# Data Driven Methods:Deep Residual Learning And PDEs.

Yiping Lu

Peiking University, School of Mathematic Science

*1500010638@pku.edu.cn*

May 26, 2017

# Overview

# Designing PDEs For Computer Vision

**Image Denoising:** Learning diffusion PDEs.

PM(**Perona and Malik**) Equation is a traditional PDE to processing image.

$$u_t = div(g(|\nabla u|)\nabla u)$$

$g(x)$ here is always taken as $g(x) = \frac{1}{1+kx^2}$

# Learning Optimized Reaction Diffusion

**Naive Approach:** PDE Design via Optimal Control(ECCV 2010)

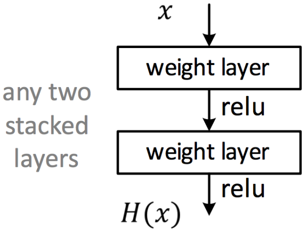
**Key Idea:** Learning Via Optimal Control.

**min** Loss function
**subject to** A PDE Designed with parameter

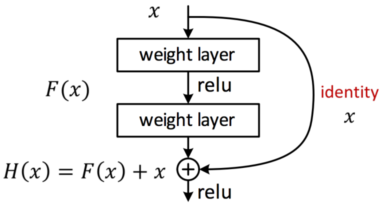
# Residual Learning(CVPR2016)

**Champion of ImageNet,COCO challenge 2015**

- Plaint net




- Residual net

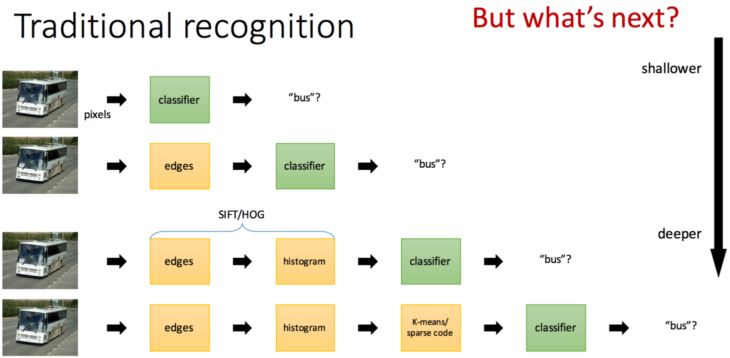



Every block of ResNet runs as

$$H(x)=F(x)+x$$

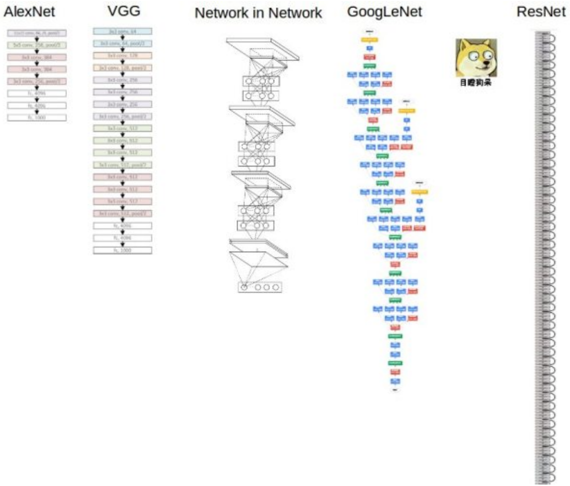
hope the 2 weight layer fit the residual $F(x)$

# Evolution in depth



# Evolution in depth

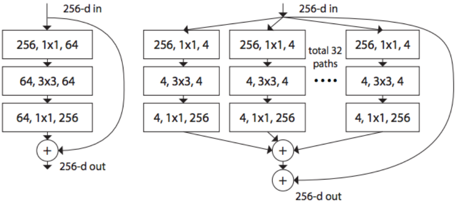
ResNet can become very very deep.



# Aggregated Residual Transformations(CVPR2017)

**ResNetX**

1. Wide
2. Depth



# Consider ResNet as PDE

**New Approach:** Deep Learning(ResNet).(CVPR 2015)

## Main Observe

- The similarity between the forward difference scheme and the shortcut in PDE.
- An FIR filter can be consider as a finite difference scheme of a differential operator.

