UNIVERSITY OF TWENTE.

An Overview of Genetic Programming with Applications to Cybersecurity

Luca Mariot

l.mariot@utwente.nl

Cybsec@SCS Seminars – November 8, 2024

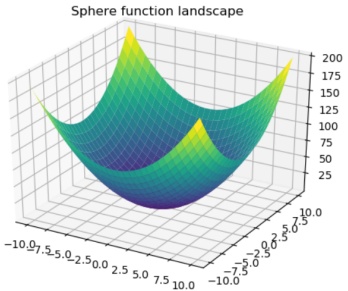
Introduction to Evolutionary Algorithms and Genetic Programming

Optimization Problems

- Set S (**search space**) equipped with a **fitness function** $fit : S \to \mathbb{R}$, giving a score to candidate solutions $x \in S$

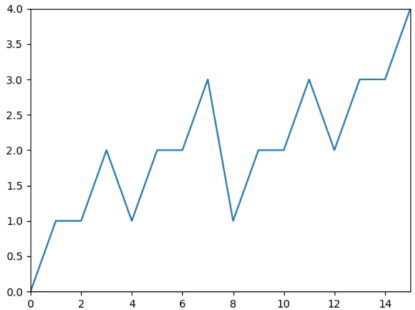
Minimization (Sphere Function):

$$x^* = argmin_{x \in S}\{fit(x)\}$$

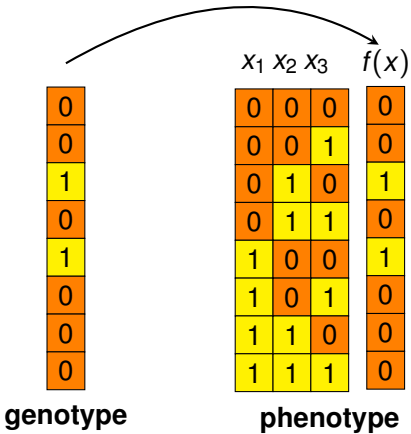


Maximization (OneMax):

$$x^* = argmax_{x \in S}\{fit(x)\}$$

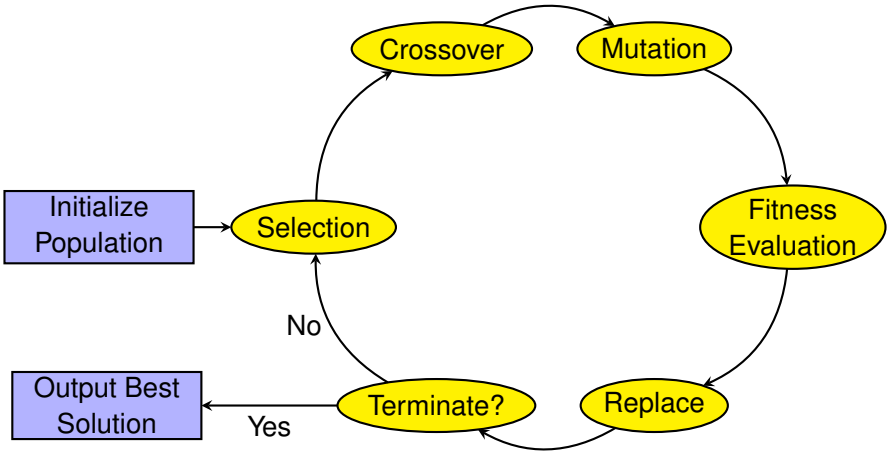


Evolutionary Algorithms (EA)



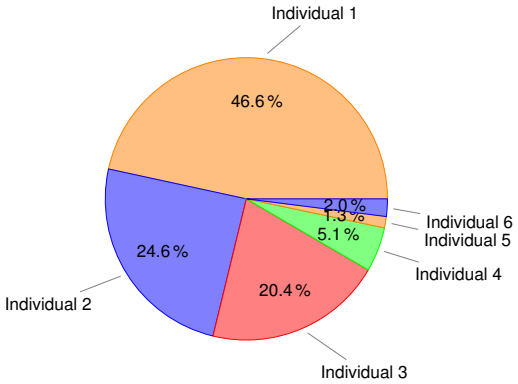
- Optimization algorithms loosely based on evolutionary principles
- **Genetic Algorithms (GA):** introduced by **J. Holland** (1975)
- **GA genotype**: fixed-length bitstrings
- **phenotype**: candidate solutions corresponding to genotypes

