

**MACHINE INTELLIGENCE**

**UE20CS302**

**Literature Survey**

**Batch No:**

**Team Members:**

Md Taseen Atehar – PES2UG20CS191

M Manonmana Udupa – PES2UG20CS181

Hrishikesh B H – PES2UG20CS144

**Project Title**: Evolving Organisms with Deep Learning and Genetic Algorithm

Name : Md Taseen Atehar

SRN: PES2UG20CS191

Section: C

***Research on an Improved Ant Colony Algorithm-Fusion with Genetic Algorithm for Route Planning***

10.1109/ITNEC48623.2020.9084730

Xiaoyan Chen, Yuhe Dai

College of Electronic Information and Automation, Tianjin University of Science & Technology, Tianjin, China

This paper focuses on an improved ant colony algorithm for solving the problem of route planning. An ant colony algorithm (ACO) fusion with genetic algorithm (GA) is proposed and verified by simulations.

The ant colony algorithm was first proposed by Dorigo in 1992, and it was inspired by the ants' population cooperation between round-trip foraging points and nests. The ant colony selects the route with a higher concentration according to the pheromone concentration, and uses the pheromone as a medium for communication activities.

Method:

In the initial stage of the algorithm, the genetic algorithm is used to calculate some results, and the initial distribution of pheromone is determined. Then, the initial distribution of pheromone is converted into the initial value of pheromone required by the ant colony algorithm through a series of evolution operations. Then use the ant colony algorithm to search the optimal solution of the problem, and play the role of the efficiency of the ant colony algorithm and the positive feedback mechanism.

Advantages:

According to the positive feedback effect of the ant colony, the shortest route to the round trip is gradually found, and the foraging efficiency is improved.

By gradually optimizing the adaptability of the new individuals in next generation, the new approximate solution can better meet the indicators of cargo route optimization.

Disadvantages:

The algorithm falls victim to large blindness, low efficiency, often falls into local optimum.

Conclusion:

Experimental tests have indicated that with the interior nodes of the interesting region increasing, the route distance obtained by the improved algorithm in this paper is shorter than GA and ACO algorithms both.

Moreover, the proposed ant colony algorithm fusion with GA has fewer iterations to obtain the optimal solution, and the calculation time cost is decreased obviously.

***A Hybrid Genetic Algorithm Based on Information Entropy and Game Theory***

10.1109/ACCESS.2020.2971060

Li Jiacheng, Li Lei

Faculty of Science and Engineering, Hosei University, Tokyo, Japan

Method:

To overcome the disadvantages of traditional genetic algorithms, which easily fall to local optima, this paper proposes a hybrid genetic algorithm based on information entropy and game theory. First, a calculation of the species diversity of the initial population is conducted according to the information entropy by combining parallel genetic algorithms, including using the standard genetic algorithm (SGA), partial genetic algorithm (PGA) and syncretic hybrid genetic algorithm based on both SGA and PGA for evolutionary operations. Furthermore, with parallel nodes, complete-information game operations are implemented to achieve an optimum for the entire population based on the values of both the information entropy and the fitness of each subgroup population. Additionally, the Rosenbrock, Rastrigin and Schaffer functions are introduced to analyse the performance of different algorithms. The results show that compared with traditional genetic algorithms, the proposed algorithm performs better, with higher optimization ability, solution accuracy, and stability and a superior convergence rate.

The optimization ability of the genetic algorithm is reflected by the diversity of the population and the convergence speed of the algorithm. The traditional genetic algorithm easily falls to local optima, and this issue is largely related to the diversity of individuals in the population. Therefore, it is necessary to maintain the diversity of the population in each generation. Additionally, the genetic algorithm involves genetic and selection operations; thus, whether the new population is retained and how it is retained will affect the final optimization result and the speed of optimization. Therefore, the concept of information entropy is introduced in this paper.

A typical problem in Game Theory research is that two or more participants (called in-house players) make decisions on a confrontational or competitive basis so that their own party obtains the best possible results. Moreover, the so-called game is a set of rules that stipulate the methods and regulations that should be followed throughout the game (or competition, struggle, etc.), including the players, strategies, outcome after strategy selection, and so on.

Advantages:

In this paper, the genetic algorithm is improved, and multi-group genetic operations are performed in parallel.

