The journey to predict the future or to automate mundane stuff or understand a large amount of data for building strategies has come a long way from statistical techniques to machine learning algorithms to neural nets and deep learning. The potential of these techniques and the corresponding results have motivated new use cases and served humanity to date.

Data Scientists, statisticians, and researchers have developed many techniques and architectures to serve specific purposes, and the architectures have been utilized in numerous other fields (a concept known as Transfer Learning). These hand-designed architectures used for a problem or even for related problems pose a question. Have we done enough? Is this the best we could do? Are there theoretically better models for the given data? Have we searched all or enough search space? Has there been a bias coming from the researcher and his experiences with search space?

Here comes NAS or Neural Architecture Search to the rescue. NAS automates network architecture engineering, learning network topology that can achieve best possible performance on a certain task.

NAS methods generally have 3 components:

1. Search Space
2. Search algorithm
3. Model evolution

In this article we would understand some early NAS architectures and implementation of a Amazon reviews text classification using Autokeras library for python.

NAS Architectures

Neural Architecture Search has drawn huge interest of the research community since 2015 and numerous research papers have been published since then and some in works even now. Here we will discuss some of the popular techniques from Reinforcement Learning based to Genetic Algorithm based to some based on Bayesian Optimisation.

**MetaQNN** ([Baker et al. 2017](https://arxiv.org/abs/1611.02167))

An agent is trained to sequentially choose convolutional layers using Q-table ([*Q-learning*](https://lilianweng.github.io/lil-log/2018/02/19/a-long-peek-into-reinforcement-learning.html#q-learning-off-policy-td-control)) following a [ϵ-greedy](https://lilianweng.github.io/lil-log/2018/01/23/the-multi-armed-bandit-problem-and-its-solutions.html#%CE%B5-greedy-algorithm) exploration and exploitation strategy. Validation accuracy is chosen as a reward.

A picture containing text

Description automatically generatedhere,

S – state

A – action

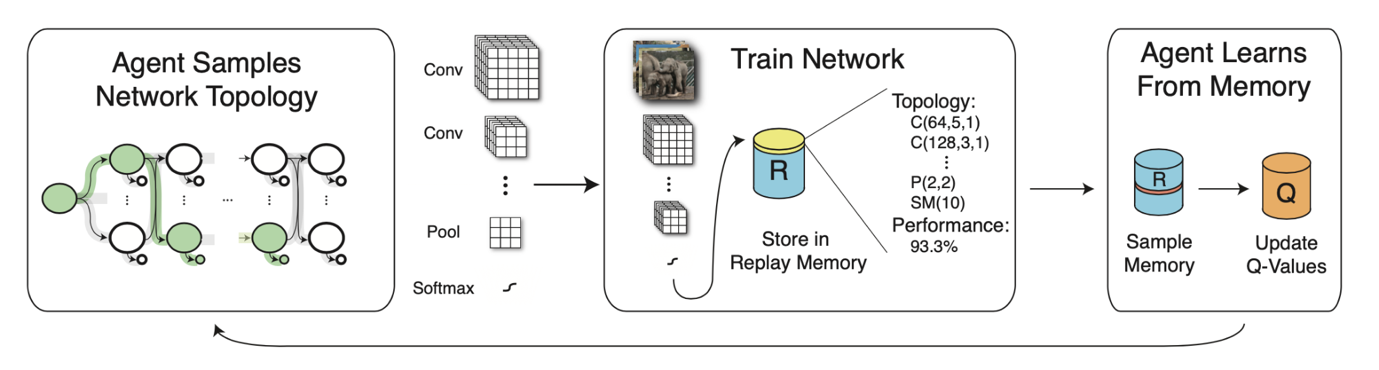
R – reward

Alpha – learning rate

Gamma – discount factor (0-immediate reward, 1-future reward)

