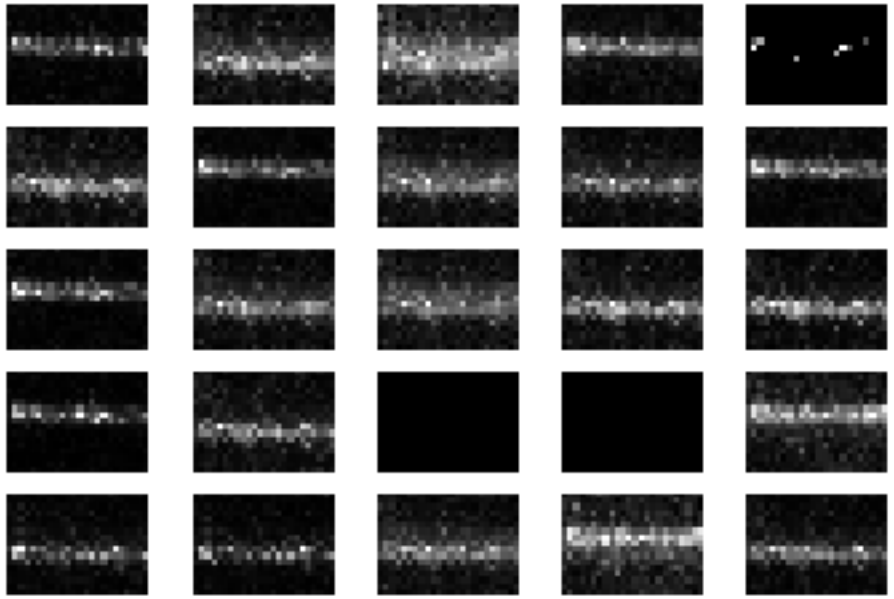
DEEP CONVOLUTIONAL GAN



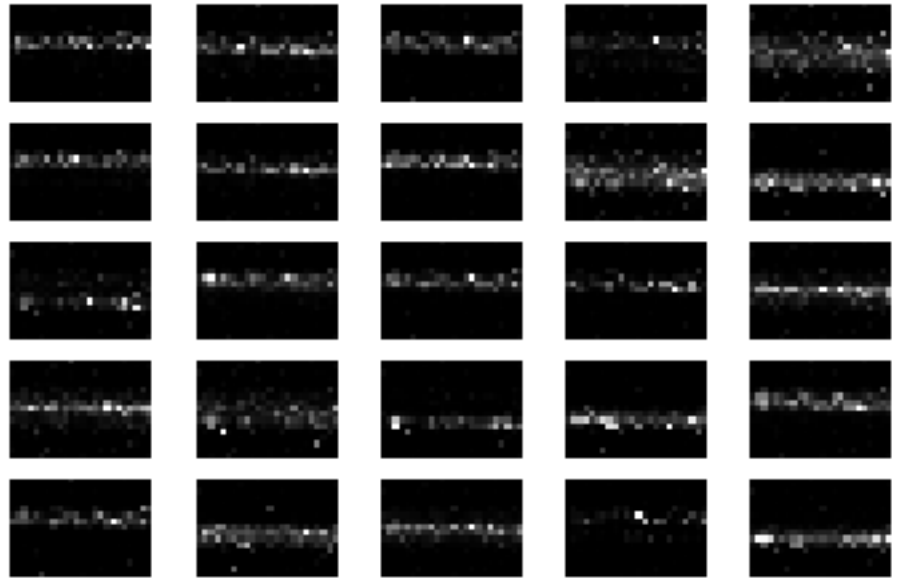
GAN RESULTS

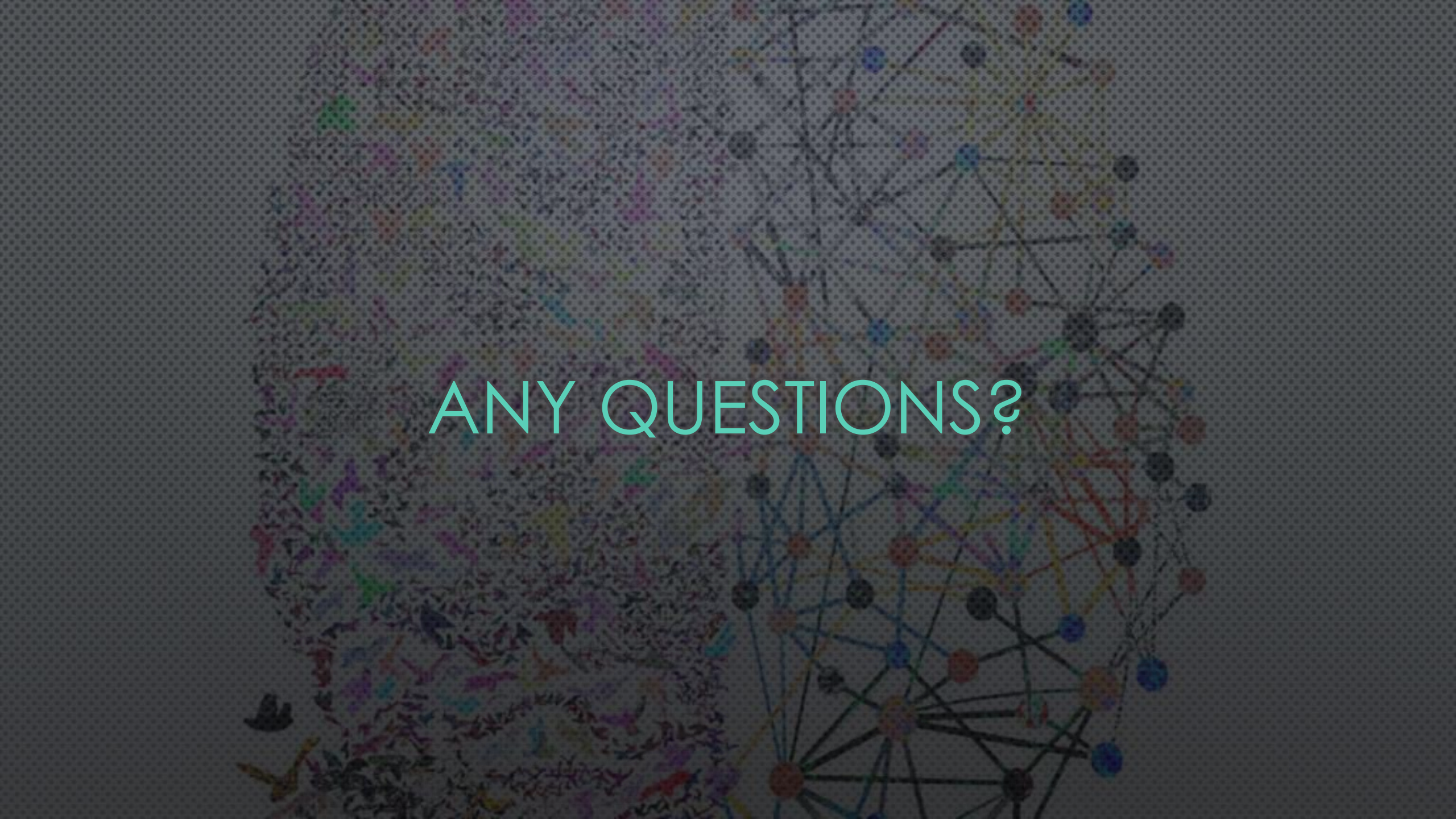


VANILLA GAN



LEAST SQUARES GAN




ANY QUESTIONS?