Information entropy is introduced to quantitatively analyse the diversity in the evolution process and ensure diversity in population genetics. Additionally, combined with game theory, various types of evolutionary processes occur. The changes in group information entropy using the game strategy that is most conducive to the diversity and adaptability of the whole group are considered to strengthen good individuals and eliminate invalid individuals. Three test functions are introduced to assess the validity and convergence of commonly used test algorithms.

Disadvantages:

Apart from complexity, there are none

***Self-Adaptive Hybrid Extreme Learning Machine for Heterogeneous Neural Networks***

10.1109/IJCNN48605.2020.9207608

This paper presents a hybrid algorithm for the creation of heterogeneous single layer neural networks (SLNNs). The proposed self-adaptive heterogeneous hybrid extreme learning machine (SA-He-HyELM) trains a series of SLNNs with different neuron types in the hidden layer utilizing the extreme learning machine (ELM) algorithm. These networks are evolved into heterogeneous networks (networks having different combinations of hidden neurons) with the help of a modified genetic algorithm (GA). The algorithm is able to handle two architecturally different neuron types: traditional low order (linear) units and higher order units with different transfer functions. The GA is fully self-adaptive and uses one novel hybrid crossover operator along with a self-adaptive mutation operator in order to retain ELM’s simplicity and minimize the number of parameters need tuning. The experimental part of the current paper involves testing SA-He-HyELM with traditional ELM and other three ELM-based methods. The experimental part utilized a series of regression and classification experiments on relatively large datasets. In all cases the proposed method managed to get lower MSE or higher classification accuracy when compared to the aforementioned methods.

Advantage:

SA-He-HyELM is able to evolve homogeneous networks into heterogeneous ones with better generalization ability while retaining ELM’s simplicity. The only user defined parameter is the selection of the neuron sub-components that are going to form the custom neurons.

Disadvantage:

None apart from complexity

Conclusion:

The main purpose of the proposed algorithm was to retain the simplicity of ELM by reducing the number of parameters need tuning. For this reason, the number of generations and the mutation rate of the GA are no longer user defined but they are automatically adjusted. The algorithm is able to exploit the fitness at each generation, in order to take vital decisions for the evolution process. These include the switching criterion between the uniform crossover and the proposed intelligent neuron ranked masked crossover operator, the mutation rate and the termination of the evolution process. Finally, by adapting the algorithm to work in multi-CPU environments, we further reduced the computationally intensive evolution process and we made SA-He-HyELM to be able to work with large datasets.

Name: M Manonmana Udupa

SRN: PES2UG20CS181

Section: C

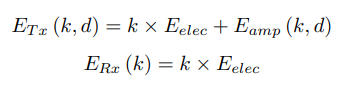
**Paper 1: *Energy-Efficient Routing Using Genetic Algorithm in Cluster-Based Wireless Sensor Networks***

**Authors**: Jaemin Kim, Younghak Kim and Younghwan Yoo

This paper proposes an energy-efficient routing protocol for cluster-based wireless sensor networks (WSNs). Forming clusters based on the location of nodes, the proposed method selects the cluster head (CH) of each cluster and discovers a route to base station (BS).

Network Model:

First, the energy cost of transmitting and receiving data is calculated.



Where,

k-no of bits of data

Eelec – sum of energy dissipated per bit

d – distance between nodes

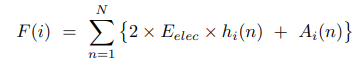
Eamp – Amplification energy (depends upon distance d)

Proposed Method

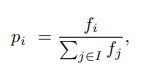
- WSNs typically use the FCM or K-means for clustering.

-The basic data structure used in the GA is called **chromosome**. Chromosome CCH is used for CH selection in the cluster and chromosome CP is used for shortest path discovery.

-The individual evaluation function in GA is the fitness function. Lower the fitness function, better the chances of the node being selected for the next generation.



-The selection probability is inversely proportional to the fitness value.



-During crossover, new off springs are born by just exchanging the CCH and CP chromosome values between the two parents.

Performance Evaluation:

Compared with PEGASIS and BCDCP, this method showed almost the same performance in the 100 × 100 m2 network. However, as the network size grew, the performance gap remarkably increased. PEGASIS constructs the chain between nodes using the greedy method, and BCDCP constructs the minimum spanning tree between CHs. Although both of them are also hierarchical methods, they cannot outperform efficient cluster-based methods in a large network where the average distance between two nodes is usually long. The transmission over long distance requires excessive energy consumption.

