
Optimizing neural networks using genetic algorithms

By Johan Nilsson Hansen

Agenda

- Introduction
- Genetic algorithms
- Outcomes
- Q & A

Introduction

- Lots of different network architectures
- Huge search space
- Time consuming or costly
- **“Good enough” results much faster**

Genetic algorithms

- Inspired by Biology - survival of the fittest
- Lots of individuals with a different set of genes
- More fit individuals have higher chance of breeding the next gen
- Unfit individuals get extinct

Genetic algorithms

- Encode parameters as genes
 - #Layers?
 - #Nodes?
 - Weight initializer?
 - Activation function?
 -

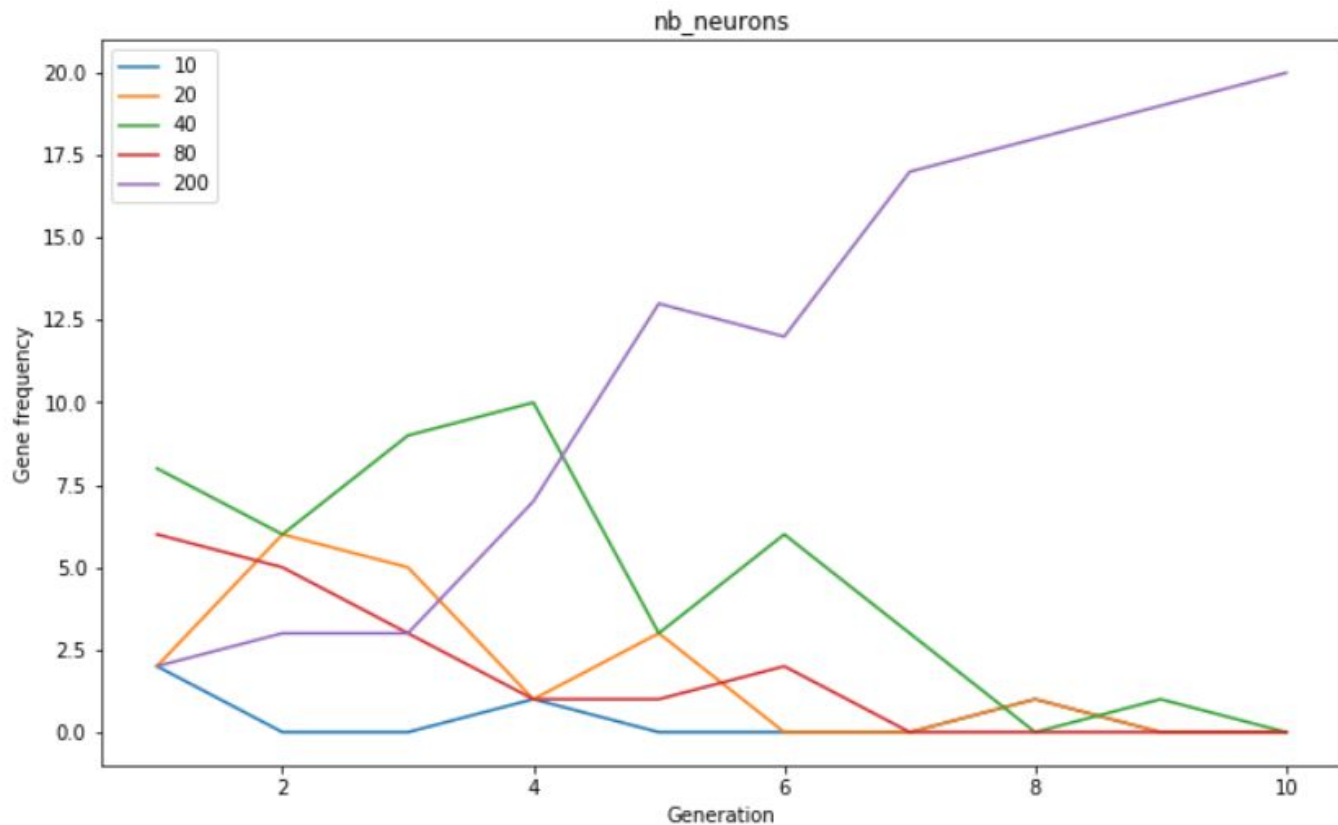
Genetic algorithms

1. Randomly create a set of neural networks (individuals)
2. Score each individual (The lower the validation error - the greater the fitness)
3. Breed the most fit individuals
 - Pick a gene from either parent
4. Randomly mutate some of the children
 - Avoid local minima
5. Kill off the individuals that didn't get to breed
6. Repeat 2) for the next generation

