Evolutionary Computation and Evolutionary Deep Learning for Image Analysis, Signal Processing and Pattern Recognition

Mengjie Zhang[1] and Stefano Cagnoni[2]
1 Evolutionary Computation Research Group, Victoria University of Wellington, Wellington, New Zealand
2 IBIS Lab, University of Parma, Parma, Italy
Mengjie.zhang@ecs.vuw.ac.nz, cagnoni@ce.unipr.it

http://gecco-2021.sigevo.org/

GECCO '21 Companion, July 10–14, 2021, Lille, France

ACM ISBN 978-1-4503-8351-6/21/07. ...$15.00
https://doi.org/10.1145/3449726.3461412

Instructors

- **Mengjie Zhang** is a Fellow of Royal Society of NZ, a Fellow of IEEE, and an IEEE Distinguished Lecturer. He is Professor of Computer Science at the School of Engineering and Computer Science, Victoria University of Wellington, New Zealand. His research is mainly focused on evolutionary computation, particularly genetic programming, evolutionary deep and transfer learning, image analysis, feature selection and reduction, and evolutionary scheduling and combinatorial optimisation. He has published over 600 academic papers in refereed international journals and conferences. He is currently an associate editor for over ten international journals (e.g. IEEE TEVC, ECJ, ACM TELO, IEEE TCYB, and IEEE TETCI). He has been serving as a steering committee member and a program committee member for over eighty international conferences. He is a reviewer of research grants for many countries/regions (e.g. Canada, Portugal, Spain, Germany, UK, Netherland, Austria, Mexico, Czech, Italy, HK, Australia, NZ).

- **Stefano Cagnoni** is an Associate Professor at the University of Parma. Recent research grants include participation in the project "SUPER: Supercomputinfg Platform Emilia Romagna" funded by Regione Emilia Romagna, and a "Marie Curie Initial Training Network" grant, for a four-year research training project in Medical Imaging using Bio-Inspired and Soft Computing. Editor-in-chief of the "Journal of Artificial Evolution and Applications" from 2007 to 2010. Since 1999, he has been chair of EvoIASP, an event dedicated to evolutionary computation for image analysis and signal processing, now merged with other events into the EvoApplications conference. Since 2005, he has co-chaired MedGEC, workshop on medical applications of evolutionary computation at GECCO. Co-editor of special issues of journals dedicated to Evolutionary Computation for Image Analysis and Signal Processing. Member of the Editorial Board of the journals "Evolutionary Computation", "Genetic Programming and Evolvable Machines", and "Algorithms".

Tutorial Agenda

- Introduction
- Evolutionary deep learning for image classification
- (Part II) From the EC attic… Re-using (and renovating) forgotten tools
- Summary

Introduction

Computer Vision

- The "art" of making computers see (and understand what they see)
- Computer vision vs image processing
- Sub-topics:
  - Image acquisition
  - Image enhancement
  - Image segmentation
  - 3D-information recovery/feature extraction
  - Image understanding

Computer and Human Vision

HUMAN	COMPUTER
Perception	Image acquisition
Selective information extraction	Feature enhancement (signal/image processing)
Grouping by 'similarity'	Segmentation
Extraction of spatial relationships	3D-information Recovery
Object recognition and semantic interpretation	Image Understanding

Computer and Human Vision

HUMAN	COMPUTER
Perception	Image acquisition
Selective information extraction	Feature enhancement (signal/image processing)
Grouping by 'similarity'	Segmentation
Extraction of spatial relationships	3D-information Recovery
Object recognition and semantic interpretation	Image Understanding

LOW-LEVEL VISION

Computer and Human Vision

HIGH-LEVEL VISION

HUMAN	COMPUTER
Perception	Image acquisition
Selective information extraction	Feature enhancement (signal/image processing)
Grouping by 'similarity'	Segmentation
Extraction of spatial relationships	3D-information Recovery
Object recognition and semantic interpretation	Image Understanding

Application Taxonomy

- EC techniques
  - GA, GP, ES, EP, PSO, DE, LCS, EMO, EDA, etc.
- Solution types
  - Optimisation of parameters of specific solutions (using GA, ES, PSO…)
    Related with a well-defined task or for a whole system
  - Generation of solutions from scratch (GP, …)
    Performance optimization based on specific objective functions
    It is difficult to choose a model with reasonable assumptions
- Role of EC techniques
  - Interactive qualitative comparisons between solutions
  - Generation of emergent collective solutions
    Achievement of higher-level and complex tasks from collective use of trivial, local, hard-wired behaviours: generation of full EC-based solutions, NOT parameter optimization tasks

Applications

- EC techniques: GP, PSO, LCS, EMO
- Image Analysis
  - Object tracking
  - Edge detection
  - Segmentation
  - Motion detection
  - Object/digit recognition
- English stress detection(signal processing)
- Pattern Recognition: feature selection and biomarker detection

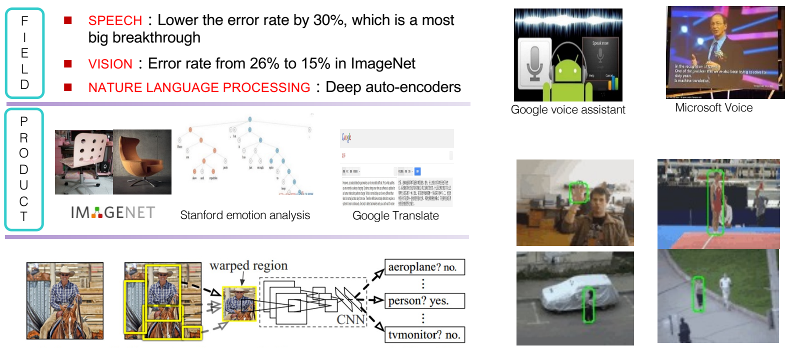
Evolutionary Deep Learning for Image Classification (Part I)

Evolutionary Deep Learning

- Deep Learning – personal view
  - Definition
  - NN-based deep learning
  - Non-NN type deep learning
- Evolutionary Deep Learning – personal view
  - evolving NNs/neuro-evolution → evolutionary deep learning
  - GAs/PSO/GP for evolving NNs
  - GP for deep learning
- Examples of EvoDL for Image Classification
  - GAs for evolving auto-encoders for image classification
  - GAs for evolving CNNs for image classification
  - PSO for evolving CNNs for image classification
  - Surrogate based method for EvoDL acceleration
  - GP for evolving deep structures for image classification
- Summary

Deep Learning -- Overview

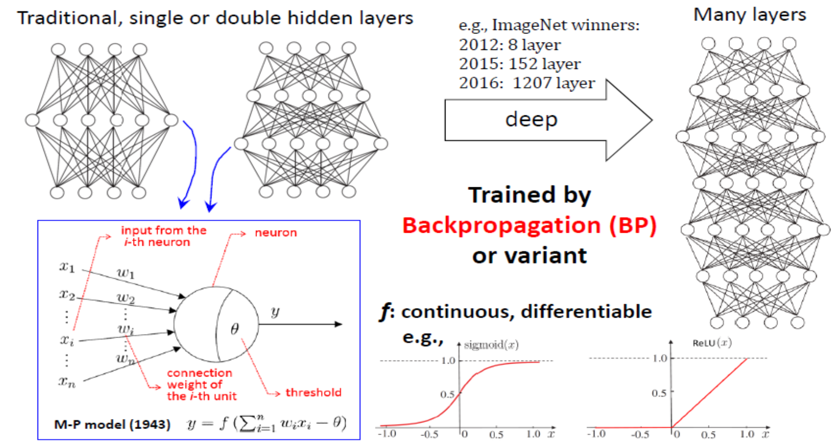
- ❖ It aims at learning hierarchical/meaningful representations through a *deep* and *non-linear* transformation



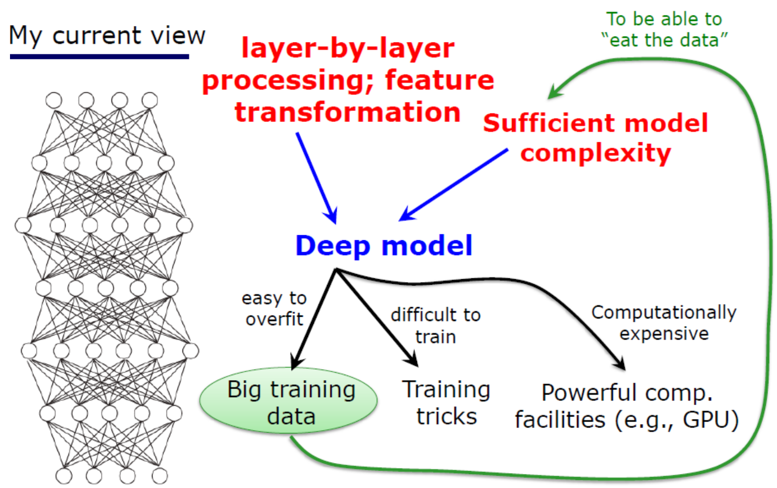
AI = ML = DL = DNNs = CNNs ?

Deep Learning -- Definition

- ❖ What is "Deep Learning"? Deep Learning = Deep Neural Networks?



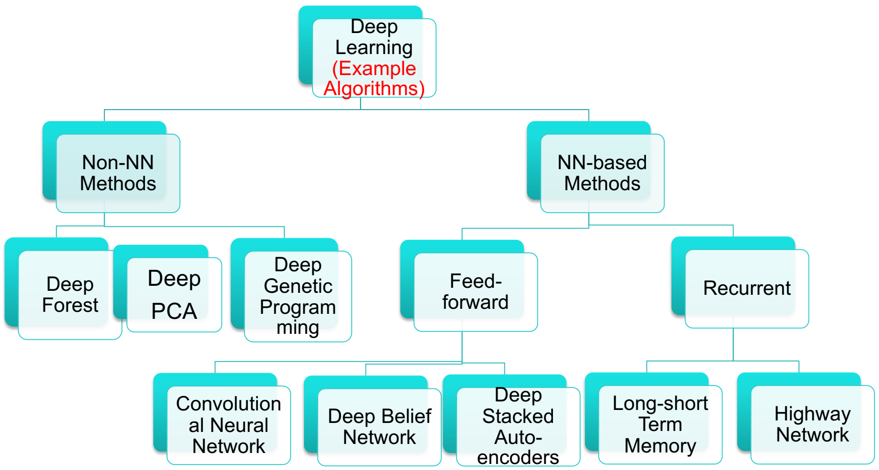
Deep Learning – Definition [Zhou]



Deep Learning – My View

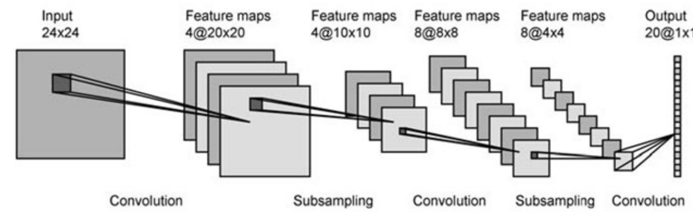
- ❖ Layer-by-layer processing
- ❖ Feature transformation
  - Feature extraction
  - Feature construction
  - Feature learning
- ❖ Sufficient model complexity
  - Complexity ≠ the number (#) of nodes, layers
  - Including function complexity
  - Not necessarily symmetrical
- ❖ #examples?
- ❖ Interpretation?

Deep Learning Methods



Convolutional Neural Networks

- Supervised Deep Learning method, dominant DL algorithm
- Rumelhart and PDP Group's T-C Problem of weight Sharing [Chap 8,1986]
- Yan LeCun's SWNNs [1989, 90, …]
- A CNN is composed of multiple convolutional layers, pooling layers and fully-connected layers [1998?]
- State-of-the-art CNNs: VGG (2015), ResNet (2015), DenseNet [2016]

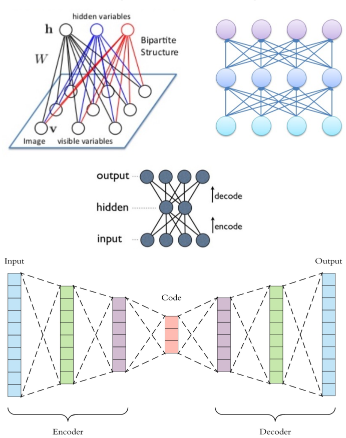


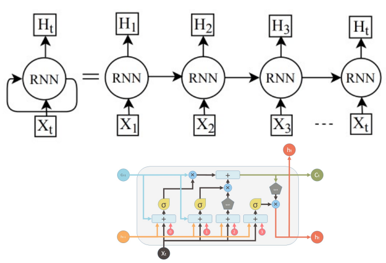
Architecture of LeNet-5

Deep Belief Networks, Deep (Stacked) Auto-encoders, and LSTMs

- Pioneering work on Deep Learning
  - A Fast Learning Algorithm for Deep Belief Nets
    Published in Neural Computing, 2006, Hinton and etc.,
- Unsupervised Deep Learning method (DRBMs)

Geoffrey Hinton





LSTM

Non-NN Deep Learning

- Deep Convex Net [2011]
- PCA Net [2014/15]
- Deep FisherNet [2016]
- Deep Forest learning [2017]
- Genetic Programming based Deep Structures/Learning
  - 2012: GP is doing (evolutionary) deep learning
  - 2018: EC is deep learning

Disadvantages of NN-based DL methods

- Too many hyper-parameters
- Currently, gradient-based algorithms are used to train the weights
- Model complexity fixed once structure decided; usually, more than sufficient
- Big training data required
- Theoretical analysis difficult
- Blackbox and interpretation hard
- Architectures of state-of-the-art NN DL methods are manually designed
  - ResNet, DensNet, VGG, Maxout, Highway, GPT, All CNN

  Issues:
  - Increasingly deep and complex
  - Require expertise in both DNNs and the problem domain
  - Computationally expensive
- EC) methods work well in addressing non-convex/no-differentiable problems, and do not require domain knowledge

Automated Design of Deep Neural Networks using EC!!!