The EA Loop


Initialize Population
Selection
Crossover
Mutation
Fitness Evaluation
Replace
Terminate?
No
Yes
Output Best Solution

Selection

Roulette-Wheel (RWS): selection probability proportional to individual's fitness

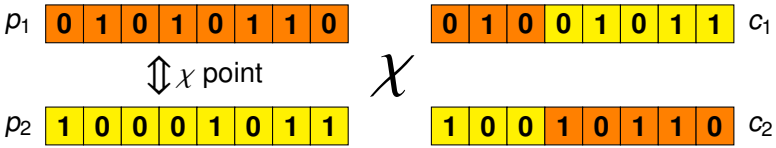


Tournament (TS): select the fittest individual from a random sample of t individuals

Crossover and Mutation

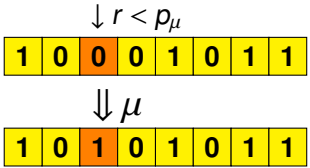
Crossover: Recombine the genes of two parents individuals (Exploitation)

GA Example: One-Point Crossover



Mutation: Introduce new genetic material in the offspring (Exploration)

GA Example: Bit-flip mutation



Replacement and Termination

- **Elitism**: keep the best individual from the previous generation
- **Termination criterion**: budget of fitness evaluations, solutions diversity, ...

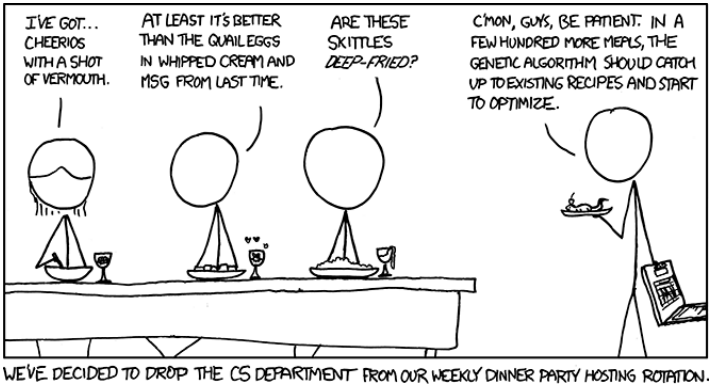
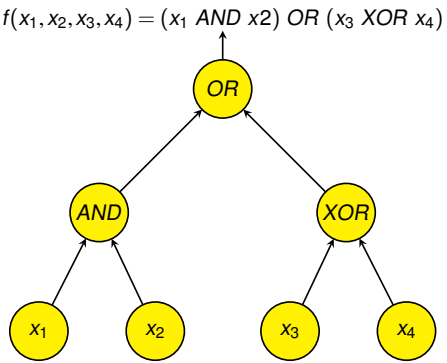


Image credit: `https://xkcd.com/720/`

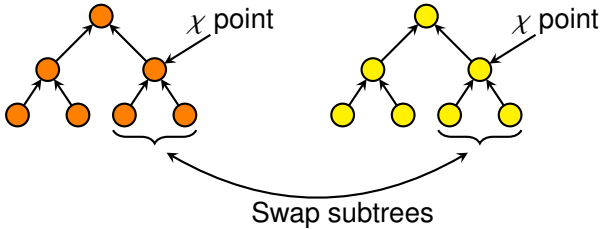
Genetic Programming (GP)

- Introduced by **J. Koza** (1992)
- **Idea:** evolve computer programs to solve specific tasks
- **GP Genotype**: a syntactic tree
- Terminal nodes: input variables of a program
- Internal nodes: operators (e.g. AND, OR, NOT, XOR, ...)