Merits:

The simulation result substantiates that this method outperforms LEACH, PEGASIS, and BCDCP. The performance gap grows as the sensor field increases.

Performs the CH selection and the route discovery in one phase.

Demerits:

Performance decreases when the sensor field is small. LEACH has a better performance in this condition as the CHs communicate with the BS in a single hop.

Random initialization of CP may create a cyclic path.

Reference: <https://ieeexplore.ieee.org/document/8891489>

**Paper 2: *Deep Reinforcement Learning using Genetic***

***Algorithm for Parameter Optimization***

**Authors**: Adarsh Sehgal, Hung Manh La, Sushil J. Louis, Hai Nguyen

Reinforcement learning (RL) enables agents to take decision based on a reward function. However, in the process of learning, the choice of values for learning algorithm parameters can significantly impact the overall learning process. In this paper, we use a genetic algorithm (GA) to find the values of parameters used in Deep Deterministic Policy Gradient (DDPG) combined with Hindsight Experience Replay (HER), to help speed up the learning agent.

The main contribution of this paper is a demonstration of better final performance at several manipulation tasks using a Genetic Algorithm (GA) to find DDPG and HER parameter values that lead more quickly to better performance at these tasks.

In this paper, model-free RL with continuous action spaces and deep neural network is used.

A genetic algorithm is used to find the best parameters for DDGP+HER algorithm and compare it with original parameters and hence compute their success rate.

Method:

The genetic algorithm searches through the space of parameter values used in DDPG + HER for values that maximize task performance and minimize the number of training epochs. We target the following parameters:

- discounting factor γ

- polyak-averaging coefficient τ

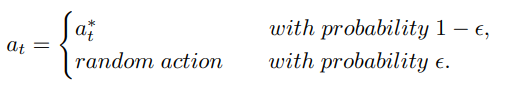
-learning rate for critic network αcritic

- learning rate for actor network αactor

- percent of times a random action is taken E

- standard deviation of Gaussian noise

Experiments show that adjusting the values of parameters did not increase or decrease the agents learning in a linear or easily discernible pattern. So, a simple hill climber will probably not do well in finding optimized parameters. Since GAs were designed for such poorly understood problems, GA is used to optimize these parameter values.



The parents are probabilistically based on rank, which is in turn decided based on the relative fitness (performance). Children are then generated using uniform crossover. Flip mutation is used with probability of mutation to be 0.1. A binary chromosome is used to encode each parameter and concatenate the bits to form a chromosome for the GA.

Fitness which is the inverse of number of epochs because GA always maximizes the objective function and this converts the minimization of number of epochs to a maximization problem.

Merits:

GAs find good parameter values leading to better task performance, faster.

Finds learning algorithm parameters that needs fewer epochs.

HER does great with extremely sparse rewards and is also significantly better for sparse rewards than shaped ones.

Gives better success rates.

Demerits:

Contains Gaussian noise which needs to be reduced.

Conclusion:

In this paper, we studied initial results that demonstrated that a genetic algorithm can tune reinforcement learning algorithm parameters to achieve better performance, faster at six manipulation tasks.

Initial results bore out the assumption that GAs are a good fit for such parameter optimization and the results on the six manipulation tasks show that the GA can find parameter values that lead to faster learning and better (or equal) performance for the chosen tasks.

Reference: https://ieeexplore.ieee.org/document/8675632

**Paper 3: *Applying m-Mutation Operator in Genetic***

***Algorithm to Solve Permutation Problems***

**Authors**: Devasenathipathi N. Mudaliar and Nilesh K Modi

Keywords- Mutation Operator, Genetic Algorithm, Permutation Problem, Fitness Function, Travelling Salesman Problem, Heuristic Technique

Routing problems, scheduling problems, transportation scheduling problems are interesting problems and variants of Permutation problems. The intention of solving these problems is to find a better path (solution) among enormous, feasible, available solutions. The better (or best) solution should provide a cost-effective path which would enable a anyone (or device) to travel to all the given cities (location) one and only once and finally return to the starting city(location). Approaches like Brute Force would not work since it is not feasible to calculate cost for all the possible paths (because rise in the number of cities would exponentially increase the number of all possible permutation of cities).