## Main Idea

Under this observe, we may write every iterate of ResNet as

$$u_t = F(u) = \sum_{\alpha} f_\alpha(D^\alpha u)$$

# Filter And Differential Operator

Identify Differential Operator via **Vansing Moment**

## Vanishing Moment

An FIR highpass filter **q** have vanishing moments of order $\alpha = (\alpha_1, \alpha_2)$, where $\alpha \in Z_+^2$ provided that

$$\sum_{k \in Z_+^2} k^\beta \mathbf{q}[k] = i^{|\beta|} \frac{\partial^\beta}{\partial \omega^\beta} \hat{\mathbf{q}}(\omega)|_{\omega=0} = 0$$

# FIR filters

**Theorem (FIR filters as Differential Operator)**

*Let* $\mathbf{q}$ *be an FIR highpass filter with vanishing moments of oder* $\alpha \in Z_+^2$. *Then for a smooth function* $F(x)$ *on* $R^2$, *we have*

$$\frac{1}{\delta^{|\alpha|}} \sum_{k \in Z^2} \mathbf{q}[k] F(x + \epsilon k) = C_\alpha \frac{\partial^\alpha}{\partial x^\alpha} F(x) + O(\epsilon)(\epsilon \to 0)$$

*Here* $C_\alpha = \frac{1}{\alpha!} \sum_k k^\alpha \mathbf{q}[k]$

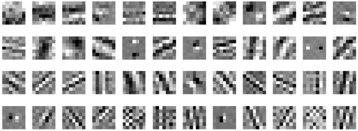
**Proof.**

The proof is by straightforward calculation based on Taylor's expansion. □

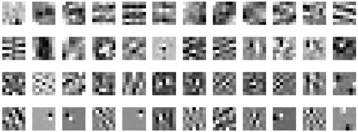
# FIR filters

**Why our filtes are better?**

- Data Driven Design.
- May change via time:some times means faster converge speed in variational problems.
- When the vanish moment is higher, it will have higher order truncation error.



(a) 48 filters of size 7 × 7 in stage 1

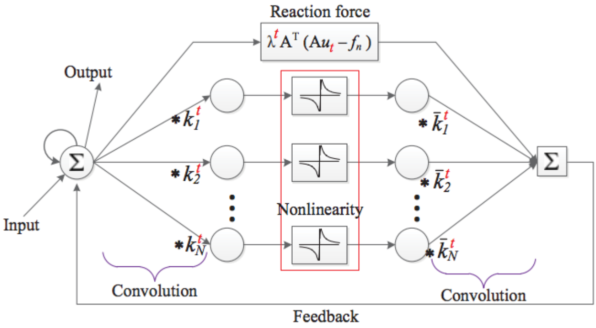


(b) 48 filters of size 7 × 7 in stage 5

# Learning Optimized Reaction Diffusion

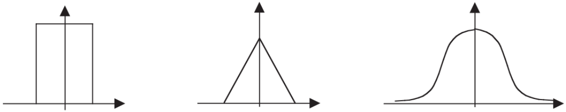
Ignoring the coupled relation between $\nabla_x u$ and $\nabla_y u$, the P-M Equation can be written as:

$$\frac{u_t - u_{t-1}}{\Delta t} = -\sum_{i=1}^{N_k} (K_i^t)^T \psi_i^t (K_i^t u_{t-1}) - \phi(u_{t-1}, f_n)$$



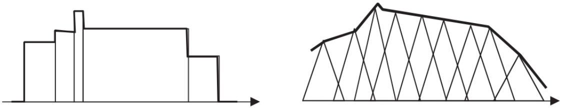
# How to approximate $\psi$

**Radial Basis Networks(RBN)!** A radial basis network is a feed-forward neural network using the radial basis activation function. Some Examples are depicted in the Fig below.



1.10 Three examples of one-dimensional radial basis functions.

We can use RBN to approximate functions by $\hat{\psi}(x) = \sum_{i=1}^{c} \omega_i \phi(\frac{||x-m_i||}{\sigma_i})$!



1.11 Two examples illustrating radial basis function approximation.

# Experiment

**Fig.**Gaussian Noise Denoising Test:(a) clean image(b) noised image (c) BM3D (d) EPLL (e) SRCNN (f) WNNM (g)5×5 filters (h) 7×7 filters (i) 5×5 filters with multi-scale learning (j)7×7 filters with multi-scale learning

# Equation We consider

Consider Heat Equation, Kolmogorov's Equation and Fokker-Planck Equation which can be written in a generalized form as:

$$u_t = WD^{\alpha}u$$

Here $W$ is the weight of a linear transform and $D^{\alpha}$ contains all differential operator whose oder no larger that $\alpha$

# Problem:Confusion

For kernel $k_1$ and $k_2$ we have

$$k_1 \circledast f + k_2 \circledast f = (k_1 + k_2) \circledast f$$

As an Example:$[1, -2, 1] + [1, -1, 0] = [2, -3, 1] + [0, 0, 0]$, the second order differential operator may disappear.