Evolutionary Deep Learning

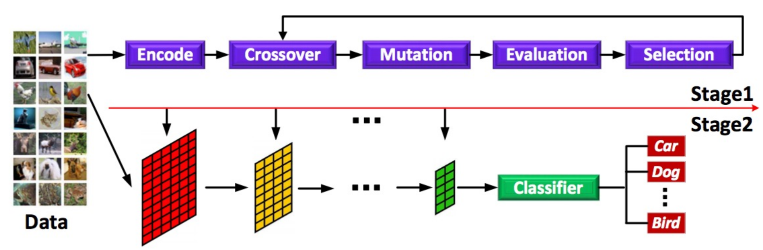
- Three stages:
  - evolving NNs →
  - neuro-evolution →
  - evolutionary deep learning
- GAs/PSO/DE/GP for evolving DNNs
- GP for deep learning

Evolutionary Deep Learning – EC for Evolving NNs

- EC methods have been successfully used to optimize the architecture and even the weights of neural networks over 20 years ago (Yao 1999)
  - Neuro-genetic evolution (Ronald 1994), Cellular Encoding (Gruau 1994)
  - GNARL (Angeline 1994), EPNet (Yao 1997), NEAT (Stanley and Miikkulainen 2002)
  - HyperNEAT (Stanley 2008), ES-HyperNEAT (Risi, Stanley 2012)
  - EANT/EANT2 (Kassahun and Sommer 2005), (Siebel and Sommer 2007)
  - ICONE (Rempis 2012), DXNN (Sher 2012), SUNA (Vargas 2016), MABE (Bohm 2016)
  - CMA-HAGA (Rostami 2016/17), …
- Neural networks were typically shallow and have a small number of parameters
- NEAT and its variants are capable of address the problem regarding median-scale neural networks
- Recently, a number of EC-based new methods have been proposed to automatically evolve/learn DNNs

Evolving Unsupervised DNN

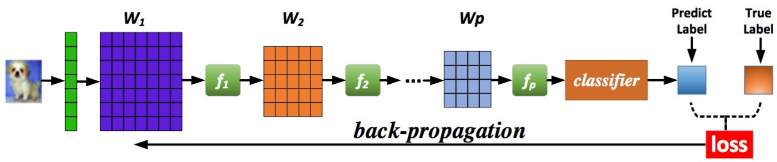
- One method using GA to automatically evolve unsupervised DNN
- The goal is achieved by two stages:
  - Architecture and initialized weights are evolved for building blocks
  - Stacked building blocks stacked are trained by Stochastic Gradient Descent



Yanan Sun, Gary G. Yen, Zhang Yi, "Evolving Unsupervised Deep Neural Networks for Learning Meaningful Representations". IEEE Transactions on Evolutionary Computation. DOI:10.1109/TEVC.2018.2808689.

Evolving Unsupervised DNN

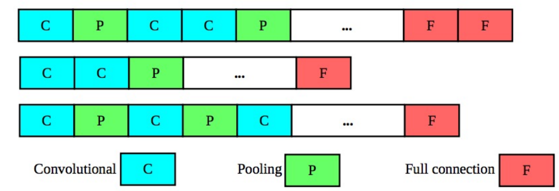
- Evolved building blocks are stacked with the architecture and weight initialization values
- Using SGD to achieve the best performance of the deep model



Benchmark	EUDNN		DAE	CAE	SAE	DBN
	AE	RBM				
MNIST	0.9878(0.00751)	**0.9885(0.00255)**	0.9820(0.00506)(+)	0.9843(0.00699)(+)	0.9832(0.00891)(+)	0.9771(0.00959)(+)
MNIST-basic	0.9674(0.00616)	0.9633(0.00473)	0.9580(0.00352)(+)	0.9635(0.00831)(+)	**0.9776(0.00585)(-)**	0.9658(0.00550)(+)
MNIST-rot	**0.7952(0.00917)**	0.7549(0.00286)	0.7274(0.00757)(+)	0.7706(0.00754)(+)	0.7852(0.00380)(+)	0.7639(0.00568)(+)
MNIST-back-rand	0.8843(0.00076)	0.8386(0.00054)	0.7725(0.00531)(+)	0.5741(0.00779)(+)	**0.8851(0.00934)(=)**	0.8221(0.00130)(+)
MNIST-back-image	0.4325(0.00569)	**0.4830(0.00469)**	0.4022(0.00012)(+)	0.4010(0.00337(+)	0.4638(0.00162)(+)	0.4587(0.00794)(+)
MNIST-rot-back-image	**0.8925(0.00906)**	0.8879(0.00815)	0.8691(0.00127)(+)	0.6574(0.00913)(+)	0.8733(0.00632)(+)	0.8830(0.00098)(=)
Rectangles	0.9627(0.00311)	**0.9681(0.00829)**	0.9232(0.00166)(+)	0.6275(0.00602)(+)	0.9408(0.00263)(+)	0.9622(0.00154)(=)
Rectangles-image	0.7521(0.00689)	0.7716(0.00048)	0.7598(0.00451)(+)	**0.7810(0.00784)(=)**	0.7725(0.00002)(-)	0.7628(0.00913)(+)
Convex	**0.8113(0.00052)**	0.8085(0.00826)	0.7930(0.00538(+)	0.8016(0.00996)(+)	0.8053(0.00878)(+)	0.7895(0.00443)(+)
Cifar10-bw	**0.4798(0.00107)**	0.4331(0.00962)	0.4309(0.00005)(+)	0.4860(0.00775)(+)	0.4423(0.00817)(+)	0.4598(0.00869)(+)
	+/-/=		10/0/0	9/0/1	7/2/1	8/0/2

EvoCNN

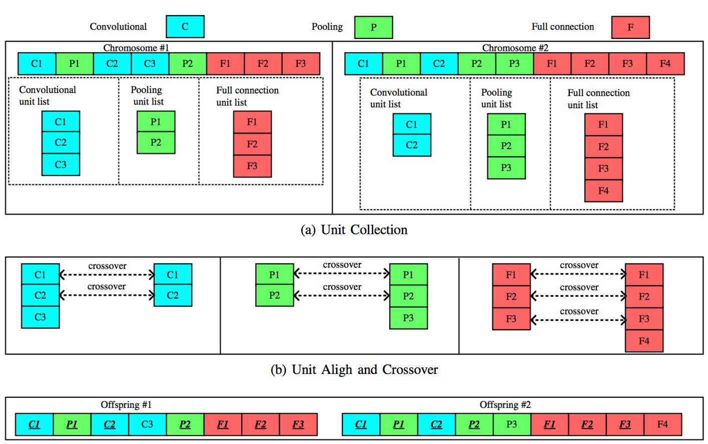
- One method using GA to evolve the architectures and the weight initialization of CNNs
- Designed a variable-length individual method encoding CNNs with unequal depths
- Proposed a crossover operator for individuals with different lengths
- Train the individual with a small number of epochs to find the potentially better one
- Find the best one when the evolutionary process terminates, and then fully trained it for the best performance
- Representation:



Sun, Yanan, Bing Xue, and Mengjie Zhang. "Evolving deep convolutional neural networks for image classification.". IEEE Transactions on Evolutionary Computation. 2019. DOI: 10.1109/TEVC.2019.2916183

EvoCNN

- Crossover operation is composed of three phases: UC, UAC and UR



(a) Unit Collection

(b) Unit Aligh and Crossover

(c) Unit Restore

EvoCNN

- Comparisons on the FASHION dataset

classifier	error(%)	# parameters	# epochs
2C1P2F+Drouout	8.40(+)	3.27M	300
2C1P	7.50(+)	100K	30
3C2F	9.30(+)	—	—
3C1P2F+Dropout	7.40(+)	7.14M	150
GRU+SVM+Dropout	10.30(+)	—	100
GoogleNet [41]	6.30(+)	101M	—
AlexNet [3]	10.10(+)	60M	—
SqueezeNet-200 [51]	10.00(+)	500K	200
MLP 256-128-64	10.00(+)	41K	25
VGG16 [52]	6.50(+)	26M	200
EvoCNN (best)	5.47	6.68M	100
EvoCNN (mean)	7.28	6.52M	100

EvoCNN

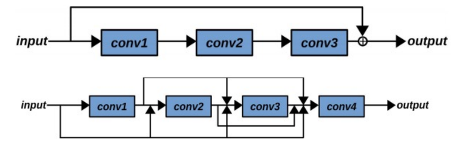
❖ Comparisons on the MNIST variants

classifier	MB	MRD	MRB	MBI	MRDBI	Rectangle	RI	Convex
CAE-2 [53]	2.48(+)	9.66(+)	10.90(+)	15.50(+)	45.23(+)	1.21(+)	21.54(+)	—
TIRBM [54]	—	4.20(-)	—	—	35.50(+)	—	—	—
PGBM+DN-1 [55]	—	—	6.08(+)	12.25(+)	36.76(+)	—	—	—
ScatNet-2 [56]	1.27(+)	7.48(+)	12.30(+)	18.40(+)	50.48(+)	0.01(=)	8.02(+)	6.50(+)
RandNet-2 [57]	1.25(+)	8.47(+)	13.47(+)	11.65(+)	43.69(+)	0.09(+)	17.00(+)	5.45(+)
PCANet-2 (softmax) [57]	1.40(+)	8.52(+)	6.85(+)	11.55(+)	35.86(+)	0.49(+)	13.39(+)	4.19(-)
LDANet-2 [57]	1.05(-)	7.52(+)	6.81(+)	12.42(+)	38.54(+)	0.14(+)	16.20(+)	7.22(+)
SVM+RBF [50]	3.03(+)	11.11(+)	14.58(+)	22.61(+)	55.18(+)	2.15(+)	24.04(+)	19.13(+)
SVM+Poly [50]	3.69(+)	15.42(+)	16.62(+)	24.01(+)	56.41(+)	2.15(+)	24.05(+)	19.82(+)
NNet [50]	4.69(+)	18.11(+)	20.04(+)	27.41(+)	62.16(+)	7.16(+)	33.20(+)	32.25(+)
SAA-3 [50]	3.46(+)	10.30(+)	11.28(+)	23.00(+)	51.93(+)	2.41(+)	24.05(+)	18.41(+)
DBN-3 [50]	3.11(+)	10.30(+)	6.73(+)	16.31(+)	47.39(+)	2.61(+)	22.50(+)	18.63(+)
EvoCNN (best)	1.18	5.22	2.80	4.53	35.03	0.01	5.03	4.82
EvoCNN (mean)	1.28	5.46	3.59	4.62	37.38	0.01	5.97	5.39

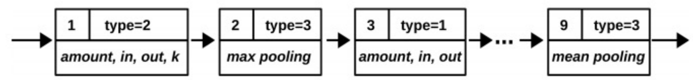
AE-CNN

❖ An automation method for CNN architecture design based on blocks

❖ The blocks are the build blocks of state-of-the-art CNNs (ResNet and DenseNet)



❖ Each gene denotes one type of block and its associated parameters, the fully-connect layer is not used, using variable-length encoding strategy



Yanan Sun, Bing Xue, Mengjie Zhang, Gary G. Yen. Completely automated CNN architecture design based on blocks. IEEE Transactions on Neural Networks and Learning Systems, DOI: 10.1109/TNNLS.2019.2919608.

AE-CNN

❖ Fully training each individual and using the classification accuracy as the fitness, compared with state-of-the art CNNs, semi-auto NAS, and Auto NAS

	CIFAR10	CIFAR100	# of Parameter	GPU Days	
DenseNet (k=12)	5.24	24.42	1.0M	–	hand-crafted architecture
ResNet (depth=101)	6.43	25.16	1.7M	–	hand-crafted architecture
ResNet (depth=1,202)	7.93	27.82	10.2M	–	hand-crafted architecture
Maxout	9.3	38.6	–	–	hand-crafted architecture
VGG	6.66	28.05	20.04M	–	hand-crafted architecture
Network in Network	8.81	35.68	–	–	hand-crafted architecture
Highway Network	7.72	32.39	–	–	hand-crafted architecture
All-CNN	7.25	33.71	–	–	hand-crafted architecture
FractalNet	5.22	22.3	38.6M	–	hand-crafted architecture
Genetic CNN	7.1	29.05	–	17	semi-automatic algorithm
Hierarchical Evolution	3.63	–	–	300	semi-automatic algorithm
EAS	4.23	–	23.4M	10	semi-automatic algorithm
Block-QNN-S	4.38	20.65	6.1M	90	semi-automatic algorithm
Large-scale Evolution	5.4	–	5.4M	2,750	completely automatic algorithm
Large-scale Evolution	–	23	40.4M	2,750	completely automatic algorithm
CGP-CNN	5.98	–	2.64M	27	completely automatic algorithm
NAS	6.01	–	2.5M	22,400	completely automatic algorithm
MetaQNN	6.92	27.14	–	100	completely automatic algorithm
AE-CNN	4.3	–	2.0M	27	completely automatic algorithm
AE-CNN	–	20.85	5.4M	36	completely automatic algorithm

Genetic CNN

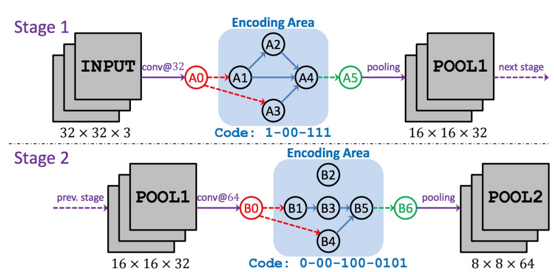
❖ One method using GA evolving CNNs

❖ The encoding process is composed of multiple stages

❖ The maximum number of stages must be predefined, which reflects the depth of the evolved CNN

❖ Each individual is directed trained from scratch

❖ Individuals have the equal lengths



Lingxi Xie and Alan Yuille, "Genetic CNN," in Proceedings of 2017 IEEE International Conference on Computer Vision, Venice, Italy, 2017, pp.1388–1397.