The genetic algorithm is a cycle of generations produced one after another. It is stopped after a satisfactory (fit) solution is obtained or a sufficient number of generations have passed without any improvement.

Every chromosome produced is tested against a fitness function to know how fit is a chromosome. This means every chromosome is awarded a fitness score.

Mutation operator ensures that the solution does not get trapped in to local maxima. The newly mutated and the rest child chromosomes produced after crossover form the next generation.

Method:

The proposed mutation operator is based on three cut points means every solution sent to mutation is divided into four parts.

The fitness function is maximization function means the more the fitness value the fitter is the chromosome.

The below steps were followed to undertake the actual experimental work:

1. Random population initialization with 1000 solutions.

2. Selection of fitter 500 chromosomes using tournament selection method.

3. Cloning the selected chromosomes and producing two list of selected chromosomes.

4. Mating each chromosome with from list 1 with a random chromosome from list 2 using partially mapped crossover.

5. Mutate the chromosome (as per mutation rate) using the proposed m-mutation operator.

6. Repeat from steps 2 to 5 until acceptable solutions are searched or the improvement in the results have stopped after consecutive generation.

Merits:

The efficiency of the proposed mutation operator is compared with the efficiency of existing mutation operators in solving the same permutation problem and the results are encouraging.

Genetic algorithm solves permutation problem by quickly searching the search space.

The mutation operator is better able to bring variation in the population which helps for better evolution.

Reverse Sequence mutation and Partial Shuffle mutation performed well compared to others in solving the traveling salesman problem.

Swap mutation performed much better compared to the standard mutation even devoid of crossover operator.

There was very less gap in the fitness value obtained by the best mutation operator and the rest mutation operators.

Demerits:

Traditional crossover operators cannot be used to perform crossover of permutation problem solutions (since ordering is important and no genes should be missed).

Conclusion:

The performance of the proposed mutation operator is weighed against the performance of other existing mutation operators viz. displacement mutation, exchange mutation and inverse mutation.

From this experiment ,the proposed mutation operator was good in searching good solutions although displacement mutation searched better solutions faster.

The proposed mutation operator could be implemented in any variants of permutation problems like bin packing problem, graph coloring problem, traveling salesman problem, etc.

Reference: https://ieeexplore.ieee.org/abstract/document/8878867

Name: Hrishikesh B H

SRN: PES2UG20CS144

Section: C

Paper 1:

Journal: A hybrid posture detection framework:

Integrating machine learning and deep neural

networks

Authors: Sidrah Liaqat, Kia Dashtipour, Kamran Arshad, Khaled Assaleh, Naeem Ramzan

Models: DL (1DCNN, 2D-CNN, LSTM, BiLSMT) and ML (random

forest, KNN, Naive Bayes, decision tree, LDA, QDA and

SVM)

Logic and principles: In order to classify the posture prediction, standard (logistic

regression, random forest, KNN, Naive Bayes, decision tree,

linear discriminant analysis, quadratic discriminant analysis

and SVM) and deep learning classifiers such as (1D-CNN,

2D-CNN, LSTM, BiLSTM) are trained. We extracted different features including skew, percentile, SR, SD, mean and

kurtosis. It is worth to mention that, there are total thirteen

experiments have been done. In addition, the 10-fold crossvalidation is used to perform the experiments.

Advantage and Disadvantages:

Advantages(random forest):

1. Random Forest is based on the bagging algorithm and uses Ensemble Learning technique. It creates as many trees on the subset of the data and combines the output of all the trees. In this way it reduces overfitting problem in decision trees and also reduces the variance and therefore improves the accuracy.

2. Random Forest can be used to solve both classification as well as regression problems.

3. Random Forest works well with both categorical and continuous variables.

4. Random Forest can automatically handle missing values.

5. No feature scaling required: No feature scaling (standardization and normalization) required in case of Random Forest as it uses rule based approach instead of distance calculation.

6. Handles non-linear parameters efficiently: Non linear parameters don't affect the performance of a Random Forest unlike curve based algorithms. So, if there is high non-linearity between the independent variables, Random Forest may outperform as compared to other curve based algorithms.

7. Random Forest can automatically handle missing values.

8. Random Forest is usually robust to outliers and can handle them automatically.

9. Random Forest algorithm is very stable. Even if a new data point is introduced in the dataset, the overall algorithm is not affected much since the new data may impact one tree, but it is very hard for it to impact all the trees.

