BHADALE GROUP OF COMPANIES
- IT AND REAL ESTATE

Sept 4 2019

AI Neural Networks, & Algorithms Catalogue

Bhadale IT Developers Pvt. Ltd | Bhadale Real Estate Developers Pvt. Ltd (registration due)

Plot No. 52, Hindwadi, Belgaum, KA, India | Mobile: +91- 9741040195 | Website: TBD

Bhadale Group of Companies

Bhadale Group of Companies consists of Bhadale IT Developers Pvt. Ltd and Bhadale Real Estate Developers Pvt Ltd.

1. **Bhadale IT Developers Pvt. Ltd** is an IT and Computer Engineering company

   This company provides consultation in areas of cutting edge technologies, research outsourcing, and software consultation related to data center and related engineering practices

2. **Bhadale Real Estate Developers Pvt. Ltd** is a Real estate company

   This company provides development of Infrastructure for IT Datacenter and allied sectors. It manages the engineering design, landscaping, civil architecture, presently serving internal projects.

Bhadale Group of Companies has aggressive programs in place to serve the niche market.

Bhadale IT Developers Pvt. Ltd Programs and Services

IT Division programs

1 Cloud Architecture
2 AI
3 Digital
4 Automation
5 R&D services
6 Engineering services
7 Mentoring Services
8 Data center services
9 Outsourcing

There are various services offered under each program, details are described below

IT Division program related services

Cloud Architecture

- Cloud Enterprise Architecture
- Cloud Business Architecture
- Cloud Information / Website data Architecture
- Cloud Solution Architecture
- Data center Virtualization and Cloud services (IaaS)
- Cloud Technical Architecture – Project specific
- Cloud ERP Solutions (SaaS, PaaS)
- Cloud Strategy and Transformation
- Cloud Systems Integration and consolidation
- Cloud Project Management
- Cloud Pre sales support
- Business Needs (RFI/RFQ/RFP assistance)

- Cloud Quality Initiatives
- Cloud Business Analysis
- Cloud Infrastructure Planning – hardware , network, storage , backup (IaaS, PaaS)
- Cloud business portfolio assessment services (workshops)

AI

- Artificial intelligence and advanced machine learning
- Intelligent applications, Intelligent things
- Conversational systems
- Mesh app and service architecture
- Adaptive security architecture

Digital

- Virtual reality and augmented reality: Brief capability, deliverables and service offering
- Digital twins
- Blockchains and distributed ledgers
- Digital technology platforms

Automation

- Robotic Process Automation
- IoT
- Manufacturing robots
- BPO call center robots
- Chatbots
- Remote workers
- Hazardous jobs robots(Mine bombs, nuclear waste, underwater, space etc)

R&D services

- PHD mentor, buddy
- BPO - Outsourced work in areas of research areas related to IT and Computer Engineering

Engineering services (Only Engg)

- Engineering services for Data centers
- Engineering services for IT Departments
- Engineering architectures, drawings, road ways, town house planning, parking, safety, outdoor maintenance, lighting etc
- All other aspects of engineering: Civil, electrical, water and sewage, safety and mechanical motors, pumps, refrigeration, cooling

Mentoring Services

- IT Mentoring
- Engineering Mentoring
- Business Mentoring
- Mentoring for specials categories based on age, and disabled
- Mentoring for special professionals like Military and Govt officials under Govt Programs

Data center services (Engg + IT)

- Data center Engineering services
- Data Center IT Services
- Data center Security services
- Data Center QA services
- Datacenter Cloud services
- Datacenter compliances services
- Data center based business solutions

Outsourcing

- Insourcing
- Outsourcing
- Near sourcing
- Cloud sourcing
- BPO services
- IT specific services
- Engineering specific services
- Training specific services