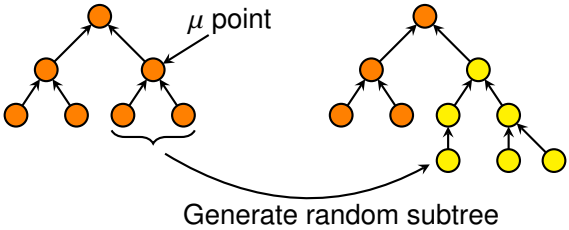


Crossover and Mutation in GP

GP Example: Subtree Crossover



GP Example: Subtree mutation

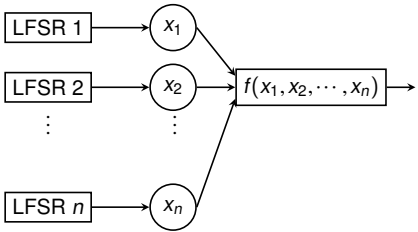


GP applications in cryptography

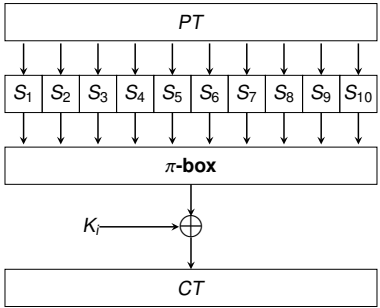
Use of EA in symmetric cryptography

Design of primitives as a **combinatorial optimization problem**, examples [C21]:

Boolean functions for stream ciphers



S-Boxes $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ for block ciphers

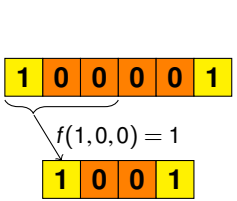


- **Idea:** embed metrics for cryptographic properties into the fitness function
- Many works in the literature [D23, P16, P17, P18]

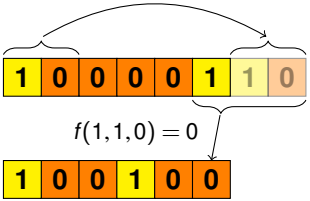
Cellular Automata

- One-dimensional Cellular Automata (CA):

Example: $n = 6$, $d = 3$, $f(s_i, s_{i+1}, s_{i+2}) = s_i \oplus s_{i+1} \oplus s_{i+2}$



No Boundary CA

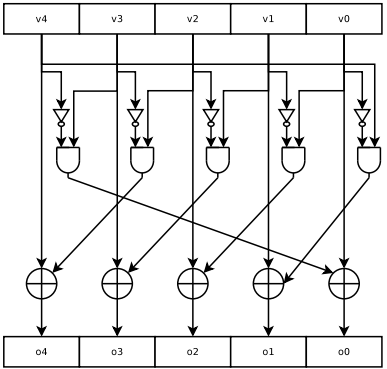


Periodic Boundary CA

- Each cell updates its state $s \in \{0,1\}$ by applying a local rule $f : \{0,1\}^d \to \{0,1\}$ to itself and the $d-1$ cells on its right [M19]

Real world CA-Based Crypto: Keccak χ S-box

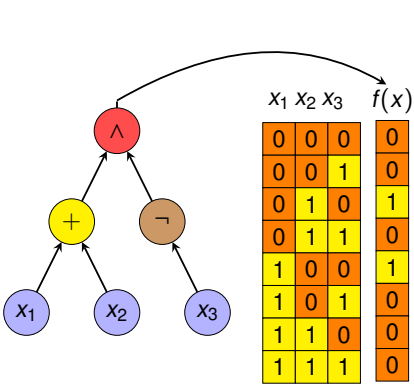
- Local rule: $\chi(x_1, x_2, x_3) = x_1 \oplus (1 \oplus (x_2 \cdot x_3))$ (rule 210)
- Invertible for every odd size n of the CA