(John-Hopskins Uni, USA)

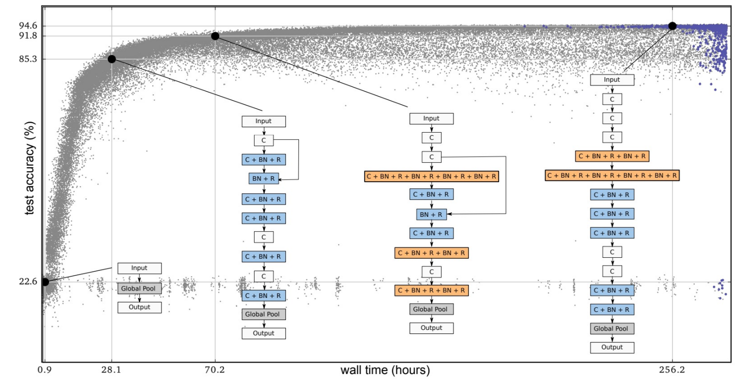
Large-scale Evolution

- One method using GA to evolve architectures of CNNs
- Individuals are with unequal lengths
- Only mutation operation, no crossover operation
  - During mutation, the setting of one convolutional operation could be changed, removing or adding new connections
  - Large-scale Evolution defined 12 operations for mutation
- Fitness is the classification accuracy in terms of image classification tasks
- Weights are inherited from the parent individual
- A set of predefined convolutional operations are provided

Esteban Real, Sherry Moore, Andrew Selle, Saurabh Saxena, Yutaka Leon Suematsu, Jie Tan, Quoc Le and Alex Kurakin, "Large-scale evolution of image classifiers," in Proceedings of Machine Learning Research, Sydney, Australia, 2017, pp. 2902–2911.
Google DeepMind

Large-scale Evolution

- Evolutionary process



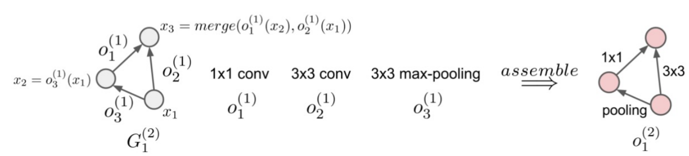
Hierarchical Evolution

- One method using GA to evolve architectures of CNNs
- The whole architecture is evolved by several steps
- In each step, only a small architecture is evolved
- Multiple small architectures are stacked to form a big/deep architecture

Hanxiao Liu, Karen Simonyan, Oriol Vinyals, Chrisantha Fernando and Koray Kavukcuoglu, "Hierarchical representations for efficient architecture search," in Proceedings of 2018 Machine Learning Research (ICML), Stockholm, Sweden, 2018.

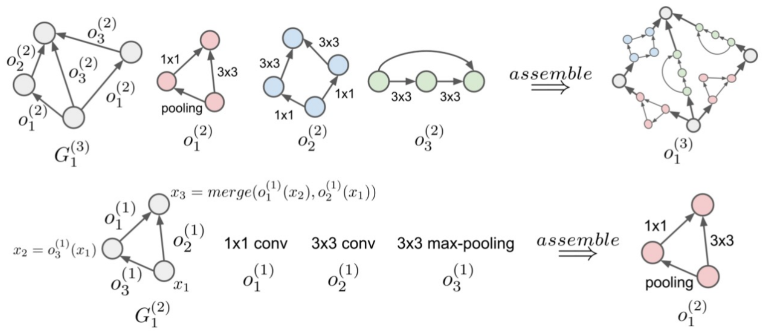
Hierarchical Evolution

- In the first step, a set of primitive operations is provided
  - 1x1 convolution of C channels
  - 3x3 depth-wise convolution
  - 3x3 separable convolution of C channels
  - 3x3 max-pooling
  - 3x3 average-pooling
  - Identify
- Randomly select several primitive operations, and then use a Directed Acyclic Graph to denote the connection between selected operations



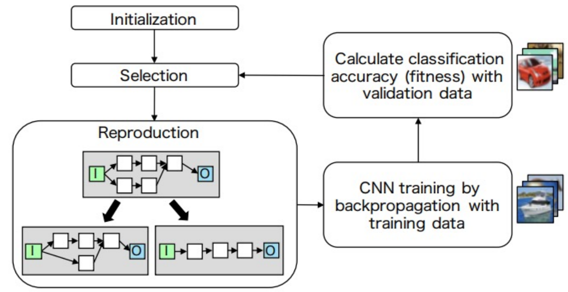
Hierarchical Evolution

- In the second step, the best one found in the previous step is as a new primitive operation, and do the same evolutionary process



CGP-CNN

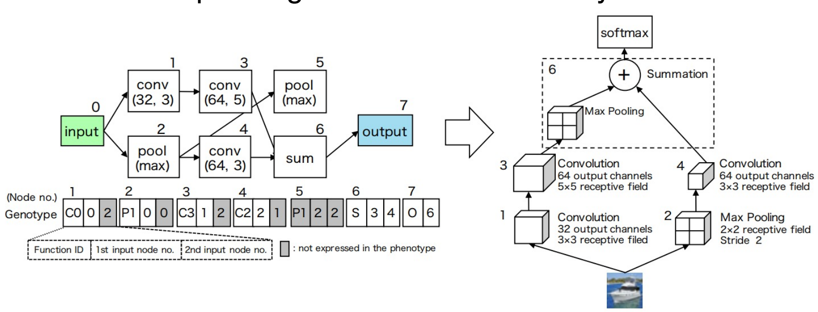
- One genetic programming approach evolving architectures of CNNs
- By providing a set of primitive operations, the Cartesian genetic programming is used to evolve different connections between the primitive operations



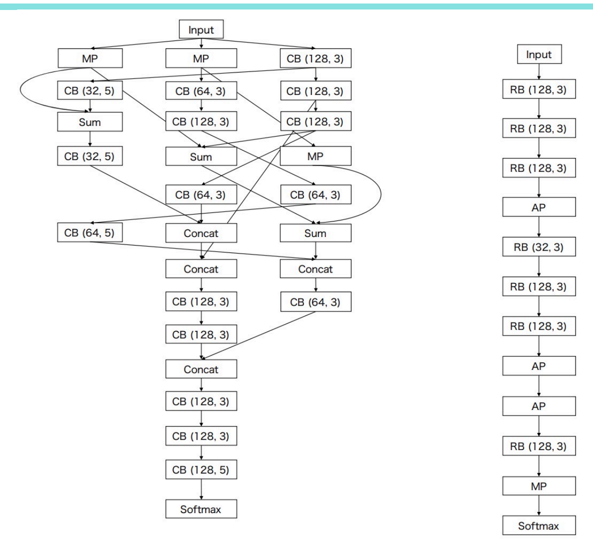
Masanori Suganuma, Shinichi Shirakawa and Tomoharu Nagao, "A genetic programming approach to designing convolutional neural network architectures," in Proceedings of the Genetic and Evolutionary Computation Conference. ACM, 2017: 497-504. Extended version in ECJ.

CGP-CNN

- In encoding process, each operation is encoded by three unit, the first is the index, the second and the third refers to the indices of its input
- Each one is converted to the CNN for fitness evaluation on image classification task, the fitness is the corresponding classification accuracy



CGP-CNN: Example Evolved Networks



(a) CGP-CNN (ConvSet) (b) CGP-CNN (ResSet)

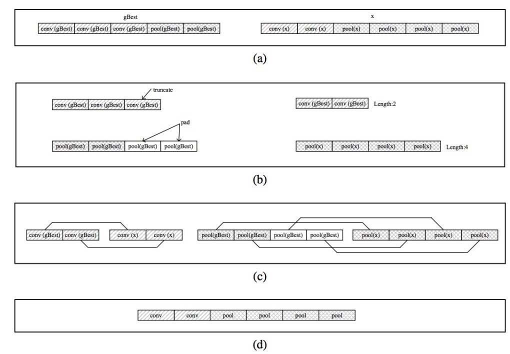
PSOAO

- A PSO method to evolve architectures of the Flexible Convolutional Auto-encoder
  - A traditional convolutional auto-encoder has one encoder and one decoder
    - one encoder is composed of on convolutional layer and on pooling layer
    - one decoder contains only one de-convolutional layer
    - State-of-the-art CNNs do not have such architectures
  - In the flexible convolutional auto-encoder
    - its encoder has multiple convolutional layers and pooling layers
    - It can form the state-of-the-art CNNs
    - but its architecture is not easy to manually tune
- Particles with different lengths to represent different flexible convolutional auto-encoders
- In PSOAO, a x-reference velocity updating strategy is proposed

Yanan Sun, Bing Xue and Mengjie Zhang, "A particle swarm optimization-based flexible convolutional auto-encoder for image classification," IEEE Transactions on Neural Networks and Learning Systems. Volume 30, Issue 8, 2019. pp. 2295 - 2309. DOI: 10.1109/TNNLS.2018.2881143

PSOAO

- In x-reference, gBest and pBest adopt the length of the current particle
- If the lengths of gBest and pBest exceed that of the current particle, truncation is performed, otherwise zeros are padded



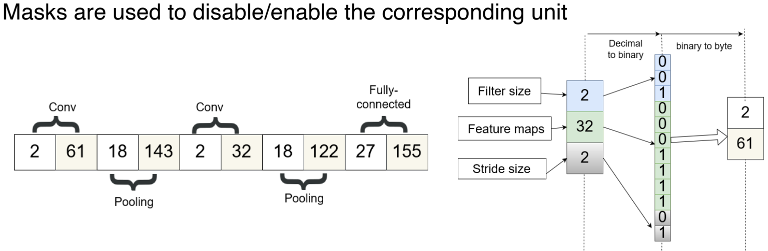
PSOAO

- The performance of flexible convolutional auto-encoder outperforms state-of-the-art auto-encoders and convolutional auto-encoders

Algorithm	CIFAR-10	MNIST	STL-10	Caltech-101
SSAE	74.0 (0.9)	96.29 (0.12)	55.5 (1.2)	66.2 (1.2)
SDAE	70.1 (1.0)	99.06 [25]	53.5 (1.5)	59.5 (0.3)
SCAE	78.2 [16]	99.29 [16]	40.0 (3.1)	58.0 (2.0)
SCRBM	78.9 [63]	99.18 [19]	43.5 (2.3)	65.4 (0.5) [19]
SCDAE-1	75.0 (1.2)	99.17 (0.10)	56.6 (0.8)	71.5 (1.6)
SCDAE-2	80.4 (1.1)	99.38 (0.05)	60.5 (0.9)	78.6 (1.2)
SFCAE-1	78.9 (0.3)	99.30 (0.03)	**61.2 (1.2)**	**79.8 (0.0)**
SFCAE-2	**83.5 (0.5)**	**99.51 (0.09)**	56.8 (0.2)	79.6 (0.0)

IPPSO

- A PSO to effectively evolve the architectures of CNNs
- The encoding strategy is based on IP protocol
- Binary string is used to encode the architecture
- Masks are used to disable/enable the corresponding unit



Bin Wang, Yanan Sun, Bing Xue, Mengjie Zhang, "Evolving deep convolutional neural networks by variable-length particle swarm optimization for image classification," Proceedings of 2018 IEEE Congress on Evolutionary Computation. Rio de Janeiro, Brazil. 8-13 July 2018. pp. 1514-1521.

IPPSO

- Best on MDRBI
- Second Best on MB
- Fifth on Convex

classier	MB	MDRBI	CS
CAE-2	2.48(+)	45.23(+)	–
TIRBM	–	35.50(+)	–
PGBM+DN-1	–	36.76(+)	–
ScatNet-2	1.27(+)	50.48(+)	6.50(-)
RandNet-2	1.25(+)	43.69(+)	5.45(-)
PCANet-2 (softmax)	1.40(+)	35.86(+)	4.19(-)
LDANet-2	1.05(-)	38.54(+)	7.22(-)
SVM+RBF	3.03(+)	55.18(+)	19.13(+)
SVM+Poly	3.69(+)	56.41(+)	19.82(+)
NNet	4.69(+)	62.16(+)	32.25(+)
SAA-3	3.46(+)	51.93(+)	18.41(+)
DBN-3	3.11(+)	47.39(+)	18.63(+)
IPPSO(mean)	1.21	34.50	12.06
IPPSO(best)	1.13	33	8.48
IPPSO(standard deviation)	0.103	2.96	2.25

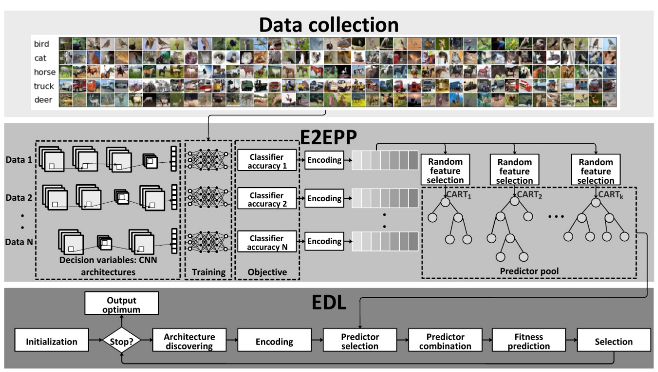
E2EPP

- Almost all state-of-the-art NAS methods train the new architecture from scratch
- Training even one architecture on modern dataset consumes hours to days on GPU servers
- This is computationally expensive problem
  - EC researchers have developed a number of promising algorithms
  - The random forest technique is used to predict the performance of each newly generated architecture

Yanan Sun, Handing Wang, Bing Xue, Yaochu Jin, Gary G. Yen, Mengjie Zhang. Surrogate-Assisted Evolutionary Deep Learning Using an End-to-End Random Forest-based Performance Predictor. IEEE Transactions on Evolutionary Computation, Vol. 24, Issue 2, 2020. pp. 350 – 364. DOI:10.1109/TEVC.2019.2924461.