10. Random Forest is comparatively less impacted by noise.

Disadvantages(random forest):1. Complexity: Random Forest creates a lot of trees (unlike only one tree in case of decision tree) and combines their outputs. By default, it creates 100 trees in Python sklearn library. To do so, this algorithm requires much more computational power and resources. On the other hand decision tree is simple and does not require so much computational resources.

2. Longer Training Period: Random Forest require much more time to train as compared to decision trees as it generates a lot of trees (instead of one tree in case of decision tree) and makes decision on the majority of votes.

Advantages(knn):

1.No Training Period- KNN modeling does not include training period as the data itself is a model which will be the reference for future prediction and because of this it is very time efficient in term of improvising for a random modeling on the available data.

2.Easy Implementation- KNN is very easy to implement as the only thing to be calculated is the distance between different points on the basis of data of different features and this distance can easily be calculated using distance formula such as- Euclidian or Manhattan

3.As there is no training period thus new data can be added at any time since it wont affect the model.

Disadvantages(knn):

1.Does not work well with large dataset as calculating distances between each data instance would be very costly.

2.Does not work well with high dimensionality as this will complicate the distance calculating process to calculate distance for each dimension.

3.Sensitive to noisy and missing data

4.Feature Scaling- Data in all the dimension should be scaled (normalized and standardized) properly.

Advantages(naive bayes):

1.Simple to Implement. The conditional probabilities are easy to evaluate.

2.Very fast – no iterations since the probabilities can be directly computed. So this technique is useful where speed of training is important.

3.If the conditional Independence assumption holds, it could give great results.

Disadvantages(naive bayes):

1.Conditional Independence Assumption does not always hold. In most situations, the feature show some form of dependency.

2.Zero probability problem : When we encounter words in the test data for a particular class that are not present in the training data, we might end up with zero class probabilities. See the example below for more details: P(bumper | Ham) is 0 since bumper does not occuer in any ham (non-spam) documents in the training data.

Advantages(decision tree):

1.Compared to other algorithms decision trees requires less effort for data preparation during pre-processing.

2.A decision tree does not require normalization of data.

3.A decision tree does not require scaling of data as well.

4.Missing values in the data also do NOT affect the process of building a decision tree to any considerable extent.

5.A Decision tree model is very intuitive and easy to explain to technical teams as well as stakeholders.

Disadvantages(decision tree):

1.A small change in the data can cause a large change in the structure of the decision tree causing instability.

2.For a Decision tree sometimes calculation can go far more complex compared to other algorithms.

3.Decision tree often involves higher time to train the model.

4.Decision tree training is relatively expensive as the complexity and time has taken are more.

5.The Decision Tree algorithm is inadequate for applying regression and predicting continuous values.

Advantages(svm):

1.SVM works relatively well when there is a clear margin of separation between classes.

2.SVM is more effective in high dimensional spaces.

3.SVM is effective in cases where the number of dimensions is greater than the number of samples.

4.SVM is relatively memory efficient

Disadvantages(svm):

1.SVM algorithm is not suitable for large data sets.

2.SVM does not perform very well when the data set has more noise i.e. target classes are overlapping.

3.In cases where the number of features for each data point exceeds the number of training data samples, the SVM will underperform.

4.As the support vector classifier works by putting data points, above and below the classifying hyperplane there is no probabilistic explanation for the classification

Doi: <http://dx.doi.org/10.1109/JSEN.2021.3055898>

Paper 2:

Journal: Shortcut Learning in Deep Neural Networks

Authors: Robert Geirhos, Jorn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge & Felix A. Wichmann

Models: DNN, NLP(BERT model),Agent-based (Reinforcement) Learning, Fairness and Algorithmic decision-making

Logic and Principles:

The four components listed below determine the inductive bias of a model and dataset: the set of

assumptions that influence which solutions are learnable, and how readily they can be learned.

Although in theory DNNs can approximate any function (given potentially infinite capacity)

, their inductive bias plays an important role for the types of patterns they prefer to learn

given finite capacity and data.

• Structure: architecture. Convolutions make it harder for a model to use location—a

prior that is so powerful for natural images that even untrained networks can be

used for tasks like image inpainting and denoising . In Natural Language Processing,

transformer architectures use attention layers to understand the context by modelling

relationships between words. In most cases, however, it is hard to understand the implicit

priors in a DNN and even standard elements like ReLU activations can lead to unexpected

effects like unwarranted confidence.