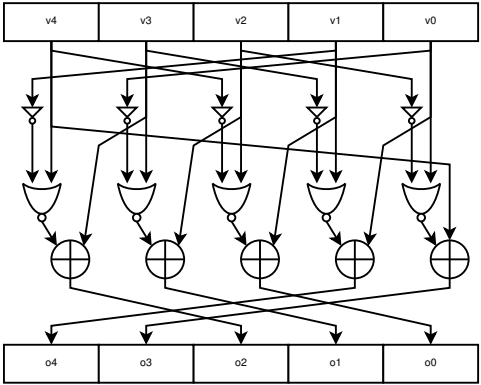


- Used as a PBCA with $n = 5$ in Keccak [M19]

CA S-boxes found by GP

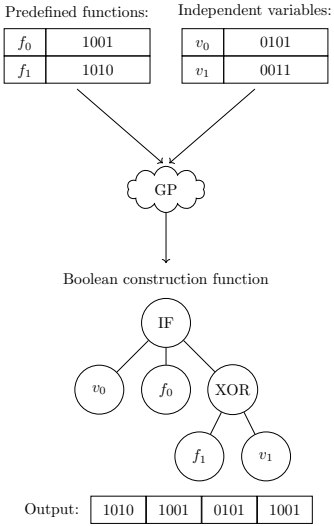
Idea: evolve the CA rule of the S-box, optimizing: **crypto** properties (nonlinearity, differential uniformity) and **implementation** properties (area, latency) [P17, M19]





- Up to size 7×7: results on par or slightly better than the state of the art

Evolving Constructions of Boolean functions with GP



- **Idea:** Do not evolve primitives directly, but rather their mathematical constructions [C22]
- Use Boolean minimizers to interpret the constructions
- **Research Question**: Does GP obtain new constructions?
- **Finding**: GP is a "lazy student"

GP as a supervised learner

Symbolic Regression with GP

- **Problem:** find a symbolic expression that minimizes the error in approximating a given set of data points

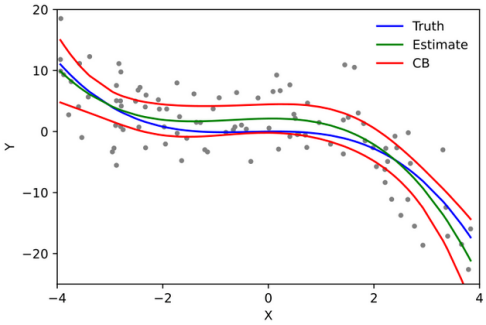


Image credit: https://en.wikipedia.org/wiki/Polynomial_regression

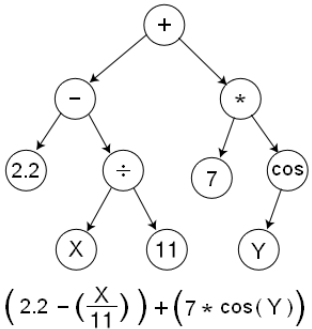


Image credit: https://en.wikipedia.org/wiki/Symbolic_regression

- **Idea:** use GP to evolve the symbolic expression [K93]

GP for Anomaly Detection [C16]

- **Problem:** given a dataset of "normal behavior" (e.g., network traffic), identify anomalies
- **Possible approach:** use kernel density estimation (KDE)
- Given training points $x_1, \ldots, x_n \in \mathbb{R}^d$, estimate the density at x as:

$$\hat{p}(x) = \frac{1}{n} \sum_{i=1}^{n} K_h(x - x_i)$$

where K_h is a kernel function (e.g. Gaussian)

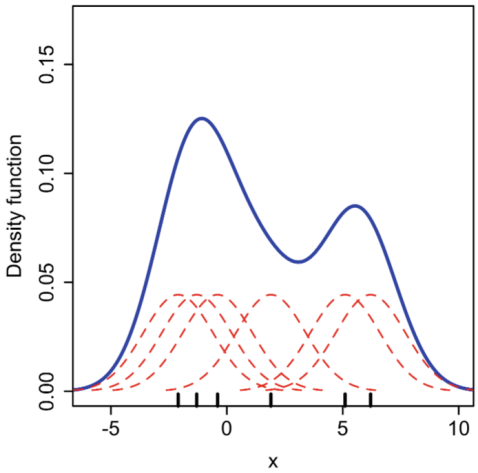


Image credit:
https://en.wikipedia.org/wiki/Kernel_density_estimation

GP for Anomaly Detection [C16]

