# [ Paper review 25 ]

# Non-linear Independent Components Estimation (NICE)

## ( Laurent Dinh, et al, 2014 )

## [ Contents ]



# 1. Abstract

propose NICE

- for modeling complex high-dimensional densities
- based on the idea that "good representation = distribution that is easy to model"

Key point

- 1) computing the determinant of Jacobian & inverse Jacobian is trivial
- 2) still learn complex non-linear transformations ( with composition of simple blocks )

# 2. Introduction

## 2.1 Variable transformation

$p_X(x) = p_H(f(x)) \left|\det \frac{\partial f(x)}{\partial x}\right|$

- $\frac{\partial f(x)}{\partial x}$ : Jacobian matrix of function $f$ at $x$

transformation $f$'s properties

- 1) easy determinant of Jacobian
- 2) easy inverse

## 2.2 Key point

split $x$ into 2 blocks $(x_1, x_2)$

$y_1 = x_1$
$y_2 = x_2 + m(x_1)$

- $m$ : arbitrarily complex function

inverse :

$x_1 = y_1$
$x_2 = y_2 - m(y_1)$

# 3. Learning Bijective Transformations of Continuous Probabilities

$\log(p_X(x)) = \log(p_H(f(x))) + \log\left(\left|\det\left(\frac{\partial f(x)}{\partial x}\right)\right|\right)$

- $p_H(h)$ : prior distribution

  ( ex. isotropic Gaussian )

  ( does not need to be constant, could also be learned )

if prior is factorial.... we obtain the following "NICE criterion"

$\log(p_X(x)) = \sum_{d=1}^{D} \log(p_{H_d}(f_d(x))) + \log\left(\left|\det\left(\frac{\partial f(x)}{\partial x}\right)\right|\right)$, where $f(x) = (f_d(x))_{d \leq D}$

Auto-encoders

- $f$ : encoder
- $f^{-1}$ : decoder

# 4. Architecture

## 4.1 Triangular Structure

obtain a family of bijections

- 1) whose Jacobian determinant is tractable
- 2) whose computation is straight forward

Jacobian determinant is the product of its layer's Jacobian determinants

$f = f_L \circ \ldots \circ f_2 \circ f_1$

affine transformations

- inverse & determinant when using diagonal matrices

$M = LU$

- $M$ : square matrices
- $L, U$ : upper and lower triangular matrices

HOW?

- method 1) build a NN with traingular weights..

  $\rightarrow$ constrained.....
- method 2) consider a family of functions with "triangular Jacobians"

# 4.2 Coupling Layer

(1) bijective transformation

(2) triangular Jacobian

(1)+(2) = "tractable Jacobian determinant"

## General Coupling layer

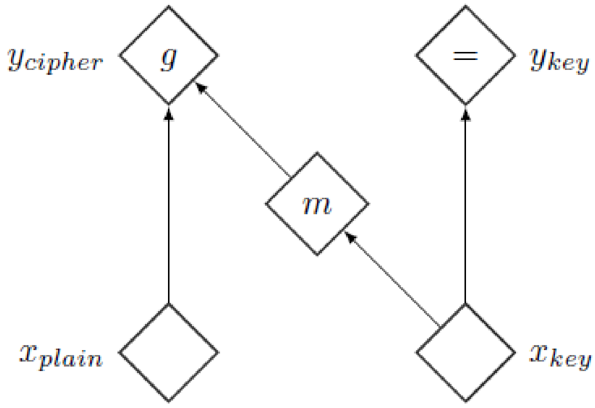


Figure 2: Computational graph of a coupling layer

$$y_{I_1} = x_{I_1}$$
$$y_{I_2} = g\left(x_{I_2}; m\left(x_{I_1}\right)\right)$$

thus, $\frac{\partial y}{\partial x} = \begin{bmatrix} I_d & 0 \\ \frac{\partial y_{I_2}}{\partial x_{I_1}} & \frac{\partial y_{I_2}}{\partial x_{I_2}} \end{bmatrix}$, and $\det \frac{\partial y}{\partial x} = \det \frac{\partial y_{I_2}}{\partial x_{I_2}}$

inverse

$$
\begin{aligned}
x_{I_1} &= y_{I_1} \\
x_{I_2} &= g^{-1}\left(y_{I_2} ; m\left(y_{I_1}\right)\right)
\end{aligned}
$$

### Additive Coupling Layer

$g\left(x_{I_2} ; m\left(x_{I_1}\right)\right)=x_{I_2}+m\left(x_{I_1}\right)$

That is...

$$
\begin{aligned}
y_{I_2} &= x_{I_2}+m\left(x_{I_1}\right) \\
x_{I_2} &= y_{I_2}-m\left(y_{I_1}\right)
\end{aligned}
$$

thus, $\frac{\partial y}{\partial x}=\left[\begin{array}{cc} I_d & 0 \\ \frac{\partial y_{I_2}}{\partial x_{I_1}} & \frac{\partial y_{I_2}}{\partial x_{I_2}} \end{array}\right]$, and $\operatorname{det} \frac{\partial y}{\partial x}=\operatorname{det} \frac{\partial y_{I_2}}{\partial x_{I_2}}=1$

### Combining Coupling Layers

## 4.3 Allowing Rescaling

each additive coupling layers has unit Jacobian determinant ( = volume preserving )

$\rightarrow$ lets include "diagonal scaling matrix $S$"

allows the learner to give more weight on some dimension!

( low $S_{ii}$, less important latent variable $z_i$ )

Then, NICE criterion :

- $\log \left(p_X(x)\right)=\sum_{i=1}^D\left[\log \left(p_{H_i}\left(f_i(x)\right)\right)+\log \left(\left|S_{ii}\right|\right)\right]$

## 4.4 Prior distributions

factorized distributions : $p_H(h)=\prod_{d=1}^D p_{H_d}\left(h_d\right)$

- Gaussian :
  $\log \left(p_{H_d}\right)=-\frac{1}{2}\left(h_d^2+\log (2 \pi)\right)$
- Logistic :
  $\log \left(p_{H_d}\right)=-\log \left(1+\exp \left(h_d\right)\right)-\log \left(1+\exp \left(-h_d\right)\right)$