Service details for Bhadale IT Developers Pvt. Ltd

IT Division programs – Neural Network types and their usage services

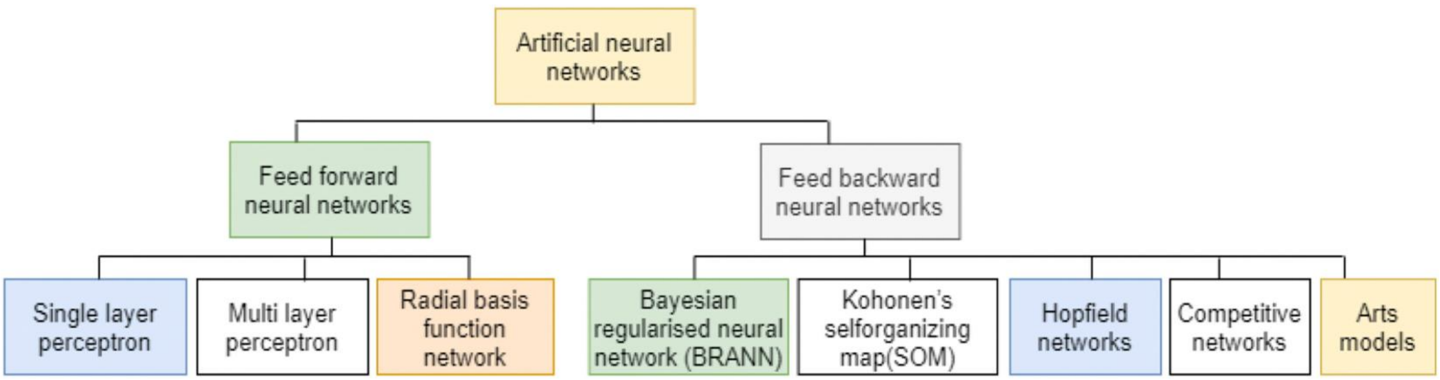

Image courtesy (The Web), no intention for copyright infringement

We have a large set of subcategories; few are mentioned below

NN No	NN Name	Key NN component features / Use cases
1 **Feed forward neural networks**	**Multilayer Perceptron (MLP)**	This is a class of feedforward artificial neural network, utilizes a supervised learning technique called backpropagation for training. This offer an approximate solution for extremely complex problem like fitness approximation. MLPs are good as classifier algorithms, and in Chemistry it predicts chemical changes in alkene structure. One of the most popular applications is Google's autonomous or self-driving cars. They are mostly used in pattern generation, pattern recognition and classification.
2	**Deep Feed Forward neural networks (DFF)**	When our data isn't linearly separable, linear ML models face problems in approximating, whereas neural networks find it easy. The hidden layers are used to increase the non-linearity and change the representation of the data for better generalization over the function. A **multilayer deep feed-forward** neural network consists of an *input layer*, one or more *hidden layers*, and an *output layer*. The backpropagation algorithm performs learning on a *multilayer feed-forward* neural network. Multilayer feed forward NW, are more accurate that single layer network

		Feedforward neural network are used for classification and regression, as well as for pattern encoding. feedforward networks to recognize handwritten characters
3	**Radial Basis Function neural networks (RBFNN)**	An RBFN performs classification by measuring the input's similarity to examples from the training set. Each RBFN neuron stores a "prototype", which is just one of the examples from the training set. When we classify a new input, each neuron computes the Euclidean distance between the input and its prototype. If the input more closely resembles class A prototypes than the class B prototypes, it is classified as class A Application of Radial Basis Function Network with a Gaussian Function of Artificial Neural Networks in Osmo-dehydration of Plant Materials. Other applications are in food and agriculture sectors for drying, extrusion, fermentation, filtration, psychrometry, thermal processing, sensory science and rheology. Radial basis function (RBF) neural network is used for the mechanical fault diagnosis of a gearbox.
4	**Recurrent Neural Networks(RNN)**	A **recurrent neural network** (**RNN**) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. RNN have memory and so context aware, however, suffer back propagation ***vanishing gradients.*** They are used in tasks such as unsegmented, connected handwriting recognition or speech recognition, and DNA sequencing. Time series as inputs are not all fed at once as in an ordinary network; however as time progresses inputs are added, that offer better predictions. They are used in Text Summarization, Text Autofill or next text recommendation, Language Translation, Call Center Analysis, and Digital Asset Management in Marketing
5	**Recursive Neural Networks (RN) / TreeNets**	A **recursive neural network** is a kind of deep neural network created by applying the same set of weights recursively over a structured input, to produce a structured prediction over variable-size input structures, or a scalar prediction on it, by traversing a given structure in topological order. RNNs have been successful, in learning sequence and tree structures in natural language processing, mainly phrase and sentence continuous representations based on word embedding, and sentiment analysis. They are used in recursive neural tensor networks for boundary segmentation, to determine word groups that are positive and negative.
6	**Long short-term memory (LSTM) networks**	**Long short-term memory** (**LSTM**) is artificial recurrent neural network (RNN) architecture. Unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points, but also entire sequences of data (such as speech or video). LSTMs were developed to deal with the exploding and vanishing gradient problems that can be encountered when training traditional RNNs. A common LSTM unit is composed of a **cell**, an **input gate**, an **output gate** and a **forget gate**. The cell remembers values over arbitrary time intervals and the three *gates* regulate the flow of information into and out of the cell. LSTM networks are used to classify, process and predict based on time series data that

		have lags of unknown duration between important events. LSTM networks have been used successfully in the following tasks 1. Language modeling - character and word level 2. Machine Translation or sequence to sequence learning 3. Image captioning (with and without attention) 4. Hand writing generation 5. Image generation using attention models 6. Question answering 7. Video to text
7	**Gated Recurrent Unit – (GRUs)**	**Gated recurrent unit**s (**GRU**s) are a gating mechanism in recurrent neural networks, introduced in 2014 by Kyunghyun Cho et al. The GRU is like a long short-term memory (LSTM) with forget gate but has fewer parameters than LSTM, as it lacks an output gate. GRU's got rid of the cell state and used the hidden state to transfer information. It also only has two gates, a reset gate and update gate. Lack of output gate makes it easier to repeat the same output for a concrete input multiple times. GRUs are currently used in sound (music), speech recognition, speech synthesis, and text generation
8	**Autoencoders (AE)**	An **autoencoder** is a type of neural network used to learn efficient data coding in an unsupervised manner. The aim of an autoencoder is to learn a representation (encoding) for a given set of data, for dimensionality reduction, which is done by training the network to ignore signal "noise". Along with the reduction side, a reconstructing side is also learnt, where the autoencoder tries to generate from the reduced encoding a representation as close as possible to its original input. Key applications are dimensionality reduction or feature learning, and for learning generative models of data. Autoencoders are used for classification, document clustering and feature compression. Another application of AE is to use it for "pretraining" a Deep Network.
9	**Variational Autoencoders (AEs)**	The idea behind a **variational autoencoder** is that instead of mapping an input to a fixed vector, it is mapped to a distribution. The only difference between the autoencoder and variational autoencoder is that vector is replaced with two different vectors one representing the mean of the distribution and the other representing the standard deviation of the distribution. VAEs, comparing to AE, compress probabilities instead of features. They are used for dimensionality reduction, information retrieval, and anomaly detection. Spatio-Temporal AutoEncoder is used for video anomaly detection
10	**Denoising AE (DAE)**	A **denoising autoencoder**, model is identical to a convolutional autoencoder. However, training and testing data are different. For training data, we add random, Gaussian noise, and test data is the original, clean image. This provides training for the denoising autoencoder to produce clean images for given noisy images.