- KDE is expensive to compute at query time
- **Idea:** use GP to learn a **surrogate** f of the density function d:

$$f(x_i) \approx d(x_i)$$

- Can be turned into a symbolic regression problem, with RMS fitness:

$$fit(f) = \sqrt{\frac{\sum_{i=1}^{n} (f(x_i) - d(x_i))^2}{n}}$$

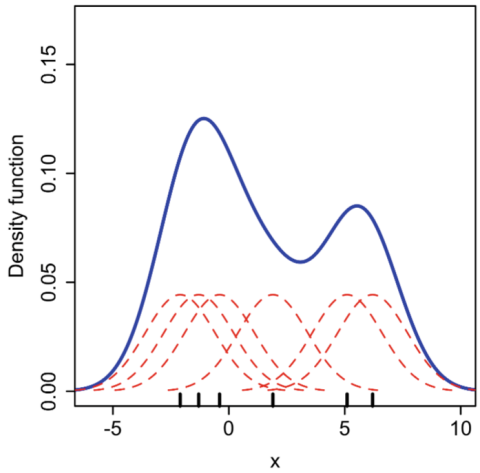


Image credit:
https://en.wikipedia.org/wiki/Kernel_density_estimation

GP for Anomaly Detection [C16]

- GP trained on SNL-KDD network intrusion detection dataset (!)
- **Finding:** GP on par with KDE, slightly outperforms 1-class SVM
- **Advantage:** much more efficient! Only the GP surrogate is used at query time

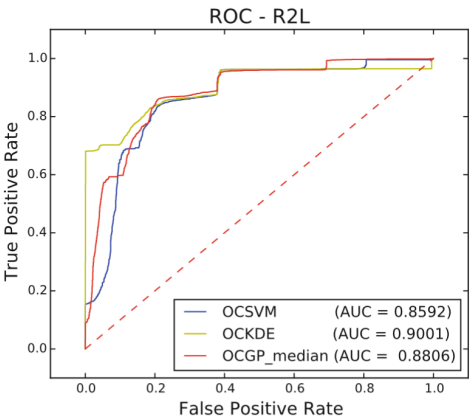


Image credit: [C16]

GP as a predictive model

Next Word Prediction (NWP)

- Task: given an initial sequence of k words $w_1, \cdots, w_k$, complete the sentence by predicting the last word w_{k+1}
- *Exact* or *plausible* prediction?

Input

the	cat	is	on	the

Original completion: **table**

Next Word Prediction (NWP)

- Task: given an initial sequence of k words $w_1, \cdots, w_k$, complete the sentence by predicting the last word w_{k+1}
- *Exact* or *plausible* prediction?

Input

the	cat	is	on	the

Plausible prediction: **chair**

Next Word Prediction (NWP)

- Task: given an initial sequence of k words $w_1, \cdots, w_k$, complete the sentence by predicting the last word w_{k+1}
- *Exact* or *plausible* prediction?

Input

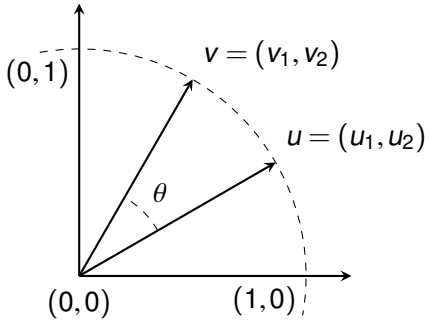
the	cat	is	on	the

Plausible (?) prediction: **tractor**

- Setting: plausible word predictions with GP [M20]