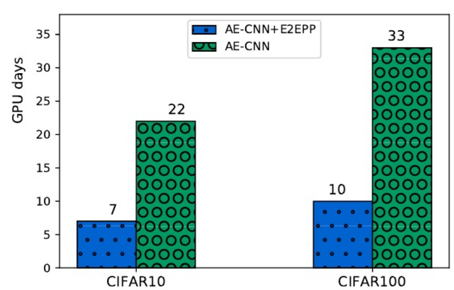
E2EPP

- This is an off-line surrogate-assisted method



E2EPP

- Integrated into AE-CNN, the performance of E2EPP is investigated



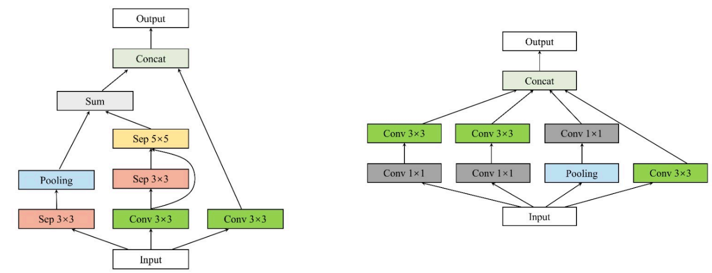
E2EPP

❖ The overall performance compared with state-of-the-art CNNs and NAS methods

	Peer Competitors	CIFAR10	CIFAR100	GPU Days
state-of-the-art CNNs manually designed	DenseNet	94.76	75.58	–
	ResNet (depth=101)	93.57	74.84	–
	ResNet (depth=1,202)	92.07	72.18	–
	Maxout	90.70	61.40	–
	VGG	93.34	7.95	–
	Network in Network	91.19	64.32	–
	Highway Network	92.28	67.61	–
	All-CNN	92.75	66.29	–
	FractalNet	94.78	77.70	–
CNN architecture design algorithms based on non-evolutionary methods	NAS	93.91	–	22,400
	MetaQNN	93.08	27.14	100
	EAS	95.77	–	10
	Block-QNN-S	95.62	79.35	90
CNN architecture design algorithms based on evolutionary methods	Genetic CNN	92.90	70.95	17
	Large-scale Evolution	94.60	77.00	2,750
	Hierarchical Evolution	96.37	–	300
	CGP-CNN	94.02	–	27
	AE-CNN + E2EPP	94.70	77.98	8.5

Graph-based Approach

- Encoding



Code = [(1, 3, 32, -1, 0), (2, 5, 1, -1, 1), (3, 2, 64, -1, 0), (4, 2, 128, -1, 0), (5, 3, 32, -1, 3), (6, 4, 32, -1, 3), (7, 7, 0, 2, 6), (8, 8, 0, 7, 4)]

Gonglin Yuan, Bing Xue, and Mengjie Zhang. " A Graph-Based Approach to AutomaticConvolutional Neural Network Construction forImage Classification ". Proceedings of 2020 the 35th International Conference on Image and Vision Computing New Zealand (IVCNZ 2019). IEEE Press. Wellington, New Zealand, 25-27 December, 2020

Results

COMPARISONS ON THE FASHION DATASET.

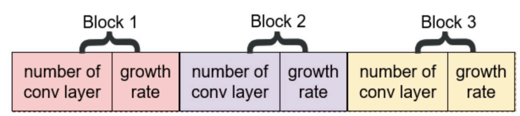
classifier	error(%)	# parameters
2C1P2F+Drouout	8.40(+)	3.27M
2C1P	7.50(+)	100K
3C2F	9.30(+)	—
3C1P2F+Dropout	7.40(+)	7.14M
GRU+SVM+Dropout	10.30(+)	—
GoogLeNet	6.30(-)	101M
AlexNet	10.10(+)	60M
SqueezeNet-200	10.00(+)	500K
MLP 256-128-64	10.00(+)	41K
VGG16	6.50(-)	26M
EvoCNN	7.28(+)	6.52M
DAGCNN(ConvOnly)	6.72	2.51M
DAGCNN	6.94	1.33M

RESULTS OF DIFFERENT METHODS.

Methods	MB	MRD	MRDBI	Convex
CAE-2	2.48(+)	9.66(+)	45.23(+)	—
TIRBM	—	4.20(+)	35.50(+)	—
PGBM+DN-1	—	—	36.76(+)	—
ScatNet-2	1.27(+)	7.48(+)	50.48(+)	6.50(+)
RandNet-2	1.25(+)	8.47(+)	43.69(+)	5.45(+)
PCANet-2	1.40(+)	8.52(+)	35.86(+)	4.19(+)
LDANet-2	1.05(+)	7.52(+)	38.54(+)	7.22(+)
SVM+RBF	3.03(+)	11.11(+)	55.18(+)	19.13(+)
SVM+Poly	3.69(+)	15.42(+)	56.41(+)	19.82(+)
NNet	4.69(+)	18.11(+)	62.16(+)	32.25(+)
SAA-3	3.46(+)	10.30(+)	51.93(+)	18.41(+)
DBN-3	3.11(+)	10.30(+)	47.39(+)	18.63(+)
EvoCNN	1.28(+)	5.46(+)	37.38(+)	5.39(+)
FGP	1.30(+)	8.44(+)	—	1.84(+)
HGAPSO	0.84(-)	—	12.23(-)	1.24(-)
DAGCNN	1.00	4.13	15.88	1.68

MOPSO for Evolving Deep CNN

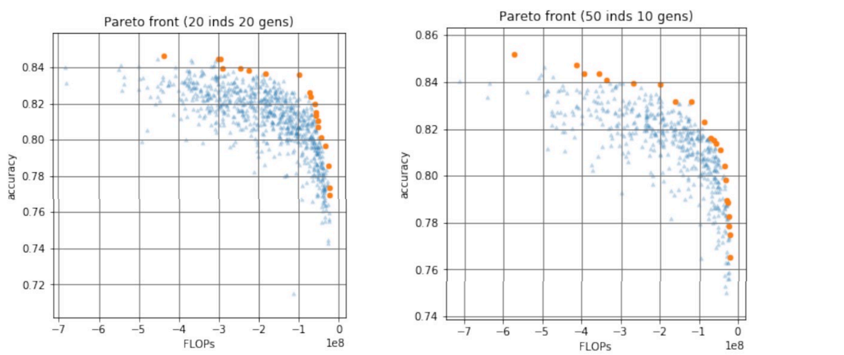
- Representation/Encoding



- Objectives:
  - Classification accuracy
  - Number of FLOPs (floating point operations) --- reflect the computational cost of both training and inference
- OMOPSO (M. R. Sierra and C.Coello Coello, 2005)
- Speedup the training
  - Stop training if accuracy does not increase for 10 epochs
  - prevent the same CNN from the duplicate training
  - Infrastructure

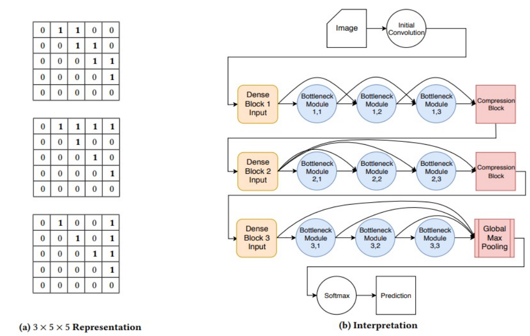
Wang, Bin, Yanan Sun, Bing Xue, and Mengjie Zhang. "Evolving deep neural networks by multi-objective particle swarm optimization for image classification." In Proceedings of the Genetic and Evolutionary Computation Conference, pp. 490-498. 2019.

MOPSO for Evolving Deep CNN



ME-SD-BDC

- Look for skip-connections for more dynamic structures
- ME-SD-BDC – Adjacency Matrix Encoding
  - Each individual encoded as a $3{\times}N{\times}N$ Boolean adjacency matrix, where $N = \frac{\ell-4}{2\cdot 3}+2$



A full example individual and interpretation.

Damien O'Neill, Bing Xue, and Mengjie Zhang. "Neural architecture search for sparse DenseNets with dynamic compression." In Proceedings of the 2020 Genetic and Evolutionary Computation Conference, pp. 386-394. 2020.

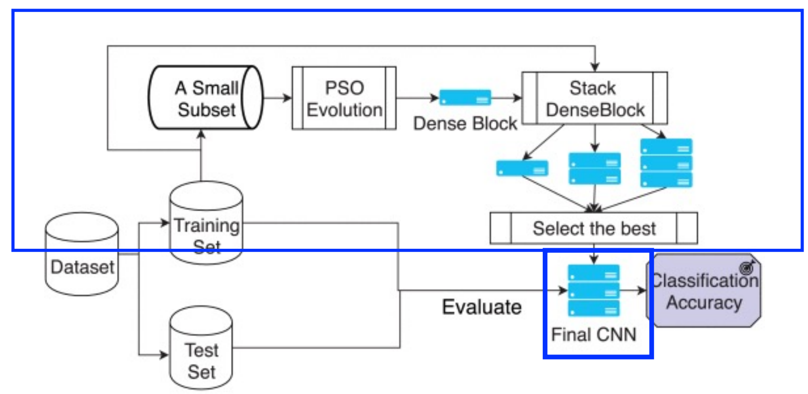
ME-SD-BDC: Results

Network/s	Increase in weights over baseline	CIFAR10 Test Error Improvement	CIFAR100 Test Error Improvement
DenseNet-BC($k = 24, \ell = 250, c = 0.5$)	14.5M	0.73%	4.51%
ME-SD-BDC	$\mu = 0.21$M	$\mu = 0.29\%$	$\mu = 0.72\%$

- Improves DenseNet-BC performance on CIFAR100 and CIFAR10 by refining skip-connection structure, with modest complexity increase
- Analysed networks for insight into novel skip-connection structures

PSO for Evolving and Stacking Transferable Blocks

- Minimise the search space: the vector only consists of two dimensions, which are the *growth rate* and the *number of layers*



Wang, Bin, Bing Xue, and Mengjie Zhang. "Particle swarm optimisation for evolving deep neural networks for image classification by evolving and stacking transferable blocks." In 2020 IEEE Congress on Evolutionary Computation (CEC), pp. 1-8. IEEE, 2020.

Results

Method	CIFAR-10 (Error rate%)	Number of Parameters	Computational Cost
ResNet-110 [3]	6.43	**1.7M**	–
DenseNet(k = 40) [4]	3.74	27.2M	–
EAS [20]	4.23	23.4M	<10 GPU-days
NASNet-A (7 @ 2304) [6]	**2.97**	27.6M	2,000 GPU-days
NASH (ensemble across runs) [21]	4.40	88M	4 GPU-days
NAS v3 max pooling [5]	4.47	7.1M	22,400 GPU-days
AmoebaNet-B (6,128) [16]	**2.98**	34.9M	3150 GPU-days
Hier. repr-n, evolution (7000 samples) [22]	3.75	–	300 GPU-days
CGP-CNN(ResSet) [10]	5.98	**1.68M**	29.8 GPU-days
DENSER [23]	5.87	10.81M	–
GeNet from WRN [9]	5.39	–	100 GPU-days
CoDeapNEAT [24]	7.3	–	–
LS-Evolution [25]	4.4	40.4M	>2,730 GPU-days
EPSOCNN (Best classification accuracy)	3.58	6.74M	<4 GPU-days
EPSOCNN (10 runs)	3.74±0.0154	4.79M±1.5363M	<4 GPU-days

Method	CIFAR-100	SVHN
Network in Network [29]	35.68	2.35
Deeply Supervised Net [28]	34.57	1.92
FractalNet [27]	23.30	2.01
Wide ResNet [30]	22.07	1.85
ResNet [33]	27.22	2.01
DenseNet(k=12) [4]	20.20	1.67
EPSOCNN (Best)	18.56	1.84
EPSOCNN (10 runs)	19.05±0.1874	1.89±0.0387

GP-based Evolutionary Deep Learning

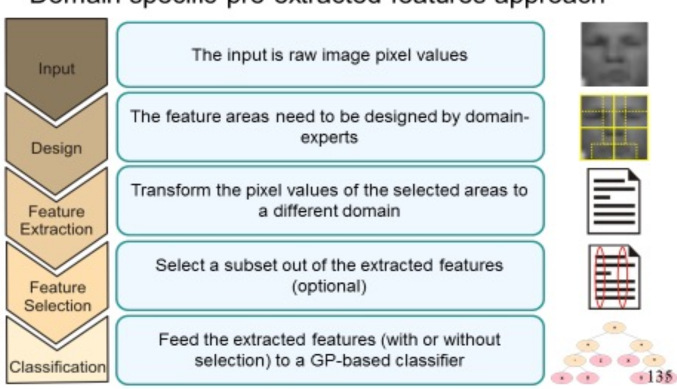
- 3-Tier/2-Tier GP for image classification [2012, 2013]
- GP-HoG [2015-16]
- MLGP [2017]
- ConvGP [2017]
- GP-Criptor – (Deep) Transfer Learning [2014-16]

3-Tier/2-Tier GP

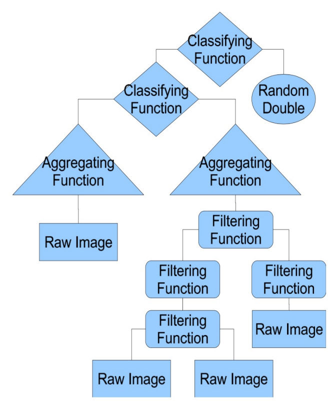
GP for Image Recognition/Classification

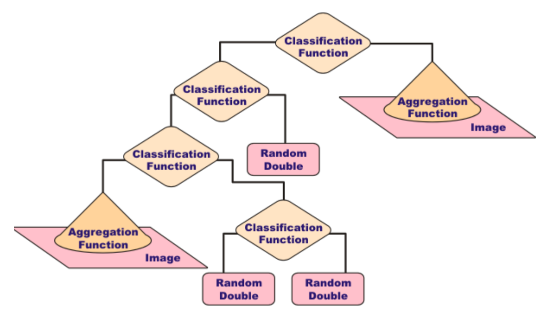
The traditional way

Domain-specific pre-extracted features approach



3-Tier/2-Tier GP

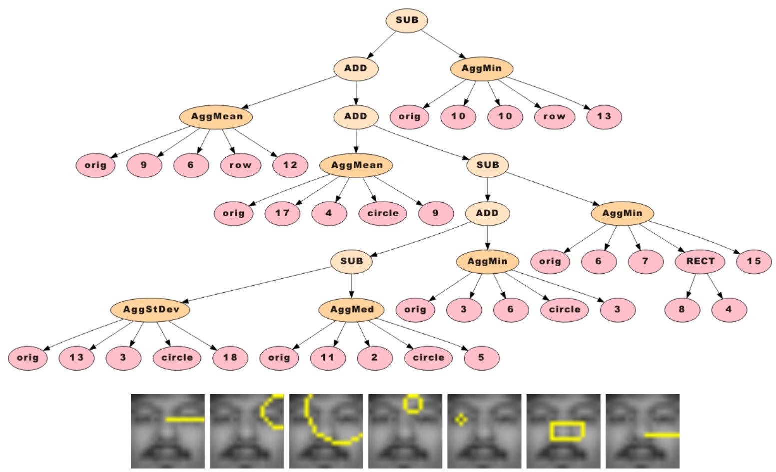




Daniel Atkins, Kourosh Neshatian and Mengjie Zhang. "A Domain Independent Genetic Programming Approach to Automatic Feature Extraction for Image Classification". Proceeding of the 2011 IEEE Congress on Evolutionary Computation. IEEE Press. New Orleans, USA. June 5-8, 2011. pp. 238-245.