		Deep Denoising Autoencoders(DDAE) offers drastic improvement in performance and has the capability to recognize whispered speech in Automatic Speech Recognition(ASR). This has been implemented in various smart devices such as Amazon Alexa.
11	**Sparse AE(SAE)**	In an ordinary autoencoder, it takes the input image or vector and learns code dictionary that transforms the raw input from one representation to another. Where as in **sparse autoencoders, a sparsity enforcer** directs a single layer network to learn code dictionary that minimizes the error in reproducing the input while restricting number of code words during reconstruction
12	**Markov Chains (MC)**	**MC** is a stochastic process containing random variables, transitioning from one state to another depending on certain assumptions and definite probabilistic rules**.** Random variable's transition is based on Markov property. Markov chains as a memory-less process that solely depends on the current state/action of a variable. Google's PageRank Algorithm is based on the idea of Markov chains. Markov chains are used in text generation and auto-completion applications, and typing word prediction
13	**Hopfield networks(HN)**	A **Hopfield network** is a form of recurrent artificial neural network popularized by John Hopfield in 1982, but described earlier by Little in 1974. Hopfield nets serve as content-addressable ("associative") memory systems with binary threshold nodes HN is useful for Associative memory, & Combinatorial optimization. Common applications are those where pattern recognition is useful, and Hopfield networks have been used for image detection and recognition, enhancement of X-Ray images, medical image restoration. HN uses human eye movement mechanisms (saccades) and is useful for scene analysis, including object representation and pattern recognition. HN is used in areas: • Medical imaging segmentation • Image Processing • Speech processing • Database Retrieval • Fault Tolerant Computing
14	**Boltzmann machines(BM)**	A **Boltzmann machine** (also called **stochastic Hopfield network with hidden units**) is a type of stochastic recurrent neural network and Markov random field. The Boltzmann machine is a Monte Carlo version of the Hopfield network Boltzmann machines can be seen as the stochastic, generative counterpart of Hopfield networks. They were one of the first neural networks capable of learning internal representations and are able to represent and (given sufficient time) solve difficult combinatory problems.

15	**Restricted BM (RBM)**	Invented by Geoffrey Hinton, a Restricted Boltzmann machine is an algorithm useful for dimensionality reduction, classification, regression, collaborative filtering, feature learning and topic modeling. RBMs are shallow, two-layer neural nets that constitute the building blocks of *deep-belief networks*
16	**Deep Belief NW (DBNs)**	In machine learning, a **deep belief network** (**DBN**) is a generative graphical model, composed of multiple layers of latent variables ("hidden units"), with connections between the layers but not between units within each layer When trained on a set of examples without supervision, a DBN can learn to probabilistically reconstruct its inputs. The layers then act as feature detectors. After this learning step, a DBN can be further trained with supervision to perform classification. Natural language call routing, DBN-based model gives a call–routing classification accuracy that is equal to the best of the other models. Other uses are computerized prognosis for Alzheimer’s disease, tumor segmentation, histo- pathological diagnosis, modeling to characterize differences in brain morphometry in schizophrenia and psychiatric neuroimaging
17	**Deep Convolution NW (DCN)**	A CNN uses a convolution which is the simple application of a filter to an input that results in activation. Repeated application of the same filter to an input, results in a map of activations called a feature map, indicating the locations and strength of a detected feature in an input, such as an image. A **DCN** uses more than 2 filters like RGB filters for processing an image. **NLP** DCN is used in NLP problems for semantic parsing, search query retrieval, sentence modeling, classification, prediction and other traditional NLP tasks **Video analysis** One approach is to treat space and time as equivalent dimensions of the input and perform convolutions in both time and space. Another way is to fuse the features of two convolutional neural networks, one for the spatial and one for the temporal stream. Long short-term memory (LSTM) recurrent units are typically incorporated after the CNN to account for inter-frame or inter-clip dependencies. **Drug discovery** AtomNet was used to predict novel candidate biomolecules for multiple disease targets, most notably treatments for the Ebola virus and multiple sclerosis **Classifying remote scenes** Deep convolutional neural networks (CNNs) have been widely used to obtain high-level representation in various computer vision tasks. Classifying remote scenes according to a set of semantic categories is a very challenging problem, because of high intra-class variability and low interclass distance

18	**De Convolution NW (DNs)**	The process of reversing a convolution is generally referred to as deconvolution. This is achieved through deconvolutional layers. A deconvolutional layer utilizes the same receptive fields from the convolution layer that it is about to reverse. The fields are then flipped 180° horizontally and vertically. The process of deconvolution is also referred as to the transposed convolution. Deconvolutional networks help to find lost features or signals that may have previously not been deemed important to a convolutional neural network's task A deconvolution may result in up-samplings depending on how paddings are applied to the images. The critical part of performing a deconvolution is to ensure that the image should return to its size before the convolution. Used for restoring lost colors, hues, and signals that was convoluted
19	**Deep Convolution Inverse Graphics NW (DCIGN)**	The DC-IGN model is composed of multiple layers of convolution and de-convolution operators and is trained using the Stochastic Gradient Variational Bayes (SGVB) algorithm. Given a static face image, model can re-generate the input image with different pose, lighting or even texture and shape variations from the base face. Mostly used for image processing, these networks can process images that they have not been trained with previously. These nets, due to their abstraction levels, can remove certain objects from image, re-paint it, or replace horses with zebras like the famous CycleGAN did.
20	**Generative Adversal NW (GAN)**	A **generative adversarial network** (**GAN**) is a class of machine learning systems invented by Ian Goodfellow and his colleagues in 2014. Two neural networks contest with each other in a game (in the sense of game theory, often but not always in the form of a zero-sum game). The *generative* network generates candidates while the *discriminative* network evaluates them. The contest operates in terms of data distributions. Typically, the generative network learns to map from a latent space to a data distribution of interest, while the discriminative network distinguishes candidates produced by the generator from the true data distribution. The generative network's training objective is to increase the error rate of the discriminative network (i.e., "fool" the discriminator network by producing novel candidates that the discriminator thinks are not synthesized (are part of the true data distribution)) pix2pix is an excellent example of such approach. Other uses are: • Generate Examples for Image Datasets • Generate Photographs of Human Faces • Generate Realistic Photographs • Generate Cartoon Characters • Image-to-Image Translation • Text-to-Image Translation • Semantic-Image-to-Photo Translation • Face Frontal View Generation • Generate New Human Poses