Q – value of an action in a particular state

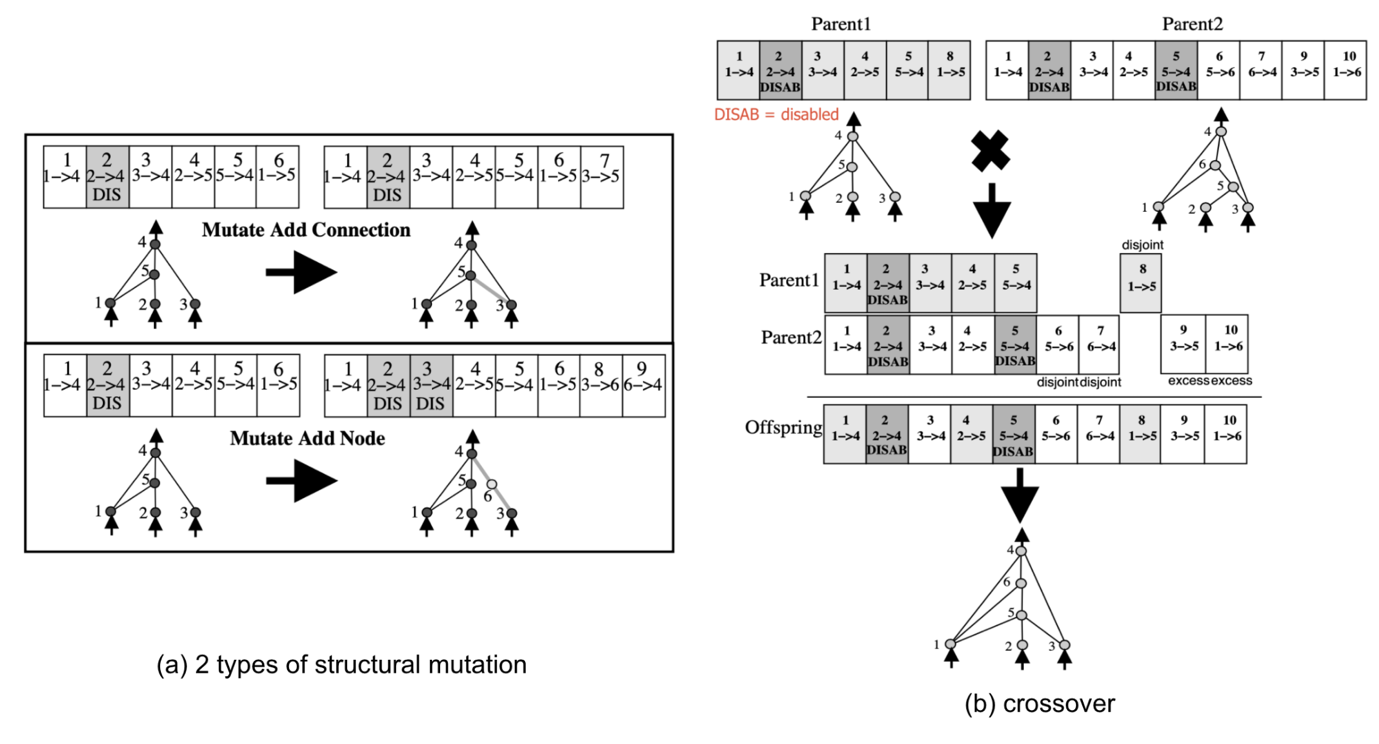
T – current step



*Overview of MetaQNN - designing CNN models with Q-Learning. (Image source:*[*Baker et al. 2017*](https://arxiv.org/abs/1611.02167)*)*