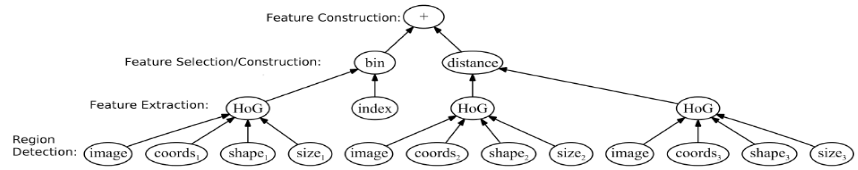
Harith Al-Sahaf, Andy Song, Kourosh Neshatian, Mengjie Zhang. "Two-Tier Genetic Programming Towards Raw Pixel Based Image Classification". *Expert Systems With Applications*. **Vol. 39**, Issue 16. 2012. pp. 12291-12301

2-Tier GP (2012)



GP-HoG [2015-16]

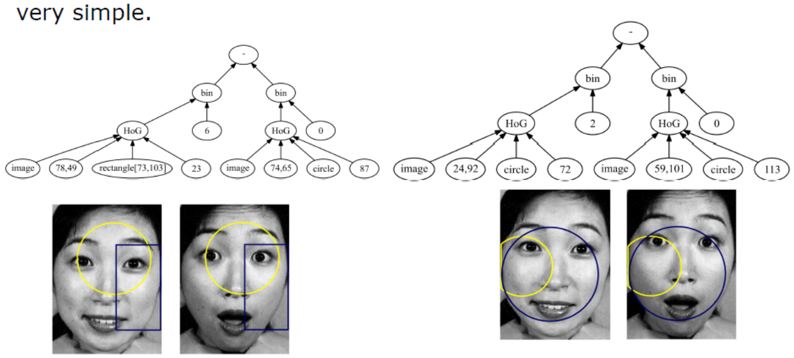
- GP-HoG uses strongly typed GP to perform three tasks in the same tree structure.
- All layers are trained simultaneously and coherently.
- Output of the tree is thresholded.



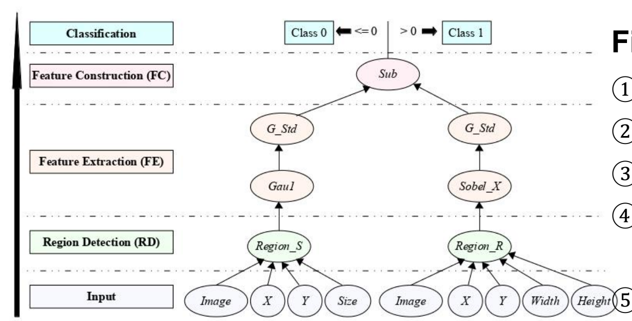
Andrew Lensen, Harith Al-Sahaf, Mengjie Zhang, Bing Xue. "Genetic Programming for Region Detection, Feature Extraction, Feature Construction and Classification in Image Data". Proceedings of the 19th European Conference on Genetic Programming (EuroGP 2016). Lecture Notes in Computer Science. Vol. 9594. Porto, Portugal. March 30 - April 1, 2016. pp. 51-67

GP-HoG

- The below tree has 98% training and 95% test performance on the Jaffe dataset despite being very simple.
- The below tree has 95% training and 100% test performance on the Jaffe dataset.

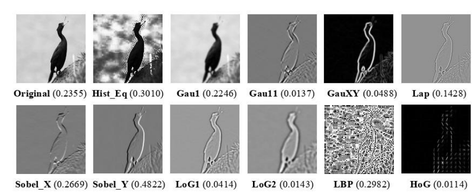


MLGP: *An Automatic Feature Extraction Approach to Image Classification Using Genetic Programming*



Five layers:

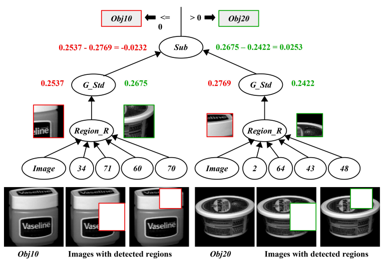
1. Input layer
2. Region detection layer
3. Feature extraction layer
4. Feature construction layer
5. Classification layer



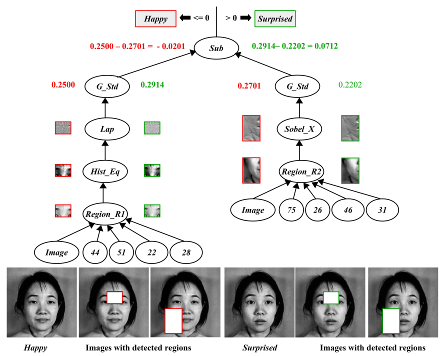
Ying Bi, Bing Xue, Mengjie Zhang. An Automatic Feature Extraction Approach to Image Classification Using Genetic Programming. Proceeding of the 21th Europen Conference on Applications of Evolutionary Computation (EvoApplications 2018). Lecture Notes in Computer Science. Parma, Italy. 4-6 April 2018. pp. 421-438.

Example Solutions

- An example solution on object images

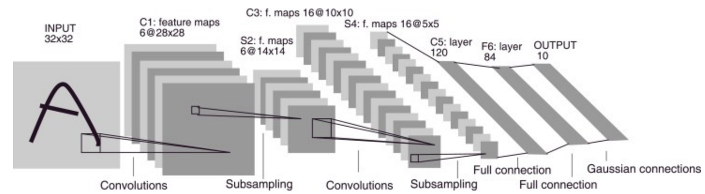


- An example solution on face images



COGP: *An Evolutionary Deep Learning Approach Using Genetic Programming with Convolution Operators for Image Classification*

Convolutional neural network

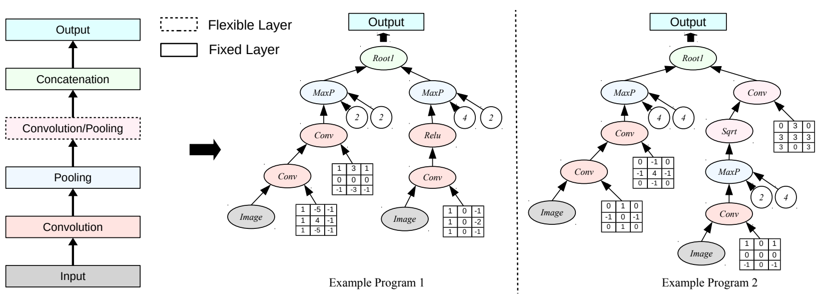


- **Convolution**
- **Subsampling/Pooling**

➡ Nodes of GP trees

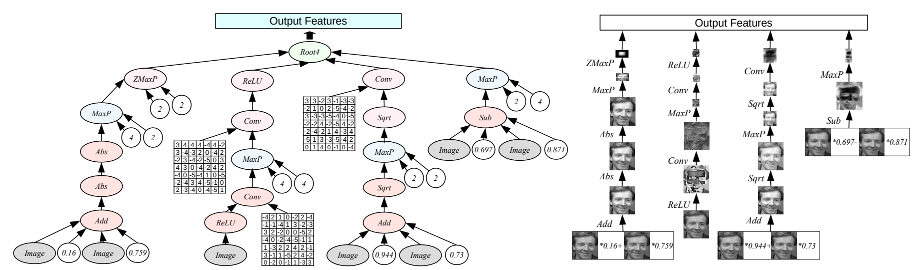
Ying Bi, Bing Xue, Mengjie Zhang. An Evolutionary Deep Learning Approach Using Genetic Programming with Convolution Operators for Image Classification. Proceedings of 2019 IEEE Congress on Evolutionary Computation (CEC), IEEE Precess. Wellington, New Zealand, 10-13 June, 2019, pp. 3197-3204.

The COGP Method

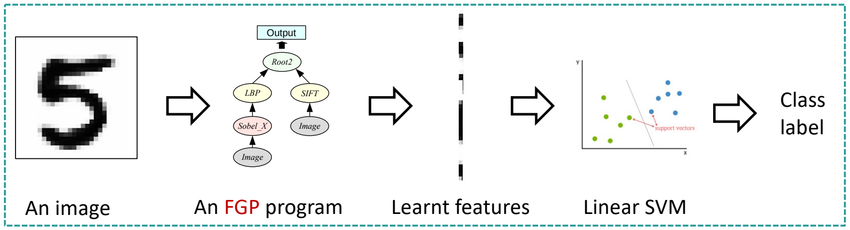


1. Input layer
2. Convolution layer
3. Pooling layer
4. Concatenationg layer
5. Output layer

Example Solution on Face Data Set



FGP: *Genetic Programming with A Flexible Program Structure and Image-Related Operators for Feature Learnign to Image Classification*

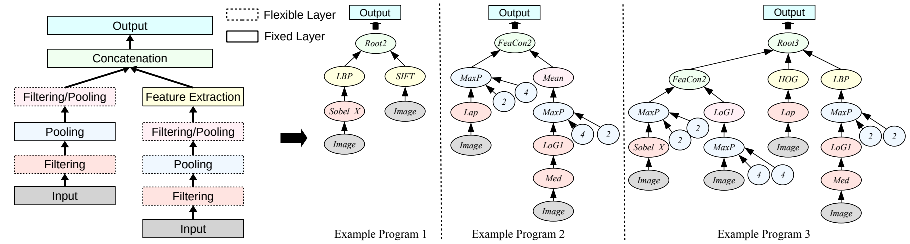


- The complexity of the FGP solutions for different tasks can be various
- The FGP method can learn various types and numbers of effective features from raw images
- FGP can be easily applied to different types of image classification tasks to achieve good classification performance
- The evolved solutions of FGP can be easily visualised, which provide more insights on the tasks

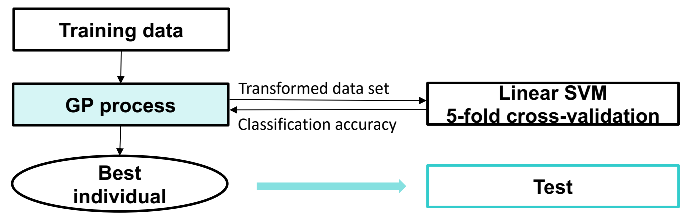
Ying Bi, Bing Xue, Mengjie Zhang. Genetic Programming with A Flexible Program Structure and Image-Related Operators for Feature Learning to Image Classification, Submitted to IEEE Transactions on Evolutionary Computation. 2020.

The FGP Method

Representation



Outline



Experimental Results

Classification error rates of the proposed FGP method

Methods	FEI_1 Min	FEI_1 Mean±St.dev	FEI_2 Min	FEI_2 Mean± St.dev	ORL Min	ORL Mean± St.dev
SVM	10.00	10.00±0.00+	12.00	12.00±0.00+	5.62	5.62±0.00+
KNN	68.00	68.00±0.00+	92.00	92.00±0.00+	5.62	5.62±0.00+
LR	8.00	8.00±0.00+	12.00	12.00±0.00+	6.25	6.25±0.00+
RF	**2.00**	**2.93±1.01**–	10.00	10.80±1.13+	6.88	7.67±0.63+
AdaBoost	20.00	21.33±1.32+	20.00	24.00±3.44+	40.62	47.73±4.00+
ERF	6.00	6.73±0.98+	8.00	9.40±0.93+	2.50	3.29±0.59+
LBP+SVM	34.00	43.27±3.66+	32.00	37.47±3.52+	12.50	12.58±0.21+
HOG+SVM	4.00	4.00±0.00–	18.00	18.00±0.00+	8.75	8.75±0.00+
SIFT+SVM	44.00	44.00±0.00+	38.00	38.00±0.00+	6.25	6.25±0.00+
CNN-5	**2.00**	4.60±1.30=	**2.00**	**4.73±1.62**–	3.12	4.71±1.06+
CNN-8	**2.00**	4.67±1.32=	4.00	9.07±1.87=	5.00	6.96±1.09+
FGP	**2.00**	5.53±2.67	4.00	8.67±3.36	**0.00**	**1.37±1.04**
Overall		7+, 2=, 2–		9+, 1=, 1–		11+

Methods	KTH Min	KTH Mean± St.dev	FS Min	FS Mean± St.dev
SVM	53.03	55.41±2.83+	79.37	79.71±0.15+
KNN	65.76	65.76±0.00+	75.65	75.65±0.00+
LR	51.21	51.21±0.00+	76.51	76.51±0.00+
RF	40.00	42.19±0.83+	62.64	63.47±0.49+
AdaBoost	62.12	66.56±1.37+	82.53	86.96±1.47+
ERF	38.48	40.17±0.86+	62.06	62.85±0.36+
LBP+SVM	21.21	26.71±4.18+	50.21	66.73±8.90+
HOG+SVM	42.73	44.04±0.64+	87.89	92.09±2.47+
SIFT+SVM	34.24	34.24±0.00+	39.08	39.08±0.00+
CNN-5	14.24	17.44±1.87+	49.86	51.97±1.16+
CNN-8	23.64	28.37±3.18+	50.84	53.21±1.01+
FGP	**1.21**	**3.93±1.13**	**25.52**	**29.41±1.74**
Overall		11+		11+