GP Input: word2vec embedding

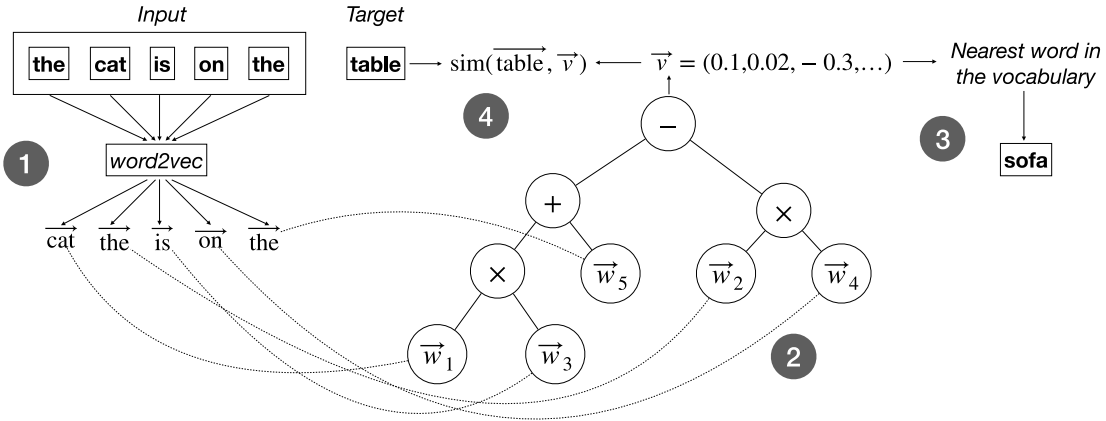
- *word2vec*: a NN-based model that learns a *word embedding* of a vocabulary over the vector space $\mathbb{R}^d$



- Similar words u, v are mapped to vectors $\vec{u}, \vec{v} \in \mathbb{R}^d$ with a high *cosine similarity*:

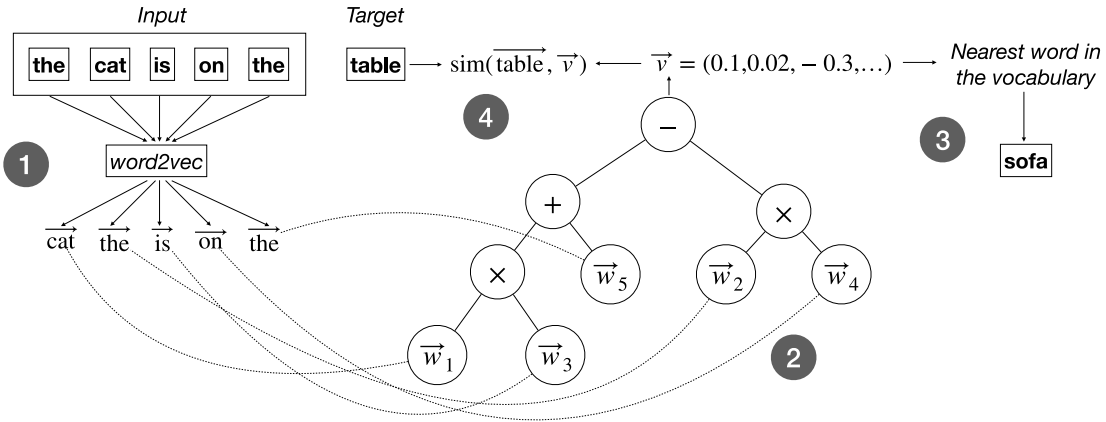
$$\text{sim}(\vec{u}, \vec{v}) = \frac{\sum_{i=1}^{d} \vec{u}_i \vec{v}_i}{\|\vec{u}\|_2 \cdot \|\vec{v}\|_2}$$