		• Photos to Emojis • Photograph Editing • Face Aging • Photo Blending • Super Resolution • Photo Inpainting • Clothing Translation • Video Prediction • 3D Object Generation • Fashionable clothes • Innovative paintings
21	**Deep Convolutional Generative Adversarial Network (DCGAN)**	The core to the DCGAN architecture uses a standard CNN architecture on the discriminative model. For the generator, convolutions are replaced with upconvolutions, so the representation at each layer of the generator is actually successively larger, as it maps from a low-dimensional latent vector onto a high-dimensional image. Deep Convolutional GANs (DCGANs) are a type of GANs that employ a deep convolutional network in both- the generator & the discriminator. Ordinary GANs use adversarial training method to teach a 'generator' to create data belonging to some real-world distribution. DCGANs use the same philosophy apart from employing a convolutional architecture to process data. Used for generating images of handwritten digits
22	**Conditional GAN (cGAN)**	Although GAN models are capable of generating new random plausible examples for a given dataset, there is no way to control the types of images that are generated other than trying to figure out the complex relationship between the latent space input to the generator and the generated images. cGAN is a recently developed algorithm to solve the instability problem of GAN, and provides a very stable Nash equilibrium solution The conditional generative adversarial network, or cGAN for short, is a type of GAN that involves the conditional generation of images by a generator model. Image generation can be conditional on a class label, if available, allowing the targeted generated of images of a given type. Use of a conditional deep convolutional generative adversarial network (cDCGAN) to design nanophotonic structures.
23	**Liquid State Machine(LSM)**	A **liquid state machine** (**LSM**) is a particular kind of spiking neural network. An LSM consists of a large collection of units (called *nodes*, or *neurons*). Each node receives time varying input from external sources (the **inputs**) as well as from other nodes. Nodes are randomly connected to each other. The recurrent nature of the connections turns the time varying input into a spatio-temporal pattern of activations in the network nodes. The spatio-temporal patterns of activation are read out by linear discriminant units. The word liquid in the name comes from the analogy drawn to dropping a stone into a still body of water or other liquid. The falling stone will generate ripples in the liquid. The input (motion of the falling stone) has been converted into a spatio-temporal pattern of

		liquid displacement (ripples). Uses: Egocentric video activity recognition is quickly becoming a pertinent application area due to first person wearable devices such as body cameras or in robotics. In these application domains, real-time learning is critical for deployment beyond controlled environments (such as deep space exploration), or to learn continuously in novel scenarios
24	**Extreme Learning machine (ELM)**	Although many efforts have been paid to enhance the back-propagation algorithm, challenging issues such as local minima, time-costing in learning, and manual parameter setups still remain in the training phase and are not well addressed in the literature. These drawbacks may limit its applications in high dimensional and large data. Unlike these conventional implementations, a new learning algorithm called **e**xtreme **l**earning **m**achine (ELM) for **s**ingle-hidden **l**ayer **f**eedforward neural **n**etworks (SLFNs) which randomly chooses hidden nodes and analytically determines the output weights of SLFNs. In theory, this algorithm tends to provide good generalization performance at extremely fast learning speed. These models are able to produce good generalization performance and learn thousands of times faster than networks trained using backpropagation **Extreme learning machines** are feedforward neural networks for classification, regression, clustering, sparse approximation, compression and feature learning with a single layer or multiple layers of hidden nodes, where the parameters of hidden nodes (not just the weights connecting inputs to hidden nodes) need not be tuned. These hidden nodes can be randomly assigned and never updated (i.e. they are random projection but with nonlinear transforms), or can be inherited from their ancestors without being changed. In most cases, the output weights of hidden nodes are usually learned in a single step, which essentially amounts to learning a linear model. Due to its different learning algorithm implementations for regression, classification, sparse coding, compression, feature learning and clustering, multi ELMs have been used to form multi hidden layer networks, deep learning or hierarchical networks. Used in: ELM in High Dimensional and Large Data Applications, ELM in Image Processing, human action recognition, video applications including hand motion classification , visual tracking, video based semantic concept detection, and video watermarking, cardiac arrhythmia classification , gene cancer identification , mammographic micro calcifications detection , epileptic diagnosis , liver parenchyma and tumor detection, EEG vigilance, magnetic resonance images (MRI) data processing , gene selection , protein sequence applications , hypoglycemia prediction , and Parkinson classification, time series prediction and forecasting , terrain reconstruction and navigation , power loss analysis , company internationalization search , XML document classification and text categorization , cloud computing , activity recognition for mini wearable devices , imbalance data processing, terrain reconstruction and navigation with ELM based approaches for aiding unmanned aerial vehicles (UAVs)
25	**Echo State NW (ESN)**	The **echo state network** (**ESN**), is a recurrent neural network with a sparsely connected hidden layer (with typically 1% connectivity). The connectivity and weights of hidden neurons are fixed and randomly assigned. The weights of output neurons can be learned