Experimental Results

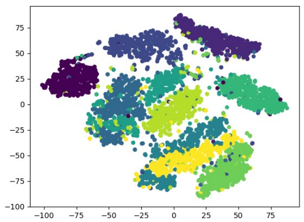
Classification error rates of the proposed FGP method

Methods	**MB**	**MRD**	**MBR**	**MBI**	**Rectangle**	**RI**	**Convex**
SVM+RBF [30]	3.03(+)	11.11(+)	14.58(+)	22.61(+)	2.15 (+)	24.04(+)	19.13(+)
SVM+Poly [30]	3.69(+)	15.42(+)	16.62(+)	24.01(+)	2.15(+)	24.05(+)	19.82(+)
SAE-3 [29]	3.46(+)	10.30(+)	11.28(+)	23.00(+)	2.14(+)	24.05(+)	–
DAE-b-3 [29]	2.84(+)	9.53(+)	10.30(+)	16.68(+)	1.99(+)	21.59(+)	–
CAE-2 [29]	2.48(+)	9.66(+)	10.90(+)	15.50(+)	1.21(+)	21.54(+)	–
SPAE [44]	3.32(+)	10.26(+)	9.01(+)	13.24(+)	–	–	–
RBM-3 [29]	3.11(+)	10.30(+)	6.73(+)	16.31(+)	2.60(+)	22.50(+)	–
ScatNet-2 [27, 28]	1.27(+)	7.48(+)	12.30(+)	18.40(+)	0.01(+)	8.02(+)	6.50(+)
RandNet-2 [28]	1.25(+)	8.47(+)	13.47(+)	11.65(+)	0.09(+)	17.00(+)	5.45(+)
PCANet-2(softmax) [28]	1.40(+)	8.52(+)	6.85(+)	11.55(+)	0.49(+)	13.39(+)	4.19(+)
LDANet-2 [28]	**1.05**	7.52(+)	6.81(+)	12.42(+)	0.14(+)	16.20(+)	7.22(+)
NNet [30]	4.69(+)	18.11(+)	20.04(+)	27.41(+)	7.16(+)	33.20(+)	32.25(+)
SAA-3 [30]	3.46(+)	10.30(+)	11.28(+)	23.00(+)	2.41(+)	24.05(+)	18.41(+)
DBN-3 [30]	3.11(+)	10.30(+)	6.73(+)	16.31(+)	2.60(+)	22.50(+)	18.63(+)
FCCNN [25]	2.43(+)	8.91(+)	6.45	13.23(+)	–	–	–
FCCNN (with BT) [25]	2.68(+)	9.59(+)	6.97(+)	10.80(+)	–	–	–
SPCN [26]	1.82(+)	9.81(+)	**5.84**	9.55(+)	0.19(+)	10.60(+)	–
FGP(best)	1.18	**7.37**	6.54	**7.48**	**0.00**	**6.10**	**1.54**
FGP(mean)	1.30	8.44	7.34	10.35	0.12	7.34	1.84
FGP(std)	0.06	0.6	0.42	1.41	0.11	0.61	0.19
Rank	2/18	**1/18**	3/18	**1/18**	**1/15**	**1/15**	**1/10**

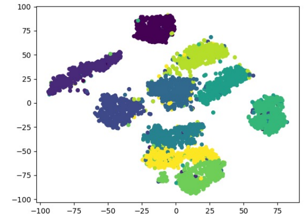
Data Visualisation Using the Learnt Features

- t-distributed Stochastic Neighbor Embedding (t-SNE)

10 classes of digits

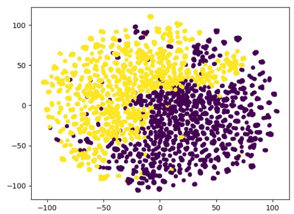


MB Original Data

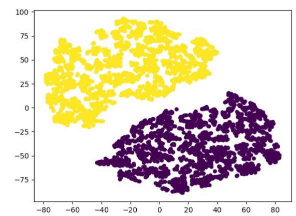


Transformed Data by FGP

2 classes of rectangles



Rectangle Original Data



Transformed Data by FGP

IEGP: *Genetic Programming with A New Representation to Automatically Learn Features and Evolve Ensembles for Image Classification*

Traditional Ensemble Methods for Image Classification

Raw Images → Feature Extraction → Base Classifiers → Class labels

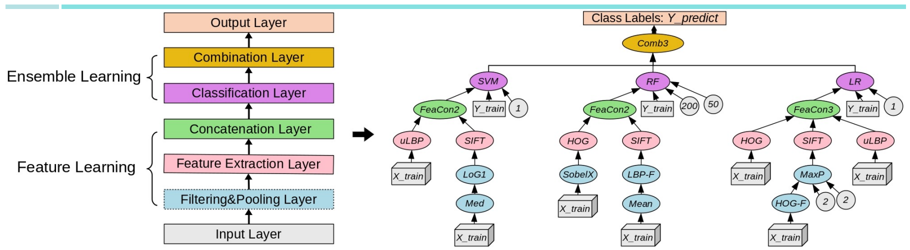
The New Approach for Image Classification

Raw Images → An IEGP solution → Class labels

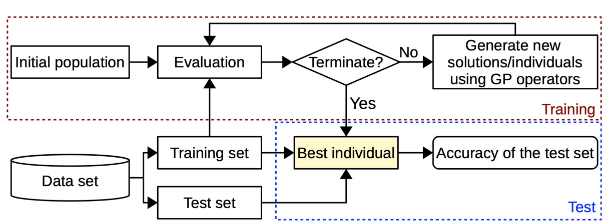
- A new multi-layer individual representation is developed in IEGP to allow it to automatically and simultaneously learn features and evolve ensembles for image classification
- IEGP can learn high-level features through multiple transformations
- IEGP can automatically select and optimise the parameters for the classification algorithms in the evolved ensemble
- IEGP can automatically address the diversity issue when building the ensembles

Ying Bi, Bing Xue, Mengjie Zhang. "Genetic Programming with A New Representation to Automatically Learn Features and Evolve Ensembles for Image Classification". IEEE Transactions on Cybernetics. DOI:10.1109/TCYB.2020.2964566. 15pp. First Online on 30 January 2020

Multi-Layer Representation of IEGP

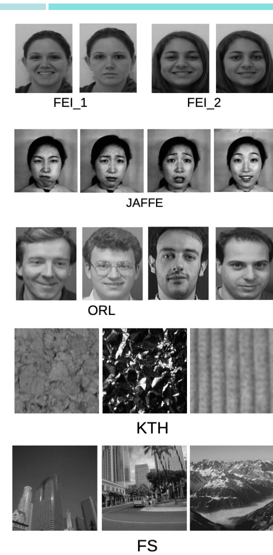


Overall Algorithm



Experimental Results

Classification accuracy of the proposed IEGP method



Method	FEI_1		FEI_2		JAFFE	
	Max	Mean±St.dev	Max	Mean±St.dev	Max	Mean±St.dev
SVM	90.00	90.00±0.00+	88.00	88.00±0.00+	93.94	**91.06±0.73**–
KNN	32.00	32.00±0.00+	8.00	8.00±0.00+	71.21	71.21±0.00+
LR	92.00	92.00±0.00+	88.00	88.00±0.00+	89.39	89.39±0.00–
RF	98.00	**97.07±1.01**=	90.00	89.20±1.13+	75.76	72.48±1.99+
AdaBoost	80.00	78.67±1.32+	80.00	76.00±3.44+	53.03	47.93±2.68+
ERF	94.00	93.27±0.98+	92.00	90.60±0.93+	77.27	73.89±1.72+
uLBP+SVM	66.00	56.73±3.66+	68.00	62.53±3.52+	31.82	26.87±3.30+
LBP+SVM	68.00	64.60±1.83+	74.00	69.80±0.00+	33.33	28.84±2.05+
HOG+SVM	96.00	96.00±0.00=	82.00	82.00±0.00+	81.82	80.30±0.40+
SIFT+SVM	56.00	56.00±0.00+	62.00	62.00±0.00+	33.33	33.33±0.00+
CNN-5	98.00	95.40±1.30+	98.00	95.27±1.62+	**95.45**	90.96±2.68–
CNN-8	98.00	95.33±1.32+	96.00	90.93±1.87+	90.91	84.54±4.33=
EGP	**100.0**	96.20±2.06=	**100.0**	**98.07±1.70**=	92.42	84.24±4.28=
IEGP	**100.0**	96.67±2.55	**100.0**	96.20±3.66	92.42	82.17±5.42
Overall		10+, 3=		12+, 1=		8+, 2=, 3–

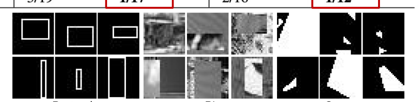
Method	ORL		KTH		FS	
	Max	Mean±St.dev	Max	Mean±St.dev	Max	Mean±St.dev
SVM	94.38	94.38±0.00+	46.97	44.59±2.83+	20.63	20.30±0.15+
KNN	94.38	94.38±0.00+	34.24	34.24±0.00+	24.35	24.35±0.00+
LR	93.75	93.75±0.00+	48.79	48.79±0.00+	23.49	23.49±0.00+
RF	93.12	92.33±0.63+	60.00	57.81±0.83+	37.36	36.53±0.49+
AdaBoost	59.38	52.27±4.00+	37.88	33.44±1.37+	17.47	13.04±1.47+
ERF	97.50	96.71±0.59+	61.52	59.83±0.86+	37.94	37.15±0.36+
uLBP+SVM	87.50	87.42±0.21+	78.79	73.29±4.18+	49.79	33.27±8.90+
LBP+SVM	88.12	87.52±0.20+	83.64	82.71±0.51+	53.50	50.45±1.80+
HOG+SVM	91.25	91.25±0.00+	57.27	55.96±0.64+	12.11	7.91±2.47+
SIFT+SVM	93.75	93.75±0.00+	65.76	65.76±0.00+	60.92	60.92±0.00+
CNN-5	96.88	95.29±1.06+	85.76	82.56±1.87+	50.14	48.03±1.16+
CNN-8	95.00	93.04±1.09+	76.36	71.63±3.18+	49.16	46.79±1.01+
EGP	99.38	97.44±1.26+	87.88	77.53±5.17+	67.17	61.07±2.91+
IEGP	**100.0**	**98.29±0.97**	**98.48**	**96.43±1.26**	**92.54**	**89.63±1.47**
Overall		13+		13+		13+

Experimental Results

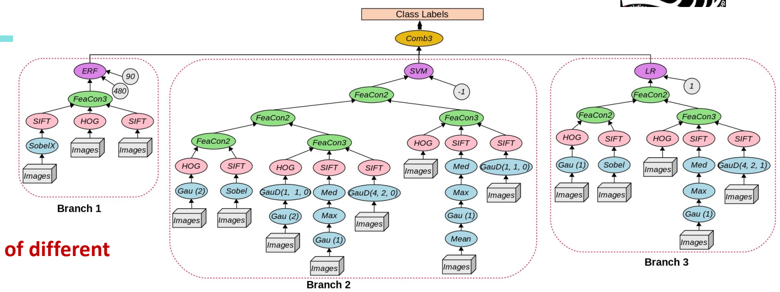
Classification accuracy of the proposed IEGP method

Method	MB	MRD	MBR	MBI	Rectangle	RI	Convex
SVM+RBF [51]	96.97(+)	88.89(+)	85.42(+)	77.39(+)	97.85(+)	75.96(+)	80.87(+)
SVM+Poly [51]	96.31(+)	84.58(+)	83.38(+)	75.99(+)	97.85(+)	75.95(+)	80.18(+)
SAE-3 [36]	96.54(+)	89.70(+)	88.72(+)	77.00(+)	97.86(+)	75.95(+)	–
DAE-b-3 [36]	97.16(+)	90.47(+)	89.70(+)	83.32(+)	98.01(+)	78.41(+)	–
CAE-2 [36]	97.52(+)	90.34(+)	89.10(+)	84.50(+)	98.79(+)	78.46(+)	–
SPAE [52]	96.68(+)	89.74(+)	90.99(+)	86.76(+)	–	–	–
RBM-3 [36]	96.89(+)	89.70(+)	93.27(+)	83.69(+)	97.40(+)	77.50(+)	–
ScatNet-2 [33, 34]	98.73(+)	92.52(+)	87.70(+)	81.60(+)	99.99(+)	91.98(+)	93.50(+)
RandNet-2 [34]	98.75(+)	91.53(+)	86.53(+)	88.35(+)	99.91(+)	83.00(+)	94.55(+)
PCANet-2 (softmax) [34]	98.60(+)	91.48(+)	93.15(+)	88.45(+)	99.51(+)	86.61(+)	95.81(+)
LDANet-2 [34]	98.95	92.48(+)	93.19(+)	87.58(+)	99.86(+)	83.80(+)	92.78(+)
NNet [51]	95.31(+)	81.89(+)	79.96(+)	72.59(+)	92.84(+)	66.80(+)	67.75(+)
SAA-3 [51]	96.54(+)	89.70(+)	88.72(+)	77.00(+)	97.59(+)	75.95(+)	81.59(+)
DBN-3 [51]	96.89(+)	89.70(+)	93.27(+)	83.69(+)	97.40(+)	77.50(+)	81.37(+)
FCCNN [35]	97.57(+)	91.09(+)	93.55(+)	86.77(+)	–	–	–
FCCNN (with BT) [35]	97.32(+)	90.41(+)	93.03(+)	89.20(+)	–	–	–
SPCN [32]	98.18(+)	90.19(+)	94.16	90.45	99.81(+)	89.40(+)	–
EvoCNN (best) [53]	98.82	94.78	97.20	95.47	99.99(+)	94.97	95.18(+)
EGP (best) [26]	97.19(+)	–	–	–	99.91(+)	–	93.97(+)
IEGP (best)	98.82	94.28	93.59	89.41	**100**	94.88	**98.26**
IEGP (mean)	98.69	93.78	92.65	88.42	99.94	89.02	97.76
IEGP (std)	0.08	0.24	0.35	0.64	0.05	2.1	0.26
Rank	2/20	2/19	3/19	3/19	**1/17**	2/16	**1/12**


0 4 1 6 5 9 7 8

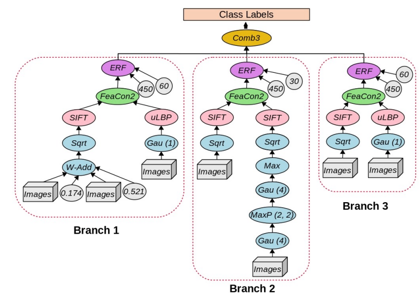

Rectangle RI Convex

Example Solutions



1

Ensemble of different classifiers



2

Ensemble of ensembles

Summary

- NN-based evolutionary deep learning has started to demonstrate great potential to outperform the manually designed state-of-the-art deep networks in image classification and analysis
- GP based evolutionary deep learning has also started, and is expected to demonstrate the advantages in effectiveness, efficiency and interpretability in image analysis
- Evolutionary deep learning is still in an early stage, but is expected to show the great accuracy, efficiency, small training set, and good interpretability of the deep models.