Evaluation of a GP tree



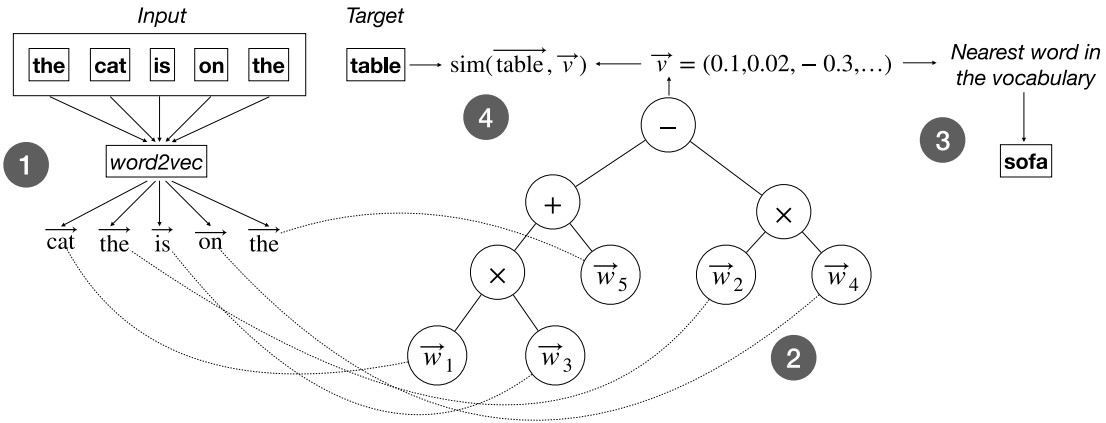
(1) The input words are converted to vectors through the word2vec embedding

Evaluation of a GP tree



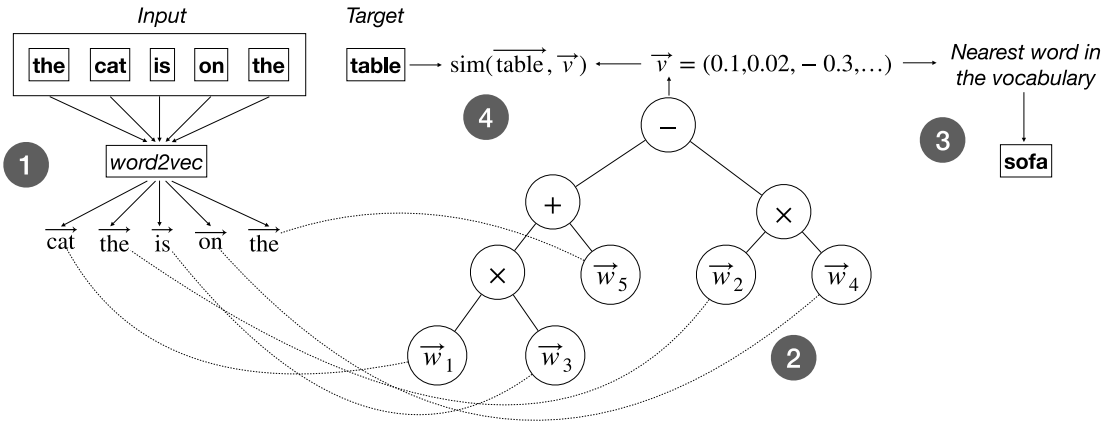
(2) The vectors of the input words are fed to the GP tree, and the output vector is evaluated at the root node

Evaluation of a GP tree



(3) The output vector is converted to the most similar word occurring in the vocabulary learned by word2vec

Evaluation of a GP tree

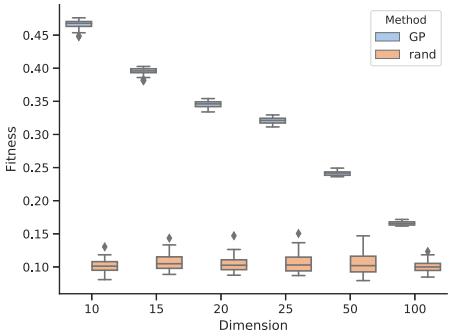


(4) Compute the similarity between the original (target) word and the word predicted by GP

Learning a language model with GP [M20]

