

ANNs – Neuroevolution

Overview

ML Technique generating increasingly better topologies, weights & hyperparameters by means of evolutionary algorithms (Hyper Parameter: Parameter whose value is used to control the learning process)

Mutation & Crossing over/removing genetic code

- Mutate topology & weights
- Combine features of high performers
- Eliminate low performers to find better solutions

(Pauls, 2020)

Each encoding in turn is decoded into the fitting phenotype, then employed and evaluated

(Summit & Webb, 2011)

Population based optimisation

- Individual stored as genome, not neural network → mapped to NN
1. Init set of genomes & apply to problem environment
 2. Assign fitness score based on how well solved problem → Optimisation loop
 3. Pop continuously mutated, recombined & evaluated → Natural Selection
 4. Eventually yield NN that solves problem well, might have been impossible to come up with by hand

(Pauls, 2020)

Generational Neuroevolution

- Whole population being in one step at a time

(Pauls, 2020)

Competitive Coevolution

- Asynchronous, performed on a per-genome basis

(Pauls, 2020)

Genetic Encoding

Faculty that maps genome to the according NN

Also responsible for granularity of search space by limiting complexity of genome encoding

Efficiently mutate & recombine

Efficient representation of ANN necessary

- “Allows frequently employed genetic code to **not have to analyse highly complex data structures** but for them to **analyse compact genetic codes that can be processed fast**”
 - ∴ can generate sensible mutations or check if 2 genomes are eligible for recombination
 - ∴ Neuroevolution algorithms **operate on genetic encoding exclusively** instead of complex data structures

(Pauls, 2020)

Representation

Genetic representation → Genotype

Mapped NN → Phenotype

Direct encoding

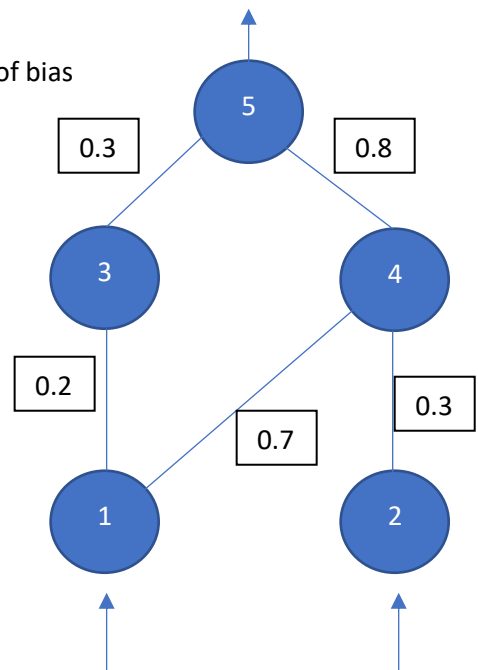
Encode connection & weight into genotype, exclude possibility of bias

- Enormous search space with fine granularity

(Pauls, 2020)

e.g.

Gene #	In	Out	Weight
Gene #1	1	3	0.2
Gene #2	1	4	0.7
Gene #3	2	4	0.3
Gene #4	3	5	0.3
Gene #5	4	5	0.8