**NEAT** (*NeuroEvolution of Augmenting Topologies*)

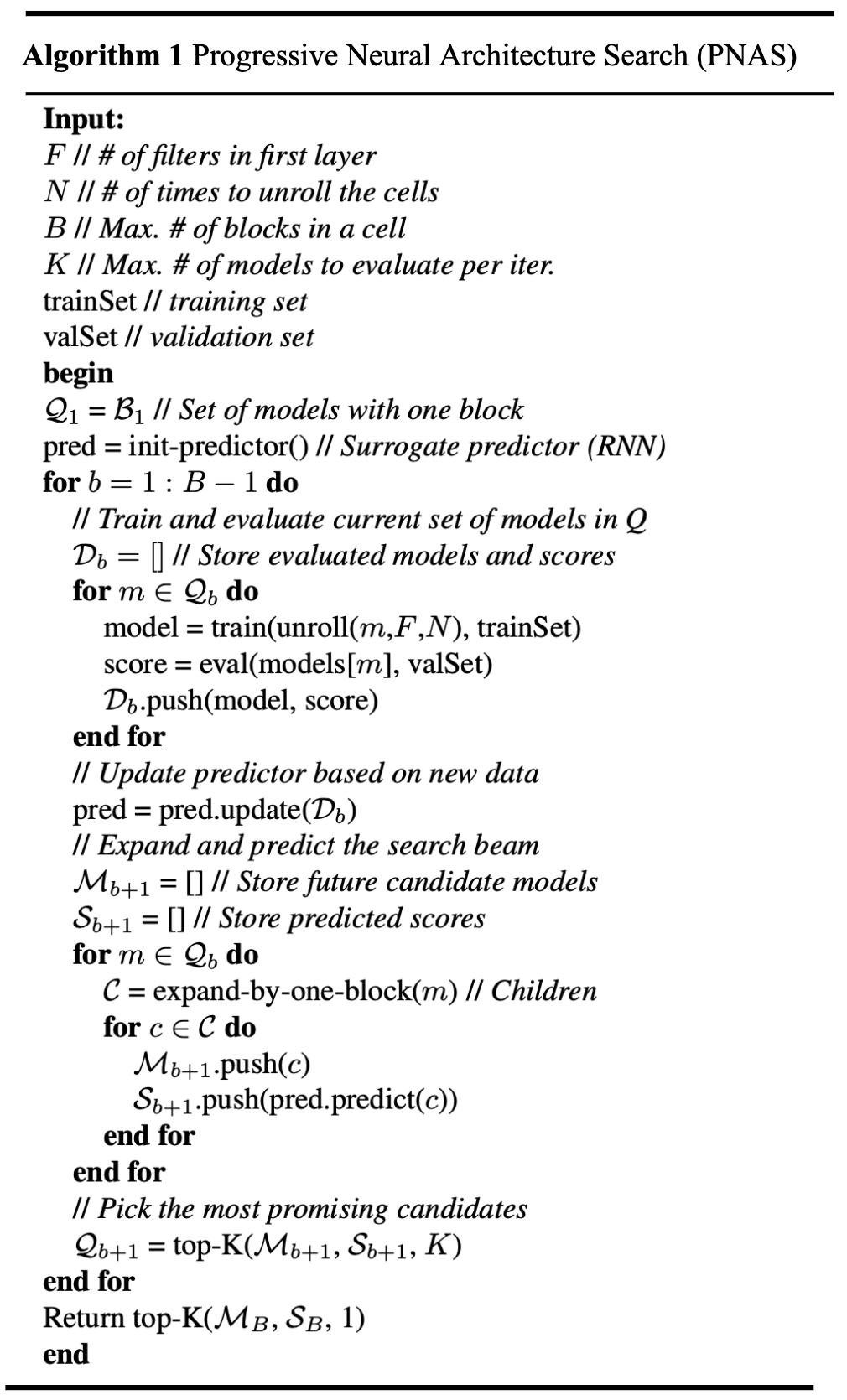
NEAT uses [genetic algorithm (GA)](https://en.wikipedia.org/wiki/Genetic_algorithm) to update connection weights and network topology simultaneously. Each gene encodes the complete information about configuring a network, i.e. node weights and edges. The population grows by applying mutation to both the weights and edges, as well as crossover between two parent genes.



*Mutations in the NEAT algorithm. (Image source: Fig 3 & 4 in*[*Stanley & Miikkulainen, 2002*](http://nn.cs.utexas.edu/downloads/papers/stanley.ec02.pdf)*)*

*Progressive NAS* (**PNAS**; [Liu, et al 2018](https://arxiv.org/abs/1712.00559))

As the name suggests PNAS searches increasingly complex models at each step using a Sequential Model-based Bayesian Optimization (SMBO) strategy. PNAS works similar to A\* search, as it searches for models from simple to hard while simultaneously learning a surrogate function to guide the search.



*The algorithm of Progressive NAS. (Image source:*[*Liu, et al 2018*](https://arxiv.org/abs/1712.00559)*)*

**Macro vs Micro NAS architectures:**

Macro - starting from small network [Huge search space]

Micro - use pre-built architecture and improve few components [Smaller search space]