# Keypoint:Multi-Scale

## Avoid Confusion Via Multi-Scale Learning

In order to avoid the confusion in linear network, we shall consider the scale. If our equation includes $n$ order of differential operators, we shall utilize $n$ scales, the reason is shown below:

At scale $x$ the processing can be written as:

$$f := f + \left(\sum_{i=1}^{n} \frac{1}{x^i} k_i\right) \circledast f$$

# Generize the form of PDE

Consider a generalized form as:

$$u_t = Wf(D^\alpha u)$$

Here $W$ is the weight of a linear transform and $D^\alpha$ contains all differential operator whose oder no larger that $\alpha$

**We assume that the function$f$ is separable.**

# Example

What our framework can do for you?

- Transport Equation: $u_t + \sum_{i=1}^{n} b_i u_{x_i} = 0$
- Heat Equation: $u_t - a^2 \Delta u = 0$
- Kolmogorov Equation & Fokker-Planck Equation
- Airy's Equation $u_t + u_{xxx} = 0$
- Beam Equation $u_t + u_{xxxx} = 0$
- Scalar Reaction-diffusion Equation $u_t - \Delta u = f(u)$
- $\cdots$
- **May Also Find New Equations.**
- **New Finite Differential Scheme Designed Date Driven.**

# Numerical Test

**Transport Equation**: $u_t + cu_x = 0$

TABLE 2. Learned $3 \times 3$ Filters From Transport Equation(Without noise)

| | | |
|---|---|---|
| 0.1565 | 0.2254 | -0.3809 |

TABLE 3. Learned $5 \times 5$ Filters From Transport Equation(Without noise)

| | | | | |
|---|---|---|---|---|
| 0.0940 | -0.0620 | -0.0310 | 0.4246 | -0.4280 |

# Numerical Test

**Heat Equation**: $u_t = \Delta u, u_t = G_\delta \circledast u_0$

| | Vanishing Moment | | | | | |
|---|---|---|---|---|---|---|
| Filters | (0,0) | (1,0) | (0,1) | (1,1) | (2,0) | (0,2) |
| **Zero Order** | **-0.010496** | -0.004907 | 0.010003 | 0.066484 | 0.014833 | 0.065231 |
| **First Order** | -0.006245 | **-0.00336** | **0.002333** | 0.053965 | 0.022562 | 0.033246 |
| **Second Order** | 0.002697 | 0.019155 | 0.013385 | **-0.020594** | **0.223597** | **0.242161** |

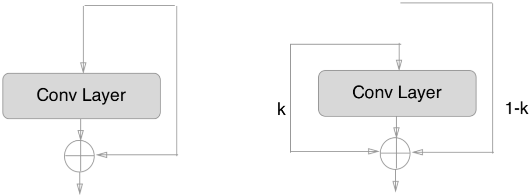
# PDE And Network.

PDE tricks can become Network tricks!
As an example:

$$\text{Primal Equation:} \qquad u_t = f(u)$$

$$\text{Accelerated Equation:} \qquad u_{tt} + cu_t = f(u)$$

Turns into Network:



# Numerical Test

Give a test on the CIFAR-10 dataset:

Error rate:

$$\textbf{ResNet20}:8.75 \to \textbf{NResNet20}:7.53$$
$$\textbf{ResNet32}:7.51 \to \textbf{NResNet32}:6.93$$

# References


K He,X Zhang,S Ren,J Sun (2015,MSRA)
Deep Residual Learning for Image Recognition
*CVPR* 2016

K He,X Zhang,S Ren,J Sun (2015,MSRA)
Identity Mappings in Deep Residual Networks
*CVPR* 2016

RK Srivastava,K Greff,J Schmidhuber (2015)
Highway Networks
*Computer Science* 2015

Y. H. Hu and J.-N. Hwang.
Handbook of neural network signal processing.
*CRC press* 2010

# References


- Bin Dong, Qingtang Jiang and Zuowei Shen
  Image restoration: wavelet frame shrinkage, nonlinear evolution PDEs, and beyond
  *Multiscale Modeling and Simulation: A SIAM Interdisciplinary Journal* ,15(1),606-660,2017.

- Yunjin Chen, Wei Yu and Thomas Pock
  On learning optimized reaction diffusion processes for effective image restoration
  *CVPR* 2015

- K Gregor,Y Lecun
  Learning Fast Approximations of Sparse Coding
  *ICML*2010

- R Liu,Z Lin,W Zhang,Z Su
  Learning PDEs for image restoration via optimal control
  *ECCV*2010

# The End