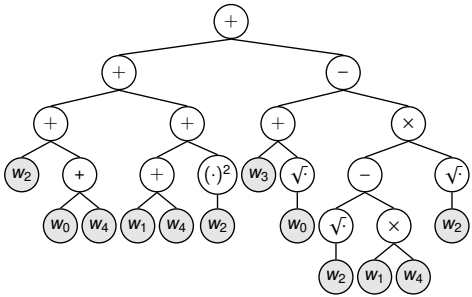
- Dataset: *Million News Headlines* (MNH)
- Headlines length: 6 words (267 292 instances in MNH)
- word2vec embedding dimensions: $d \in \{10, 15, 20, 25, 50, 100\}$
- **Main finding**: *The GP evolutionary process is able to learn, to a certain extent, a representation of the MNH dataset*

GP vs. Random predictor



Example of tree evolved by GP

Example of individual evolved by GP and its predictions (embedding dimension 10)



Predicted headline	Original
Regional education to fund youth **preschool**	allowance
Aerial footage of flooded Townsville **houses**	homes
Greens renew call for tax **changes**	review
Napthine to launch new Portland **rail**	marina
4 charged over 10000 jewellery **robberies**	heist
Vanstone defends land rights act **overhaul**	changes
Community urged to seek infrastructure **funds**	funding
Govt. pressured on company tax **bureaucracy**	rates
Petition urges probe into abattoir **maintenance**	closure
Rain does little for central **towns**	Victoria

Wrap-up

Conclusions and Future Directions

Conclusions:

- GP is a versatile evolutionary algorithm: can be used for optimization, supervised learning, prediction, ...
- Advantage: **Interpretability** of the solutions
- Many other variants exist! Cartesian GP, Linear GP, Semantic GP, ...

Interesting Ideas for the Future:

- GP text generator to craft adversarial examples in LLMs
- Privacy-preserving evolution and evaluation of GP trees (e.g., with secure multiparty computation)

References


[B11] G. Bertoni, J. Daemen, M. Peeters, G. Van Assche: The Keccak reference (2011)


[C16] V. L. Cao, M. Nicolau, J. McDermott: One-Class Classification for Anomaly Detection with Kernel Density Estimation and Genetic Programming. Proceedings of EuroGP 2016, pp. 3–18 (2016)


[C22] C. Carlet, M. Djurasevic, D. Jakobovic, L. Mariot, S. Picek: Evolving constructions for balanced, highly nonlinear boolean functions. Proceedings of GECCO 2022, pp. 1147–1155 (2022)


[C21] C. Carlet: Boolean functions for cryptography and coding theory. Cambridge University Press (2021)


[D23] M. Djurasevic, D. Jakobovic, L. Mariot, S. Picek: A survey of metaheuristic algorithms for the design of cryptographic Boolean functions. Cryptography and Communications 15(6): 1171–1197 (2023)


[K93] J. R. Koza, M. A. Keane, J. P. Rice: Performance improvement of machine learning via automatic discovery of facilitating functions as applied to a problem of symbolic system identification. Proceedings of ICNN 1993, pp. 191–198 (1993)


[M20] L. Manzoni, D. Jakobovic, L. Mariot, S. Picek, M. Castelli: Towards an evolutionary-based approach for natural language processing. Proceedings of GECCO 2020, pp. 985–993 (2020)


[M19] L. Mariot, S. Picek, A. Leporati, and D. Jakobovic. Cellular automata based S-boxes. Cryptography and Communications 11(1):41–62 (2019)


[P18] S. Picek, K. Knezevic, L. Mariot, D. Jakobovic, A. Leporati: Evolving Bent Quaternary Functions. Proceedings of CEC 2018, pp. 1–8 (2018)


[P17] S. Picek, L. Mariot, B. Yang, D. Jakobovic, N. Mentens: Design of S-boxes defined with cellular automata rules. Conf. Computing Frontiers 2017: 409-414 (2017)


[P16] S. Picek, D. Jakobovic, J.F. Miller, L. Batina, M. Cupic: Cryptographic Boolean functions: One output, many design criteria. Appl. Soft Comput. 40: 635-653 (2016)