		so that the network can (re)produce specific temporal patterns. The main interest of this network is that although its behaviour is non-linear, the only weights that are modified during training are for the synapses that connect the hidden neurons to output neurons. Thus, the error function is quadratic with respect to the parameter vector and can be differentiated easily to a linear system. Some publicly available implementations of ESNs are: (i) aureservoir: an efficient C++ library for various kinds of echo state networks with python/numpy bindings; and (ii) Matlab code: an efficient matlab for an echo state network.
26	**Deep Residual NW (DRN)**	A **residual neural network** (**ResNet**) is an artificial neural network (ANN) of a kind that builds on constructs known from pyramidal cells in the cerebral cortex. Residual neural networks do this by utilizing *skip connections*, or *short-cuts* to jump over some layers. Typical *ResNet* models are implemented with double- or triple- layer skips that contain nonlinearities (ReLU) and batch normalization in between. An additional weight matrix may be used to learn the skip weights; these models are known as *HighwayNets*. Models with several parallel skips are referred to as *DenseNets*. In the context of residual neural networks, a non-residual network may be described as a *plain network*. This speeds learning by reducing the impact of vanishing gradients, as there are fewer layers to propagate through. The network then gradually restores the skipped layers as it learns the feature space. Towards the end of training, when all layers are expanded, it stays closer to the manifold and thus learns faster. ResNet makes it possible to train up to hundreds or even thousands of layers and still achieves compelling performance. Uses: prediction of physical attributes from face images
27	**Kohonen NW (KN)/ SOM**	Suppose we have some pattern of arbitrary dimensions, however, we need them in one dimension or two dimensions. Then the process of feature mapping would be very useful to convert the wide pattern space into a typical feature space. Now, the question arises why do we require self-organizing feature map? The reason is, along with the capability to convert the arbitrary dimensions into 1-D or 2-D, it must also have the ability to preserve the neighbor topology A **self-organizing map** (**SOM**) or **self-organizing feature map** (**SOFM**) is a type of artificial neural network (ANN) that is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples, called a **map**, and is therefore a method to do dimensionality reduction. Self-organizing maps differ from other artificial neural networks as they apply competitive learning as opposed to error-correction learning (such as backpropagation with gradient descent), and in the sense that they use a neighborhood function to preserve the topological properties of the input space. Uses: One of the earliest and well known applications of the SOM is the phonetic typewriter of Kohonen. It is set in the field of speech recognition, and the problem is to classify phonemes in real time so that they could be used to drive a typewriter from dictation.

		Hierarchical structure composed of many self-organizing feature maps is used to control a five degree of freedom robot arm. Yet another use is in the field of speech recognition.
28	**Support Vector Matrix (SVM)**	In machine learning, **support-vector machines** (**SVMs**, also **support-vector networks**) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. SVM creates a hyperplane that separates a given dataset into classes. According to SVM, we need to find the points that lie closest to both the classes. These points are known as support vectors. In the next step, we find the proximity between our dividing plane and the support vectors. The distance between the points and the dividing line is known as margin. The aim of an SVM algorithm is to maximize this very margin. When the margin reaches its maximum, the hyperplane becomes the optimal one. Uses: • **Face detection** – SVM classify parts of the image as a face and non-face and create a square boundary around the face. • **Text and hypertext categorization** – SVMs allow Text and hypertext categorization for both inductive and transductive models. They use training data to classify documents into different categories. It categorizes on the basis of the score generated and then compares with the threshold value. • **Classification of images** – Use of SVMs provides better search accuracy for image classification. It provides better accuracy in comparison to the traditional query-based searching techniques. • **Bioinformatics** – It includes protein classification and cancer classification. We use SVM for identifying the classification of genes, patients on the basis of genes and other biological problems. • **Protein fold and remote homology detection** – Apply SVM algorithms for protein remote homology detection. • **Handwriting recognition** – We use SVMs to recognize handwritten characters used widely. • **Generalized predictive control (GPC)** – Use SVM based GPC to control chaotic dynamics with useful parameters.
29	**Neural Turing Machine (NTM)**	A **Neural Turing machine** (**NTMs**) is a recurrent neural network model. NTMs combine the fuzzy pattern matching capabilities of neural networks with the algorithmic power of programmable computers. An NTM has a neural network controller coupled to external memory resources, which it interacts with through attentional mechanisms. The memory interactions are differentiable end-to-end, making it possible to optimize them using gradient descent. An NTM with a long short-term memory (LSTM) network controller can infer simple algorithms such as copying, sorting, and associative recall from examples alone Uses: NTM can learn simple algorithms, used in language modeling / autocomplete, and answering after storytelling

30	**Bidirectional Recurrent Neural Networks (BRNN)**	**Bidirectional Recurrent Neural Networks** (**BRNN**) connect two hidden layers of opposite directions to the same output. With this form of generative deep learning, the output layer can get information from past (backwards) and future (forward) states simultaneously. Invented in 1997 by Schuster and Paliwal, BRNNs were introduced to increase the amount of input information available to the network. For example, multilayer perceptron (MLPs) and time delay neural network (TDNNs) have limitations on the input data flexibility, as they require their input data to be fixed. On the contrary, BRNNs do not require their input data to be fixed. Moreover, their future input information is reachable from the current state BRNN are especially useful when the context of the input is needed. For example, in handwriting recognition, the performance can be enhanced by knowledge of the letters located before and after the current letter. Other applications of BRNN include : • Speech Recognition (Combined with Long short-term memory) • Translation • Handwritten Recognition • Protein Structure Prediction • Part-of-speech tagging • Dependency Parsing • Entity Extraction
31	**Combination of a CNN and a BRNN**	In a mixed CNN and RNN architecture the positive features of a RNN are used to improve the CNN. Liang and Hu have described architecture for object detection and for scene labeling. The combined network is called RCNN. The key modules of this RCNN are the recurrent convolution layers (RCL), which introduce recurrent connection into a convolution layer. With these connections the network can evolve over time though the input is static and each unit is influenced by its neighboring units. This property integrates the context information of an image, which is important for object detection. One example network is trained with BPTT and therefore with an unfolding algorithm. The unfolding facilitates the learning process and sharing weights reduce the parameters. Used in audio surveillance systems, object detection, scene labeling, and Multi-label Text Categorization
32	**Sequence-To-Sequence Models**	**Seq2seq** was first introduced for machine translation, by Google, consists of two recurrent neural networks. As the name suggests, seq2seq takes as input a sequence of words (sentence or sentences) and generates an output sequence of words. It does so by use of the recurrent neural network (RNN). Although the vanilla version of RNN is rarely used, its more advanced version i.e. LSTM or GRU are used. It mainly has two components i.e *encoder* and *decoder*, and hence sometimes it is called the **Encoder-Decoder Network**. Examples of sequence to sequence problems can be: **Machine Translation,** & **Video Captioning**