Acknowledgement

- Thanks my colleagues and research students particularly Dr Bing Xue, Dr Yanan Sun, Ying Bi, Bin Wang, Dr Harith Al-Sahaf, Dr Andrew Lensen, Ben Evans, and other members in our ECRG Group.
- Thanks GECCO2020 organisers
- Funding Agents:
  1. Marsden Fund of New Zealand award number(s): VUW 1509, VUW 1615, RTVU1914
- 2. University Research Fund at Victoria University of Wellington award number(s): 223805/3986

More Recent Group Photo

35 people -- several people are missing!

References

- Yanan Sun, Handing Wang, Bing Xue, Yaochu Jin, Gary Yen, Mengjie Zhang. "Surrogate-Assisted Evolutionary Deep Learning Using an End-to-End Random Forest-based Performance Predictor". IEEE Transactions on Evolutionary Computation. DOI: 10.1109/TEVC.2019.2924461. 2019.
- Yanan Sun, Bing Xue, Mengjie Zhang, Gary Yen. "Completely Automated CNN Architecture Design Based on Blocks". IEEE Transactions on Neural Networks and Learning Systems. 2019. DOI: 10.1109/TNNLS.2019.2919608. First Online on 20 June 2019.
- Yanan Sun, Bing Xue, Mengjie Zhang, Gary Yen. "Evolving Deep Convolutional Neural Networks for Image Classification". IEEE Transactions on Evolutionary Computation. 2019. DOI: 10.1109/TEVC.2019.2916183.
- Yanan Sun, Gary G. Yen, Zhang Yi, "Evolving Unsupervised Deep Neural Networks for Learning Meaningful Representations". IEEE Transactions on Evolutionary Computation. DOI:10.1109/TEVC.2018.2808689.
- Bing Xue, Mengjie Zhang, Will N. Browne: Particle Swarm Optimization for Feature Selection in Classification: A Multi-Objective Approach. IEEE T. Cybernetics 43(6): 1656-1671 (2013)
- Harith Al-Sahaf, Ying Bi, Qi Chen, Andrew Lensen, Yi Mei, Yanan Sun, Binh Tran, Bing Xue*, and Mengjie Zhang*. "A Survey on Evolutionary Machine Learning". Journal of the Royal Society of New Zealand (TNZR). 2019. pp. 205-228. DoI: 10.1080/03036758.2019.1609052.
- Yanan Sun, Bing Xue, Mengjie Zhang, Gary Yen. "A Particle Swarm Optimization-based Flexible Convolutional Auto-Encoder for Image Classification". IEEE Transactions on Neural Networks and Learning Systems. Volume 30, Issue 8, 2019. pp. 2295 - 2309. DOI: 10.1109/TNNLS.2018.2881143

References

- Yanan Sun, Bing Xue, Mengjie Zhang, Gary G. Yen, An Experimental Study on Hyper-parameter Optimization for Stacked Auto-Encoders, 2018 IEEE Congress on Evolutionary Computation (CEC). Accepted.
- Bing Xue, Mengjie Zhang, Will Browne, Xin Yao. "A Survey on Evolutionary Computation Approaches to Feature Selection", IEEE Transaction on Evolutionary Computation, vol. 20, no. 4, pp. 606-626, Aug. 2016. doi: 10.1109/TEVC.2015.2504420.
- Stefano Cagnoni, GECCO 2008 and GECCO 2014 Tutorial on IASP
- Will Smart, Mengjie Zhang. "Tracking Object Positions in Real-time Video using Genetic Programming". In *Proceeding of Image and Vision Computing International Conference,* 2004. pp. 113-118.
- Yanan Sun, Bing Xue, Mengjie Zhang, Gary G. Yen, Jiancheng Lv. "Automatically Designing CNN Architectures Using the Genetic Algorithm for Image Classification". IEEE Transactions on Cybernetics. DOI: 10.1109/TCYB.2020.2983860. April 2020.
- Bin Wang, Bing Xue, Mengjie Zhang. "Particle Swarm Optimisation for Evolving Deep Convolutional Neural Networks for Image Classification: Single- and Multi-objective Approaches". Deep Neural Evolution. Natural Computing Series, DOI: 10.1007/978-981-15-3685-4_6 . 2020.
- Bin Wang, Bing Xue, Mengjie Zhang and Yanan Sun. "A Hybrid GA-PSO Method for Evolving Architecture and Short Connections of Deep Convolutional Neural Networks". Proceedings of the 16th Pacific Rim International Conference on Artificial Intelligence, Part III (PRICAI(3) 2019). Lecture Notes in Computer Science. Vol. 11672, Cuvu, Fiji. August 26-30, 2019. pp.650-663.

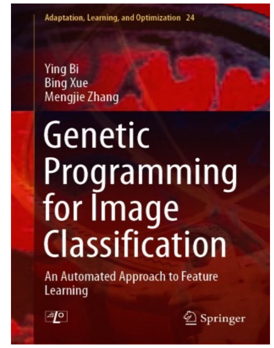
References

- Andy Song, Mengjie Zhang: Genetic programming for detecting target motions. Connect. Sci. 24(2-3): 117-141 (2012)
- Harith Al-Sahaf, Andy Song, Kourosh Neshatian, Mengjie Zhang: Two-Tier genetic programming: towards raw pixel-based image classification. Expert Syst. Appl. 39(16): 12291-12301 (2012)
- Harith Al-Sahaf, Andy Song, Kourosh Neshatian, Mengjie Zhang: Extracting image features for classification by two-tier genetic programming. IEEE Congress on Evolutionary Computation 2012: 1-8
- Ying Bi, Bing Xue, Mengjie Zhang. An Automatic Feature Extraction Approach to Image Classification Using Genetic Programming. Proceeding of the 21th Europen Conference on Applications of Evolutionary Computation (EvoApplications 2018). Lecture Notes in Computer Science. Parma, Italy. 4-6 April 2018. pp. 421-438.
- Ying Bi, Bing Xue, Mengjie Zhang. An Evolutionary Deep Learning Approach Using Genetic Programming with Convolution Operators for Image Classification. Proceedings of 2019 IEEE Congress on Evolutionary Computation (CEC), IEEE Precess. Wellington, New Zealand, 10-13 June, 2019, pp. 3197-3204.
- Ying Bi, Bing Xue, Mengjie Zhang. Genetic Programming with A Flexible Program Structure and Image-Related Operators for Feature Learning to Image Classification, Submitted to IEEE Transactions on Evolutionary Computation.
- Ying Bi, Bing Xue, Mengjie Zhang. Genetic Programming with A New Representation to Automatically Learn Features and Evolve Ensembles for Image Classification. IEEE Transactions on Cybernetics. 2020. DOI:10.1109/TCYB.2020.2964566.

References

- Bin Wang, Yanan Sun, Bing Xue and Mengjie Zhang. "A Hybrid Differential Evolution Approach to Designing Deep Convolutional Neural Networks for Image Classification". *Proceedings of the 31st Australasian Joint Conference on Artificial Intelligence (AI 2018). Lecture Notes in Computer Science.*. **Vol. 11320**. Wellington, New Zealand. 11-14 December 2018. pp. 237-250.
- Bin Wang, Yanan Sun, Bing Xue, Mengjie Zhang. "Evolving Deep Neural Networks by Multi-objective Particle Swarm Optimization for Image Classification". Proceedings of 2019 Genetic and Evolutionary Computation Conference (GECCO 2019). Prague, Czech Republic. 13-17 July 2019. pp. 490-498.
- Bin Wang, Yanan Sun, Bing Xue and Mengjie Zhang. "Evolving Deep Convolutional Neural Networks by Variable-length Particle Swarm Optimization for Image Classification". *Proceedings of 2018 IEEE Congress on Evolutionary Computation*. IEEE Press. Rio de Janeiro, Brazil. 8-13 July 2018. pp. 1514-1521
- Ying Bi, Bing Xue, and Mengjie Zhang. "An Effective Feature Learning Approach Using Genetic Programming with Image Descriptors for Image Classification". IEEE Computational Intelligence Magazine. Vol. 15, Issue 2, 2020. pp. 65-77. DOI: 10.1109/MCI.2020.2976186.
- Ying Bi, Bing Xue, and Mengjie Zhang. “Genetic Programming for Image Classification: An Automated Approach to Feature Learning”, Springer International Publishing 2021, XXVIII, 258pages, DOI: https:doi.org10.1007978-3-030-65927-1.

Genetic Programming for Image Classification: An Automated Approach to Feature Learning --New Book



1. Provide a throughout **background** of evolutionary computation, computer vision and machine learning
2. Introduces a series of typical **GP-based approaches** to feature learning in image classification
3. Provides broad perceptive insights on what and how GP can offer and shows a **comprehensive and systematic research route** in this field
4. Discusses the use of different techniques in GP to improve the **generalization performance** and/or **computational efficiency** for image classification

- ***The first book on GP for image classfication***
- ***Prof. Wolfgang Banzhaf wrote the foreword***
- ***All the codes are released at: https://github.com/YingBi92/BookCode***

Ying Bi, Bing Xue, and Mengjie Zhang. “Genetic Programming for Image Classification: An Automated Approach to Feature Learning”, *Springer International Publishing 2021*, XXVIII, 258pages, DOI: https:doi.org10.1007978-3-030-65927-1.

Part II

From the EC attic…
Re-using (and renovating) forgotten tools

SUB-MACHINE CODE GENETIC PROGRAMMING (Poli, Langdon 1998)

Inputs: Unsigned long (32 or 64 bit words) that encode arrays of binary inputs. The bit string may encode consecutive samples of a temporal sequence, a row or a window within an image, etc.

Function set: bitwise logical operators + circular shifts

A whole block of data is affected by a single bit-wise Boolean operation (SIMD paradigm).

Output: a 32/64 bit string, it may represent 32/64 possible outputs of a binary classifier.

So, 32/64 (not independent) solutions are evaluated for each individual.

SUB-MACHINE CODE GENETIC PROGRAMMING (Poli, Langdon 1998)

Advantages

- Several solutions computed at the same time
- Possible multiple-bit outputs representing different classes
- 1-to-1 translation of instructions into machine code
- Extremely fast execution speed
- Possible direct translation into hardware

Basic SmcGP: an example

License-plate Recognition System *(Adorni et al., 2000)*

Main task: car license-plate recognition

Data: 130 images of running cars

Sub-tasks:
plate extraction and character recognition

Traditional Plate Extraction

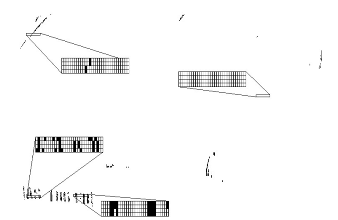
Compute and binarize the image horizontal gradient.
The plate region is the one where the density of vertical edges (peaks of the horizontal gradient) is highest

SmcGP-based Plate Detection

Input data: 4 unsigned long words encoding a window, of size 32x4 pixels, from the binarized gradient image

Desired output: 1 if the window belongs to the plate
(left uppermost pixel) 0 otherwise

Convolving the input window of the GP tree with the whole image the license plate should become a black rectangle.



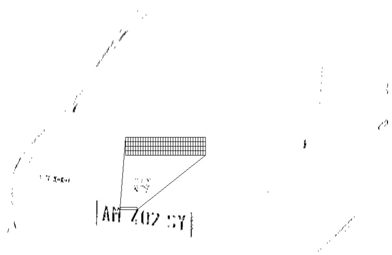
Training Set

80/130 images

Input data: binarized horizontal gradient image

366 negative + 4824 positive = 5190 training samples

Empty samples (that can be found both inside and outside the plate) have been purged.



Function Set + ERCs

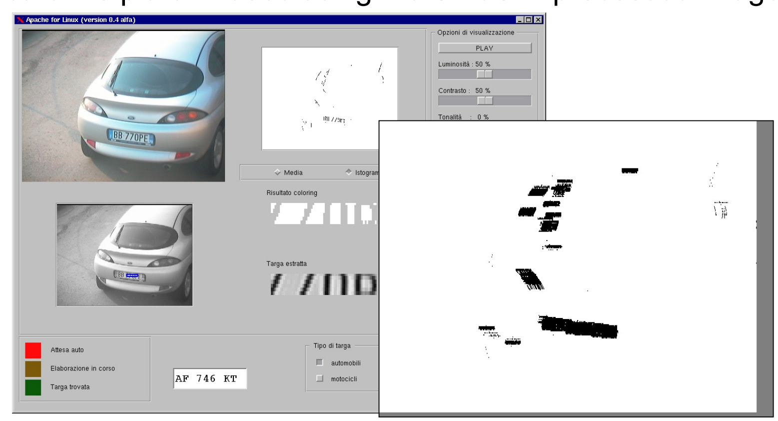
- Binary bitwise operators: AND OR NOT XOR
- Circular shift operators: SHR, SHR2, SHR4, SHL, SHL2, SHL4
- Ephemeral Random Constants (ERC): 32-bit unsigned long

Typical Results

The same algorithm that our reference (manually-designed) plate detection system directly applies to the gradient image can be applied to this image to improve plate localization

Success case

Increased plate-detection rate: the original system could not locate the plate. It does using the SmcGP -processed image



Question:

Why should we unbury this paradigm more than 20 years later?