Indirect encodings

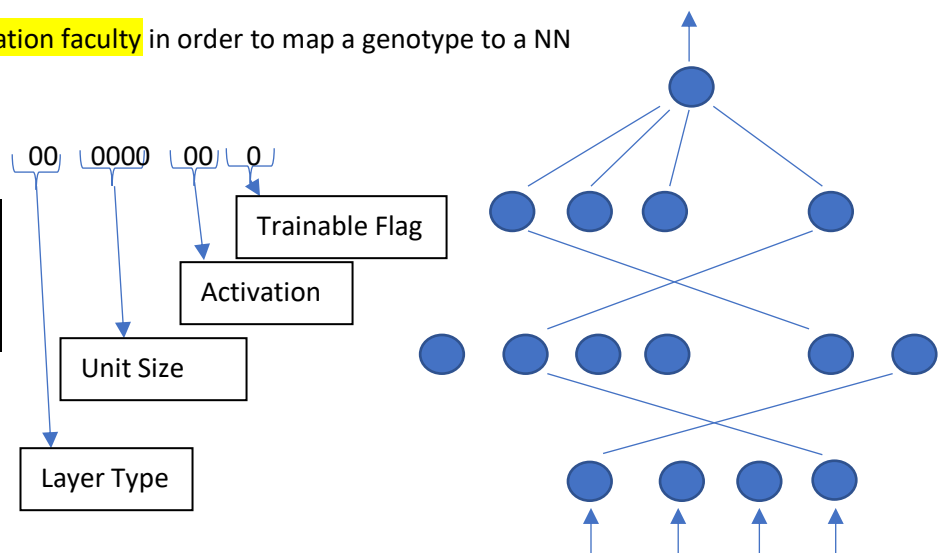
Custom encoding not intuitively translatable into a NN

- **Requires separate translation faculty** in order to map a genotype to a NN

(Pauls, 2020)

e.g.

Gene #1	000001001
Gene #2	001100001
Gene #3	000100100
Gene #4	000000011



Both Encoding demonstrate how well the genetic encode determines the size of the search space.

(Pauls, 2020)

Mutation, Recombining and Reproduction

Reproduction

Traversal of Search Space

Create new genomes **through mutating or recombining** genomes fit to be parents

(Pauls, 2020)

Mutations

Explore viability of ANN architecture, weight distribution and hyperparameters

Define 3 aspects:

1. **What part of genetic encoding**
 - a. Topology, weights, hyperparameters, random selection of bits
2. **To what extent**
 - a. Large changes if it's doing bad, small changes if its doing good
3. **Somehow directional or random**
 - a. If it can map a lack of performance in the problem environment

(Pauls, 2020)

Recombining

Merge 2 promising genomes & distinct features

(Pauls, 2020)

Evaluation

Map genome to NN, apply it to the environment, calculate fitness score

Only one thing can be modified for evaluation → The way the fitness scores are calculated → Calculated on a case-to-case basis

(Pauls, 2020)

Evolving

Optimise weights to a fixed neural network architecture

Evolve a section of the population → Evolve solution components instead of full solutions

Fitness expresses how well each neuron cooperates with others to forma full network

Topology can have a large effect on function

- Evolving appropriate topologies can achieve good performance faster than evolving weights alone

(Summit & Webb, 2011)

Natural and Parental selection

Choice of genomes for parents and for genomes to be removed

It's heavily dependent on the algorithm used

Fractional number for parents and those removed → Parent fraction ≠ removed fraction

(Pauls, 2020)

Competitive Coevolution selection is difficult due to asynchronosity

(Pauls, 2020)

Therefore, each individual genome must be evaluated, not the NN in itself so that the genomes can be removed, mutated or combined.

Application

Self-driving cars

- Trained virtually in UE5 or something similar before deployed

(Sainath, Vignesh, Siddarth & Suganya, 2021)

References

Pauls, P. (2020, January 19). *A Primer on the Fundamental Concepts of Neuroevolution*. Retrieved from Towards data science: <https://towardsdatascience.com/a-primer-on-the-fundamental-concepts-of-neuroevolution-9068f532f7f7>

Summit, C., & Webb, G. I. (Eds.). (2011). *Encyclopedia of Machine Learning*. Springer New York, NY. doi:<https://doi.org/10.1007/978-0-387-30164-8>

Sainath, G., Vignesh, S., Siddarth, S., Suganya, G. (2021). *Application of Neuroevolution in Autonomous Cars*. In: Kannan, R.J., Geetha, S., Sashikumar, S., Diver, C. (eds) *International Virtual Conference on Industry 4.0. Lecture Notes in Electrical Engineering*, vol 355. Springer, Singapore. https://doi.org/10.1007/978-981-16-1244-2_2