<table>
<tr><td>33</td><td>Deep stacking network (DSN)</td><td>
The central idea of the DSN design relates to the concept of stacking, where simple modules of functions or classifiers are composed first and then they are stacked on top of each other in order to learn complex functions or classifiers.
Deep stacking networks (DSN) are a special type of deep model equipped with parallel and scalable learning. SVM-DSN can iteratively extract data representations layer by layer as a deep neural network but with parallelizability, and from a local view, each stacked SVM can converge to its optimal solution and obtain the support vectors, which compared with neural networks, could lead to interesting improvements in anti-saturation and interpretability. Experimental results on both image and text data sets demonstrate the excellent performances of SVM-DSN compared with some competitive benchmark models
Design of deep stacking network (DSN) is stacked each of base module which using a simple form of the multilayer perceptron. Using DSN is suitable for complex data like microarray dataset (for breast cancer diagnosis) because it has a deep architecture (deep learning). Furthermore, DSN model does not use stochastic gradient descent which is difficult to be implemented on large scale of machine learning.
Deep Learning Architectures, abridge the components of Deep Boltzmann Machines (DBM), Deep Stacking Networks (DSN), Compound Hierarchical Deep Models (CHDM), Deep Convolutional Neural Network (DCNN) and Deep Belief Network (DBN) their learning algorithms.
</td></tr>
<tr><td>34</td><td>General regression neural network (GRNN)</td><td>
Generalized regression neural network (GRNN) is a variation to radial basis neural networks . GRNN represents an improved technique in the neural networks based on the nonparametric regression. The idea is that every training sample will represent a mean to a radial basis neuron
Similar to RBFNN, GRNN has the following advantages:

Single-pass learning so no backpropagation is required.
High accuracy in the estimation since it uses Gaussian functions.
It can handle noises in the inputs.

The main disadvantages of GRNN are:

Its size can be huge, which would make it computationally expensive.
There is no optimal method to improve it.

GRNN was used in different applications related to modeling, system identification, prediction, and control of dynamic systems including: feedback linearization controller, HVAC process identification and control, modeling and monitoring of batch processes, cooling load prediction for buildings, fault diagnosis of a building’s air handling unit, intelligent control, optimal control for variable-speed wind generation systems, annual power load forecasting model, vehicle sideslip angle estimation, detection of time-varying inter-turn short circuit in a squirrel cage induction machine, system identification of nonlinear rotorcraft heave mode, and modeling of traveling wave ultrasonic motors.
</td></tr>
</table>

35	**Hierarchical RNN**	HRNNs are a class of stacked RNN models designed with the objective of modeling hierarchical structures in sequential data (texts, video streams, speech, programs, etc.). In context of texts, these hierarchies could constitute characters at the lowest level combining to form words, words combining to form sentences, while in the context of computer programs; these could refer to modules calling a sub-module. The key idea, therefore, lies in updating the weights of neurons belonging to different layers of stack corresponding to the layers of hierarchies in the data. HRNNs can learn across multiple levels of temporal hierarchy over a complex sequence. Usually, the first recurrent layer of an HRNN encodes a sentence (e.g. of word vectors) into a sentence vector. The second recurrent layer then encodes a sequence of such vectors (encoded by the first layer) into a document vector. This document vector is considered to preserve both the word-level and sentence-level structure of the context. Uses: predict scene graphs for images, Information Extraction from Lawsuit Documents, document modeling
36	**Neocognitron**	The **neocognitron** is a hierarchical, multilayered artificial neural network proposed by Kunihiko Fukushima in 1979. It has been used for handwritten character recognition and other pattern recognition tasks, and served as the inspiration for convolutional neural networks The neocognitron consists of multiple types of cells, the most important of which are called *S-cells* and *C-cells.* The local features are extracted by S-cells, and these features' deformation, such as local shifts, is tolerated by C-cells. Local features in the input are integrated gradually and classified in the higher layers. Uses: Situation analysis, rotation-invariant character recognition, & rotated pattern recognition
37	**Spiking neural networks (SNN)**	**Spiking neural networks** (**SNNs**) are artificial neural networks that more closely mimic natural neural networks. In addition to neuronal and synaptic state, SNNs incorporate the concept of time into their operating model. The idea is that neurons in the SNN do not fire at each propagation cycle (as it happens with typical multi-layer perceptron networks), but rather fire only when a membrane potential – an intrinsic quality of the neuron related to its membrane electrical charge – reaches a specific value. When a neuron fires, it generates a signal that travels to other neurons which, in turn, increase or decrease their potentials in accordance with this signal. SNNs can model the central nervous system of biological organisms, such as an insect seeking food without prior knowledge of the environment, study the operation of biological neural circuits, real-time image and audio processing , spiking neural system to anticipate programming unwavering quality
38	**Multilayer kernel machines (MKM)**	Multilayer Kernel Machines (MKMs) is an attempt to build a kernel machine architecture with **multiple layers of feature extraction**. It composed of many layers of kernel PCA based feature extraction units with support vector machine having arc-cosine kernel as the final classifier