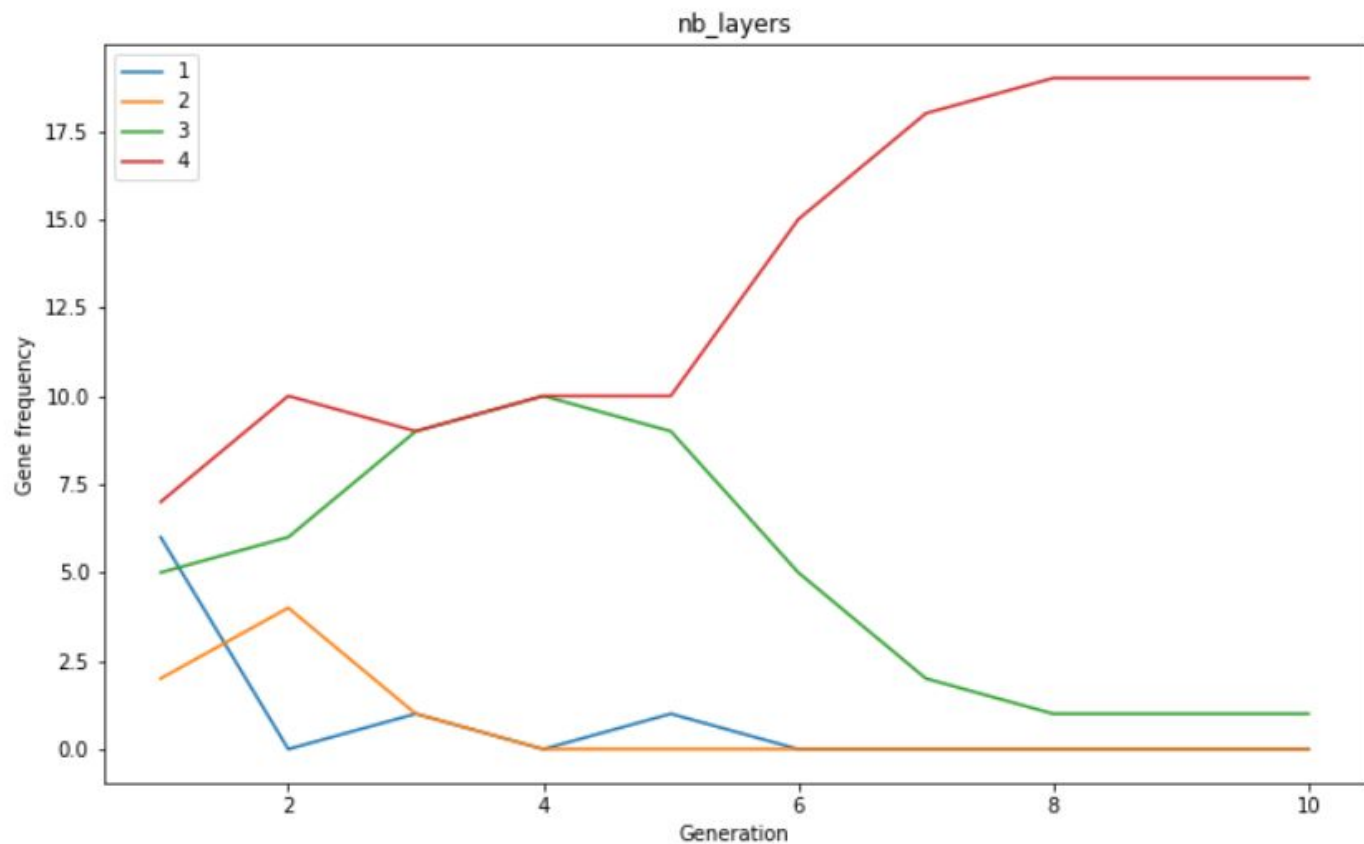
Outcomes

- The winner
- Dominant genes
- Magnitudes faster

Outcomes



Outcomes



Resources

- <https://blog.coast.ai/lets-evolve-a-neural-network-with-a-genetic-algorithm-code-included-8809bece164>
- <https://lethain.com/genetic-algorithms-cool-name-damn-simple/>

Thank you

Q & A