• Experience: training data. As discussed in , shortcut opportunities are present

in most data and rarely disappear by adding more data. Modifying

the training data to block specific shortcuts has been demonstrated to work for reducing

adversarial vulnerability and texture bias.

• Goal: loss function. The most commonly used loss function for classification, crossentropy, encourages DNNs to stop learning once a simple predictor is found; a modification can force neural networks to use all available information. Regularisation terms

that use additional information about the training data have been used to disentangle intended features from shortcut features.

• Learning: optimisation. Stochastic gradient descent and its variants bias DNNs towards

learning simple functions. The learning rate influences which patterns

networks focus on: Large learning rates lead to learning simple patterns that are shared

across examples, while small learning rates facilitate complex pattern learning and memorisation. The complex interactions between training method and architecture

are poorly understood so far; strong claims can only be made for simple cases

Advantages and Disadvantages:

1. DNN:

Advantages:

1.Features are automatically deduced and optimally tuned for desired outcome. Features are not required to be extracted ahead of time. This avoids time consuming machine learning techniques.

2.Robustness to natural variations in the data is automatically learned.

The same neural network based approach can be applied to many different applications and data types.

3.Massive parallel computations can be performed using GPUs and are scalable for large volumes of data. Moreover it delivers better performance results when amount of data are huge.

4.The deep learning architecture is flexible to be adapted to new problems in the future.

Disdvantages:

1.It requires very large amount of data in order to perform better than other techniques.

2.It is extremely expensive to train due to complex data models. Moreover deep learning requires expensive GPUs and hundreds of machines. This increases cost to the users.

3.There is no standard theory to guide you in selecting right deep learning tools as it requires knowledge of topology, training method and other parameters. As a result it is difficult to be adopted by less skilled people.

4.It is not easy to comprehend output based on mere learning and requires classifiers to do so. Convolutional neural network based algorithms perform such tasks.

2.NLP

Advantages:

1.NLP helps users to ask questions about any subject and get a direct response within seconds.

2.NLP offers exact answers to the question.

It does not offer unnecessary and unwanted information.

3.NLP helps computers to communicate to humans(Any languages).

4. It is very time efficient.

Disadvantages:

1.The efficiency of documentation processes.

2.Accuracy of documentation.

3.Identify the information from large databases.

3. Agent bases (Reinforcement learning):

Advantages:

1.Reinforcement learning can be used to solve very complex problems that cannot be solved by conventional techniques.

2.This technique is preferred to achieve long-term results, which are very difficult to achieve.

3.This learning model is very similar to the learning of human beings. Hence, it is close to achieving perfection.

4.The model can correct the errors that occurred during the training process.

Once an error is corrected by the model, the chances of occurring the same error are very less.

5.It can create the perfect model to solve a particular problem.

6.Robots can implement reinforcement learning algorithms to learn how to walk.

In the absence of a training dataset, it is bound to learn from its experience.

7.Reinforcement learning models can outperform humans in many tasks. 8.DeepMind’s AlphaGo program, a reinforcement learning model, beat the world champion Lee Sedol at the game of Go in March 2016.

9.Reinforcement learning is intended to achieve the ideal behavior of a model within a specific context, to maximize its performance.

10.It can be useful when the only way to collect information about the environment is to interact with it.

11.Reinforcement learning algorithms maintain a balance between exploration and exploitation. Exploration is the process of trying different things to see if they are better than what has been tried before. Exploitation is the process of trying the things that have worked best in the past. Other learning algorithms do not perform this balance.

Disadvantages:

1.Reinforcement learning as a framework is wrong in many different ways, but it is precisely this quality that makes it useful.

2.Too much reinforcement learning can lead to an overload of states, which can diminish the results.

3.Reinforcement learning is not preferable to use for solving simple problems.

4.Reinforcement learning needs a lot of data and a lot of computation. It is data-hungry. That is why it works really well in video games because one can play the game again and again and again, so getting lots of data seems feasible.

5.Reinforcement learning assumes the world is Markovian, which it is not. The Markovian model describes a sequence of possible events in which the probability of each event depends only on the state attained in the previous event.