		Applications to optical character recognition and DNA analysis, traffic signal recognition .
39	**Neuro-fuzzy**	Neuro-fuzzy hybridization results in a hybrid intelligent system that synergizes these two techniques by combining the human-like reasoning style of fuzzy systems with the learning and connectionist structure of neural networks. The advantages of a combination of ANN and Fuzzy Inference system (FIS) are obvious . One variety is Adaptive neuro-fuzzy inference systems (ANFIS) The strength of neuro-fuzzy systems involves two contradictory requirements in fuzzy modeling: interpretability versus accuracy. In practice, one of the two properties prevails. The neuro-fuzzy in fuzzy modeling research field is divided into two areas: linguistic fuzzy modeling that is focused on interpretability, mainly the Mamdani model; and precise fuzzy modeling that is focused on accuracy, mainly the Takagi-Sugeno-Kang (TSK) model. Application in **dialysis** using an adaptive-network-based fuzzy inference system (ANFIS) for the modeling and predicting important variables in hemodialysis process, for industrial automation, control, Predict the Removal of Pb-lead(II) Ions from the Aqueous Solution by Using Magnetic Graphene/Nylon 6 Another well-known example is **insect visual navigation** which can be achieved by combining simple rules such as “slow down when approaching an object”. We can abstract from the biological systems by representing such heuristics using fuzzy reasoning. Likewise, the many nonlinear relationships inherent in vision which are difficult to represent analytically can be more easily expressed within neural networks
40	**Hierarchical temporal memory (HTM)**	HTM is based on neuroscience and the physiology and interaction of pyramidal neurons in the neocortex of the mammalian (in particular, human) brain. At the core of HTM are learning algorithms that can store, learn, infer and recall high-order sequences. Unlike most other machine learning methods, HTM learns (in an unsupervised fashion) time-based patterns in unlabeled data on a continuous basis. HTM is robust to noise, and it has high capacity, meaning that it can learn multiple patterns simultaneously. When applied to computers, HTM is well suited for prediction, anomaly detection, classification and ultimately sensorimotor applications. A typical HTM network is a tree-shaped hierarchy of *levels* (which should *not* be confused with the "*layers*" of the neocortex, as described below) that are composed of smaller elements called *region*s (or nodes). A single level in the hierarchy possibly contains several regions. Higher hierarchy levels often have fewer regions. Higher hierarchy levels can reuse patterns learned at the lower levels by combining them to memorize more complex patterns Cortical learning algorithms are currently being offered as commercial SaaS by Numenta (such as Grok. The Numenta Platform for Intelligent Computer (NuPIC) is one of several available HTM implementations. Some are provided by Numenta, while some are developed and maintained by the HTM open source community

		The following commercial applications are available using NuPIC: • Grok – anomaly detection for IT servers, see www.grokstream.com • Cortical.io – advanced natural language processing, see www.cortical.io The following tools are available on NuPIC: • HTM Studio – find anomalies in time series using your own data, see www.numenta.com/htm-studio/ • Numenta Anomaly Benchmark – compare HTM anomalies with other anomaly detection techniques, see https://numenta.com/numenta-anomaly-benchmark/ The following example applications are available on NuPIC, see http://numenta.com/applications/: • HTM for stocks – example of tracking anomalies in the stock market • Rogue behavior detection – example of finding anomalies in human behavior • Geospatial tracking – example of finding anomalies in objectives moving through space and time
41	**Holographic Associative Memory (HAM)**	**Holographic Associative Memory** (HAM) Is a form of information storage where two pieces of information are saved and retrieved by associating them with one another in a pattern such that any part of the pattern contains them both and either piece can be used to retrieve the other. It has its roots in the principles of Holography. Holograms are made by using two beams of light, called a "reference beam" and an "object beam". They produce a pattern on the film that contains them both. Afterwards, by reproducing the reference beam, the hologram recreates a visual image of the original object Holographs have been shown to be effective for associative memory tasks, generalization, and pattern recognition with changeable attention. Used for searching image or digital patterns from a small sample of it, content-based search mechanism has been developed for querying into image database (IDB). Other uses are: • Small Target Recognition from Massive Digital Data. • Partial Pattern Matching. • Content-based Image Retrieval (CBIR). • Control Systems which need to work with Incomplete/Partial Sensory Inputs.
42	**Capsule NN**	The idea is to add structures called “capsules” to a convolutional neural network (CNN), and to reuse output from several of those capsules to form more stable (with respect to various perturbations) representations for higher order capsules. The output is a vector consisting of the probability of an observation, and a pose for that observation. This vector is similar to what is done for example when doing *classification with localization* in CNNs Capsnets address the "Picasso problem" in image recognition: images that have all the right parts but that are not in the correct spatial relationship (e.g., in a "face", the positions of the mouth and one eye are switched). In 2000, Geoffrey Hinton et. al. described an imaging system that combined segmentation and recognition into a single