SUB-MACHINE CODE GENETIC PROGRAMMING (Poli, Langdon 1998)

Q: Why should we unbury this paradigm more than 20 years later?

Answer

Consider what happened with Deep Learning: when neural nets became *en vogue* once again in the late ‘80s, defining most of the principles on which modern DL relies, technology would not be able to support as complex networks and as many data as DL requires. But, nowadays, …

Similarly, 25 years ago, a set of basic classifiers as described before (one per class) was probably the most we could afford to train. But, nowadays, …, perhaps…

SUB-MACHINE CODE GENETIC PROGRAMMING (Poli, Langdon 1998)

Consider that:

- The full output of such a tree/module is itself a complex binary feature that such a module extracts from the input
- Such features have local properties, since usually the value of one bit depends on the values of input bits located in its neighborhood (shift operations are limited to 4 bits)
- As usual in GP, a tree may have a single output and multiple inputs
- The full outputs of several trees can be ‘summarized’ bitwise by some ‘pooling’ operators
- Single bits from different outputs can be selected and ‘re-ordered’ into a new feature

Thinking about deep learning,

doesn’t this ring a bell?

Are we heading towards Sub-machine Code Deep Genetic Programming?

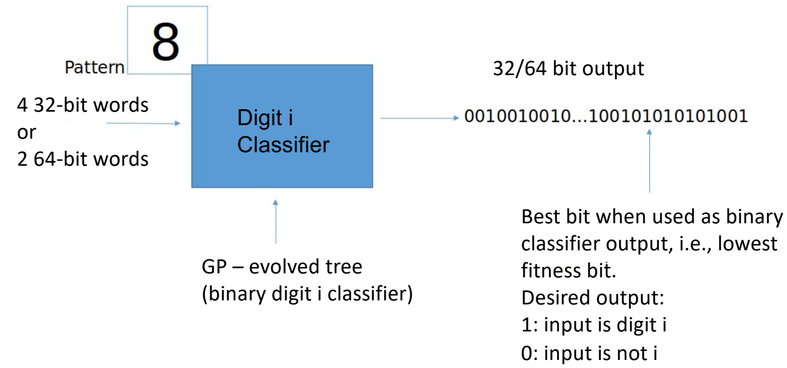
Character Recognition

Recognition of digits represented by binary two-dimensional patterns: 10 specialized binary classifiers have the pattern as input and produce as output:

1 if the input pattern belongs to the class corresponding to the classifier

0 if the pattern belongs to another class

SmcGP-based digit recognition (Cagnoni et al., 2005)



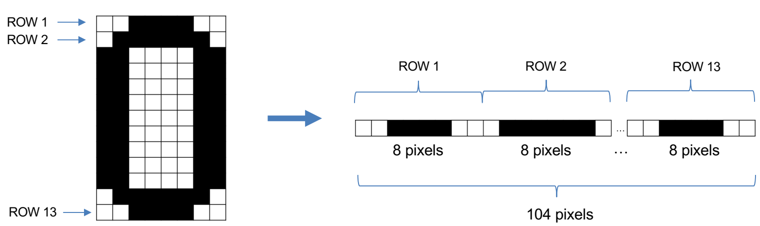
Dataset

Real-world dataset collected by Società Autostrade SpA at highway toll booths

- 11034 binary patterns representing the ten digits from 0 to 9

  6024 in the training set
  5010 in the test set (exactly 501 per class)

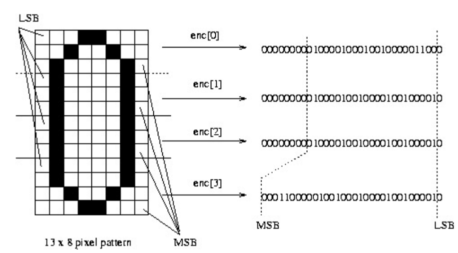
- Size: 13x8 pixels → strings of 104 binary features



Input Encoding (Cagnoni et al., 2005)

Input Pattern: binary digits of size 13x8
104 bits may be represented using 4 32-bit words (e.g., 4 unsigned long variables in C).

72 bits of the pattern are packed into the 24 least significant bits of the first 3 unsigned long variables
The remaining 32 are packed into the fourth one



Fitness Function *(Cagnoni et al., 2005)*

$$F = \sqrt{\frac{FP^2 + FN^2 + K_S * \text{size}}{N_p^2 + N_n^2}}$$

(FP = False Positives, FN = False Negatives, K_s= constant
N_p = Positive cases, N_n = Negative cases, size= tree size)

NB If training data are uniformly distributed, then the negative cases shown to each classifier are 9 times as many as the positive ones

=> F strongly favors specificity

Evolution Parameters *(Cagnoni et al., 2005)*

Population : 1000
Survival rate : 17 %
Crossover rate: 80%
Mutation rate : 3%

Tournament selection with tournament size = 7

300 to 2000 iterations

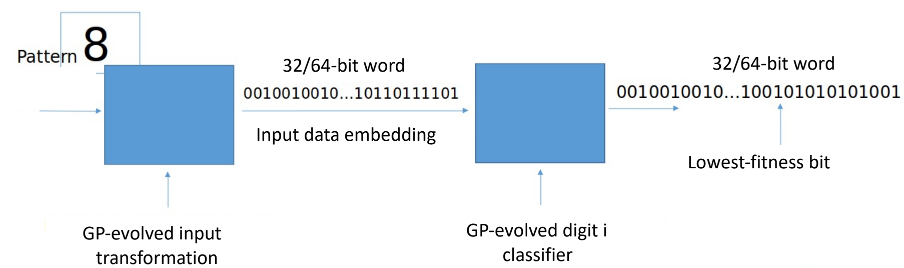
Classifiers *(Cagnoni et al., 2005)*

Set of 10 binary classifiers (one for each class)

For each classifier:
Input : unsigned long pattern[4]
Output : unsigned long out

Each individual (classifier) actually produces 32 binary outputs (32 distinct fitness cases): the highest-fitness bit, and the corresponding fitness, are taken as the individual's output and fitness.

New architecture: SmcGP + Embedding (a tiny step towards Deep GP)

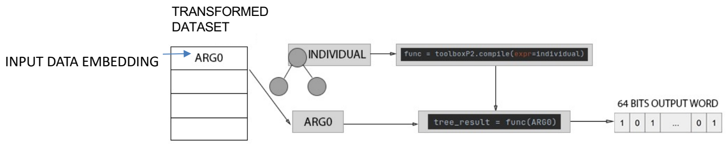


Co-evolutionary algorithm

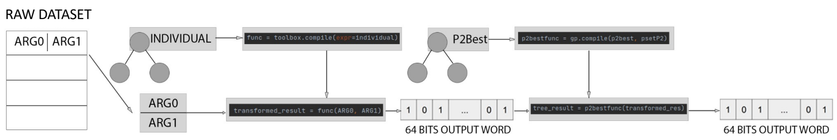
Generate a random initial embedding (P1Best)

Repeat

1. Compute the training set transformed by P1best
2. (*process P2*) Evolve the classifier
3. P2best ← best classifier



4. (*process P1*) Evolve the input data transformation using P2best as a classifier
5. P1best ← best transformation



Until the termination condition is reached

Parameters

Evolution of an embedding/classifier pair for one digit:

- Population: 1000
- Max number of generations: 1000 (20 generations for each GP x 25 iterations)
- Termination condition/ overfitting control: 40 consecutive generations without fitness improvement on the validation set (*if selected*)
- 5 runs
- Evolution parameters same as in the original paper
- Training set: 4218 patterns (*almost balanced*)
- Validation set: 1806 patterns (*almost balanced*)
- Test set: 5010 patterns (501 per digit)

Configurations

- Original SmcGP implementation
- Original SmcGP + overfitting control *(validation set)*
- SmcGP + Embedding
- SmcGP + Embedding + overfitting control

The method has been implemented in Python using DEAP (Distributed Evolutionary Algorithms in Python)

Results

		0		5		6		7		8	
		TNR(%)	TPR(%)	TNR(%)	TPR(%)	TNR(%)	TPR(%)	TNR(%)	TPR(%)	TNR(%)	TPR(%)
Standalone	Mean	99.45	93.30	99.62	88.66	99.13	86.83	99.65	93.25	99.30	89.62
	St. Dev.	0.41	4.77	0.13	1.07	0.57	5.10	0.08	**0.39**	0.29	1.64
	Best	99.78	97.01	99.76	89.42	99.51	**95.61**	99.67	94.01	99.45	**91.62**
	Worst	98.76	85.43	99.38	86.63	98.23	80.64	99.49	93.01	99.38	86.63
Standalone + Overfitting Control	Mean	99.25	88.70	99.29	87.90	98.64	83.91	99.60	93.90	99.41	86.35
	St. Dev.	0.45	8.98	0.21	1.32	0.38	4.48	0.14	0.43	0.14	**0.96**
	Best	99.71	**97.41**	99.27	90.02	98.71	92.22	99.67	94.41	99.60	88.02
	Worst	98.78	88.02	99.07	86.03	97.98	79.04	99.31	**93.21**	99.33	85.63
Embedding	Mean	99.26	91.89	99.61	89.30	98.97	85.75	99.61	93.61	99.34	89.14
	St. Dev.	0.37	4.11	**0.09**	1.62	0.34	3.00	**0.07**	1.42	0.09	1.18
	Best	99.76	96.21	99.56	91.82	99.47	90.62	99.67	95.21	99.51	91.02
	Worst	99.05	86.03	**99.49**	87.62	98.54	82.63	99.53	91.62	99.38	87.62
Embedding + Overfitting Control	Mean	**99.82**	**96.41**	**99.65**	**90.86**	**99.40**	**92.41**	**99.66**	**94.61**	**99.57**	**89.74**
	St. Dev.	**0.07**	**1.06**	0.22	**0.94**	**0.15**	**0.43**	0.15	1.15	**0.08**	1.07
	Best	**99.93**	97.21	**99.84**	**92.02**	**99.69**	93.01	**99.89**	**96.21**	**99.65**	91.22
	Worst	**99.87**	**95.81**	99.20	**89.62**	**99.45**	**91.82**	**99.60**	92.61	**99.58**	**88.02**

Remarks

Although results are preliminary and obtained on only a few runs, some clear indications are:

- Introducing only the embedding has apparently a negative effect: most probably, since it simplifies the input representation, it favors overfitting.
- Adding embedding + overfitting control produces results that are generally better than all other configurations
- Without embedding, introducing overfitting control slightly worsens accuracy (termination threshold may need to be changed), which reinforces the above hypothesis about embedding and overfitting

Concluding remarks

- Although we are still far from Deep (and even from fully Multi-layer) GP, it seems possible to create hierarchical architectures using SmcGP modules as the building blocks.
- Similar ideas could be applied to more traditional GP trees with continuous inputs and outputs
- In this latter case, if all the GP functions in the function set were differentiable, it would be possible to associate weights to the connections between GP-evolved modules and use backpropagation or other gradient-based algorithms at some point.

Future Work

- Completing the classifier by adding a full 'layer' with 10 embeddings, one per digit
- Considering adding other elements between the embedding layer and the classification layer, having all 10 embeddings as inputs
  - A feature selector/pooler layer ?
  - A further embedding layer ?
  - Something else ?…
- In a sequential layer-by-layer co-evolution as described, but with n>2 layers, is it better to propagate evolution forward? Backwards? alternating forward and backward steps?

Credits

- The work described has been developed as their B.Eng. theses by Fabrizio De Santis and Dario Cavalli
- Many thanks also to Federico Sello, B.Eng., Andrea Bettati, M.Eng, and Marco Carraglia, M.Eng, for contributing to the development of SmcGP in DEAP.

References

- Adorni G., Bergenti F., Cagnoni S., Mordonini M. (2000) License-Plate Recognition for Restricted-Access Area Control. In: Foresti G.L., Mähönen P., Regazzoni C.S. (eds) Multimedia Video-Based Surveillance Systems. The Springer International Series in Engineering and Computer Science, vol 573. Springer, Boston, MA.
- Cagnoni S., Bergenti F., Mordonini M., Adorni G., "Evolving binary classifiers through parallel computation of multiple fitness cases". *IEEE Transactions on Systems, Man and Cybernetics Part B-Cybernetics*, 35(3):548-555, 2005.

Summary

Concluding Remarks

- Evolutionary computer vision and image analysis is still a big and hot topic
  - Evolutionary deep learning will play a significant role
  - GP-based deep learning will have more developments
  - Interpretability and expandability will be a major focus
- EC techniques will be more popular in pattern recognition
  - Classification, Clustering
  - GP, GAs, PSO, DE,
  - EC will be in more main stream conferences and journals
- GPU will be a popular tool

Future Events

- CEC Special Session on Evolutionary Deep Learning and Applications, IEEE CEC 2021
  - Organisers: Yanan Sun, Bing Xue, Chuan-Kang Ting, Mengjie Zhang
  - 28th June - 1st July, 2021, Krakow (POLAND)
  - Websites: https://cec2021.mini.pw.edu.pl/en/program/special-sessions
- CEC Special Session on Evolutionary Computer Vision and Image Processing, Pablo Mesejo, Harith Al-Sahaf
  - 28th June - 1st July, 2021, Krakow (POLAND)
  - Websites: https://cec2021.mini.pw.edu.pl/en/program/special-sessions
- IEEE SSCI 2021 Symposia:
  - CI in Feature Analysis, Selection and Learning in Image and Pattern Recognition (**IEEE FASLIP**))
  - CI for Multimedia Signal and Vision Processing (**IEEE CIMSIVP**)
  - Paper Submissions: **06 August 2021**