6.The curse of dimensionality limits reinforcement learning heavily for real physical systems. According to Wikipedia, the curse of dimensionality refers to various phenomena that arise when analyzing and organizing data in high-dimensional spaces that do not occur in low-dimensional settings such as the three-dimensional physical space of everyday experience.

7.Another disadvantage is the curse of real-world samples. For example, consider the case of learning by robots. The robot hardware is usually very expensive, suffers from wear and tear, and requires careful maintenance. Repairing a robot system is costs a lot.

8.To solve many problems of reinforcement learning, we can use a combination of reinforcement learning with other techniques rather than leaving it altogether. One popular combination is Reinforcement learning with Deep Learning.

Doi: <https://doi.org/10.1038/s42256-020-00257-z>

Paper 3

Path Loss Prediction Based on Machine Learning Techniques: Principal Component Analysis, Artificial Neural Network, and Gaussian Process

Authors: Han-Shin Jo, Chanshin Park, Eunhyoung Lee , Haing Kun Choi and Jaedon Park

Models: Artificial neural network (ANN); principle component analysis (PCA); Gaussian process; multi-dimensional regression; shadowing; feature selection

Logic and Principles:

Path loss models are classified into empirical, deterministic, and semi-deterministic models. The empirical models are based on the measurements and use statistical characteristics. This model is a simple way of estimating path loss, not taking all of the many parameters that determine path loss into account. Distance is the most essential parameter of path loss model and its physical effect on the path loss is mathematically expressed using a linear log-distance function in literature. Thus, linear log-distance path loss plus shadowing model is used as a baseline for most empirical models and given by



where d is the length of the path and PL0 is the path loss at the close-in reference distance d0, which is generally determined by the measuring points close to the transmitter. n is the path loss exponent representing the rate of change of path loss as the distance increases. The shadow fading denoted by Xσ is a Gaussian random variable with zero mean and σ standard deviation in dB

PCA for extracting key features. Dimensionality reduction is justified from the fact that the actual dimension might be larger than the intrinsic dimension. Then, the objectives of dimensionality reduction are to convert input data into a reduced representation set of features, while keeping as much relevant information as possible. We mainly develop a combined path loss and shadowing model, where ANN multilayer perceptron (ANN-MLP) learns path loss variations dependent on the input data with reduced features. For analysing shadowing effect, we use Gaussian Process (GP) priors for extracting variance over training data. Based on generalized variance bounds, we can give more accurate confidence level over the path loss mean value. Finally, we attempt to combine ANN and Gaussian process models to build an optimal model in two key aspects: mean path loss value and shadowing effect variance

Advantages and Disadvantages:

Advantages (Artificial neural network (ANN)):

1.Artificial neural networks have the ability to provide the data to be processed in parallel, which means they can handle more than one task at the same time.

2.Artificial neural networks have been in resistance. This means that the loss of one or more cells, or neural networks, influences the performance of Artificial Neural networks.

3.Artificial neural networks are used to store information on the network so that, even in the absence of a data pair, it does not mean that the network is not generating results.

4.Artificial neural networks are gradually being broken down, which means that they will not suddenly stop working and these networks are gradually being broken down.

5.We can train ANN’s that these networks learn from past events and make decisions.

Disadvantages (ANN):

1.As we mentioned before, with ANN arms hanging along with the execution of parallel processing, and so they need processors that support parallel processing, so the ANNs are dependent on the hardware.

2.Since it’s similar to the functionality of the human brain, we may not be able to determine what is the proper network structure of an Artificial Neural network.

3.Not only do artificial neural networks, but also the statistical models can be trained with only numeric data, so it makes it very difficult for ANN to understand the problem statement.

4.When an artificial neural network that provides a solution to the problem statements that we really don’t know on what basis it will give the solution, and this time, ANN is not reliable

Advantages (principle component analysis (PCA)):

**1.Removes Correlated Features:** In a real-world scenario, this is very common that you get thousands of features in your dataset. You cannot run your algorithm on all the features as it will reduce the performance of your algorithm and it will not be easy to visualize that many features in any kind of graph. So, you MUST reduce the number of features in your dataset.   
  
You need to find out the correlation among the features (correlated variables). Finding correlation manually in thousands of features is nearly impossible, frustrating and time-consuming. PCA does this for you efficiently.  
  
After implementing the PCA on your dataset, all the Principal Components are independent of one another. There is no correlation among them.  
**2. Improves Algorithm Performance:**With so many features, the performance of your algorithm will drastically degrade. PCA is a very common way to speed up your Machine Learning algorithm by getting rid of correlated variables which don't contribute to any decision making. The training time of the algorithms reduces significantly with a smaller number of features.  
  
So, if the input dimensions are too high, then using PCA to speed up the algorithm is a reasonable choice.   
  
**3. Reduces Overfitting:** Overfitting mainly occurs when there are too many variables in the dataset. So, PCA helps in overcoming the overfitting issue by reducing the number of features.  
  
**4. Improves Visualization:** It is very hard to visualize and understand the data in high dimensions. PCA transforms a high dimensional data to low dimensional data (2 dimension) so that it can be visualized easily.   
  
We can use 2D Scree Plot to see which Principal Components result in high variance and have more impact as compared to other Principal Components.   
  
Even the simplest IRIS dataset is 4 dimensional which is hard to visualize. We can use PCA to reduce it to 2 dimensions for better visualization.

Disadvantages (PCA):

**1. Independent variables become less interpretable:** After implementing PCA on the dataset, your original features will turn into Principal Components. Principal Components are the linear combination of your original features. Principal Components are not as readable and interpretable as original features.  
  
**2. Data standardization is must before PCA:** You must standardize your data before implementing PCA, otherwise PCA will not be able to find the optimal Principal Components.   
  
For instance, if a feature set has data expressed in units of Kilograms, Light years, or Millions, the variance scale is huge in the training set. If PCA is applied on such a feature set, the resultant loadings for features with high variance will also be large. Hence, principal components will be biased towards features with high variance, leading to false results.  
  
Also, for standardization, all the categorical features are required to be converted into numerical features before PCA can be applied.  
  
PCA is affected by scale, so you need to scale the features in your data before applying PCA. Use **StandardScaler**from **Scikit Learn** to standardize the dataset features onto unit scale (mean = 0 and standard deviation = 1) which is a requirement for the optimal performance of many Machine Learning algorithms.  
  
**3. Information Loss:** Although Principal Components try to cover maximum variance among the features in a dataset, if we don't select the number of Principal Components with care, it may miss some information as compared to the original list of features.

Advantages (Gaussian process):

1.Predictions can interpolate observations

2.Predictions are probabilistic so that one can compute confidence intervals

3.Flexibility: you can use different kernels

Disadvantages (Gaussian Process):

They lose efficiency in high dimensional spaces – namely when the number of features exceeds a few dozens

Advantages (multi-dimensional regression):

There are two main advantages to analysing data using a multiple regression model. The first is the ability to determine the relative influence of one or more predictor variables to the criterion value. The real estate agent could find that the size of the homes and the number of bedrooms have a strong correlation to the price of a home, while the proximity to schools has no correlation at all, or even a negative correlation if it is primarily a retirement community.

The second advantage is the ability to identify outliers, or anomalies. For example, while reviewing the data related to management salaries, the human resources manager could find that the number of hours worked, the department size and its budget all had a strong correlation to salaries, while seniority did not. Alternatively, it could be that all of the listed predictor values were correlated to each of the salaries being examined, except for one manager who was being overpaid compared to the others.

Disadvantages (multi-dimensional regression)

Any disadvantage of using a multiple regression model usually comes down to the data being used. Two examples of this are using incomplete data and falsely concluding that a correlation is a causation.

When reviewing the price of homes, for example, suppose the real estate agent looked at only 10 homes, seven of which were purchased by young parents. In this case, the relationship between the proximity of schools may lead her to believe that this had an effect on the sale price for all homes being sold in the community. This illustrates the pitfalls of incomplete data. Had she used a larger sample, she could have found that, out of 100 homes sold, only ten percent of the home values were related to a school's proximity. If she had used the buyers' ages as a predictor value, she could have found that younger buyers were willing to pay more for homes in the community than older buyers.

In the example of management salaries, suppose there was one outlier who had a smaller budget, less seniority and with fewer personnel to manage but was making more than anyone else. The HR manager could look at the data and conclude that this individual is being overpaid. However, this conclusion would be erroneous if he didn't take into account that this manager was in charge of the company's website and had a highly coveted skillset in network security.

Doi: http://dx.doi.org/10.3390/s20071927