		inference process using parse trees. So-called credibility networks described the joint distribution over the latent variables and over the possible parse trees. A capsule is a set of neurons that individually activate for various properties of a type of object, such as position, size and hue. Formally, a capsule is a set of neurons that collectively produce an *activity vector* with one element for each neuron to hold that neuron's instantiation value (e.g., hue).
43	**Hierarchical Convolutional Deep Maxout Network**	Multiscale hierarchical convolutional networks are structured deep convolutional networks where layers are indexed by progressively higher dimensional attributes, which are learned from training data. Each new layer is computed with multidimensional convolutions along spatial and attribute variables. Maxout activation function for CNN, which has shown to outperform the rectifier activation function in fully connected DNNs. The pooling operation of CNNs and the maxout function are closely related, and so the two technologies can be readily combined to build convolutional maxout networks For **phone recognition task**, a CNN built from maxout units yields a relative phone error rate reduction of about 4.3 % over ReLU CNNs. Applying the hierarchical modelling scheme to this CNN results in a further relative phone error rate reduction of 5.5 %. Using dropout training, the lowest error rate we get on TIMIT is 16.5 %, which is currently the best result
44	**Ensembles-DNN/CNN/RNN**	Current deep neural networks suffer from two problems; first, they are hard to interpret, and second, they suffer from overfitting. There have been many attempts to define interpretability in neural networks, but they typically lack causality or generality. A myriad of regularization techniques have been developed to prevent overfitting, and this has driven deep learning to become the hot topic it is today; however, while most regularization techniques are justified empirically and even intuitively, there is not much underlying theory A successful approach to reducing the variance of neural network models is to train multiple models instead of a single model and to combine the predictions from these models. This is called ensemble learning and not only reduces the variance of predictions but also can result in predictions that are better than any single model. Uses: • Classifying partial discharge (PD) patterns based on ensemble neural network (ENN) learning. • Applied in pooled flood frequency analysis for estimating the index flood and the 10-year flood quantile.
45	**Feed backward neural networks-General**	These are simplest class of NN and have simple input/ output nodes arranged in a forward chain. Applied to tasks like un-segmentation, and pattern recognition (connected handwriting

		recognition), examples are Kohonen's self organizing map and recurrent neural network (RNN). They are used in content addressable memories
46	**Wavelet networks**	Combining wavelets and neural networks results in wavelet networks. This is a feed forward network with one hidden layer. Forecasting the daily total amount of solar radiation

AI Algorithms – for ML, & DL

Algo No	Algorithm Name	Key Algorithm features, uses
1 **Stochastic Algorithms**	**Random Search**	Random search is a direct search method as it does not require derivatives to search a continuous domain. A combination of continuous and discrete variables arises in many complex systems including engineering design problems, scheduling and sequencing problems, and other applications in biological and economic systems. This base approach is related to techniques that provide small improvements such as Directed Random Search, and Adaptive Random Search. Random search algorithms are useful for many ill-structured global optimization problems with continuous and/or discrete variables. Typically random search algorithms sacrifice a guarantee of optimality for finding a good solution quickly with convergence results in probability. Because the methods typically only rely on function evaluations, rather than gradient and Hessian information, they can be coded quickly, and applied to a broad class of global optimization problems. A disadvantage of these methods is that they are currently customized to each specific problem largely through trial and error. The strategy of Random Search is to sample solutions from across the entire search space using a uniform probability distribution. Each future sample is independent of the samples that come before it. Uses : VLSI layout, Floor planning, Channel routing, Compaction, Vehicle routing, Traveling sales man, Scheduling , Process planning
2	**Adaptive Random Search**	The Adaptive Random Search algorithm was designed to address the limitations of the fixed step size in the Localized Random Search algorithm. The strategy for Adaptive Random Search is to continually approximate the optimal step size required to reach the global optimum in the search space. This is achieved by trialing and adopting smaller or larger step sizes only if they result in an improvement in the search performance.
3	**Stochastic Hill Climbing**	The strategy of the Stochastic Hill Climbing algorithm is iterate the process of randomly selecting a neighbor for a candidate solution and only accept it if it results in an improvement. The strategy was proposed to address the limitations of deterministic hill climbing techniques that were likely to get stuck in local optima due to their greedy acceptance of neighboring moves.

4	**Iterated Local Search**	The objective of Iterated Local Search is to improve upon stochastic Multi-Restart Search by sampling in the broader neighborhood of candidate solutions and using a Local Search technique to refine solutions to their local optima. Iterated Local Search explores a sequence of solutions created as perturbations of the current best solution, the result of which is refined using an embedded heuristic.
5	**Guided Local Search**	The strategy for the Guided Local Search algorithm is to use penalties to encourage a Local Search technique to escape local optima and discover the global optima. A Local Search algorithm is run until it gets stuck in a local optima. The features from the local optima are evaluated and penalized, the results of which are used in an augmented cost function employed by the Local Search procedure. The Local Search is repeated a number of times using the last local optima discovered and the augmented cost function that guides exploration away from solutions with features present in discovered local optima.
6	**Variable Neighborhood Search**	The strategy for the Variable Neighborhood Search involves iterative exploration of larger and larger neighborhoods for a given local optima until an improvement is located after which time the search across expanding neighborhoods is repeated. The strategy is motivated by three principles: 1) a local minimum for one neighborhood structure may not be a local minimum for a different neighborhood structure, 2) a global minimum is a local minimum for all possible neighborhood structures, and 3) local minima are relatively close to global minima for many problem classes.
7	**Greedy Randomized Adaptive Search**	The objective of the Greedy Randomized Adaptive Search Procedure is to repeatedly sample stochastically greedy solutions, and then use a local search procedure to refine them to a local optima. The strategy of the procedure is centered on the stochastic and greedy step-wise construction mechanism that constrains the selection and order-of-inclusion of the components of a solution based on the value they are expected to provide.
8	**Scatter Search**	The objective of Scatter Search is to maintain a set of diverse and high-quality candidate solutions. The principle of the approach is that useful information about the global optima is stored in a diverse and elite set of solutions (the reference set) and that recombining samples from the set can exploit this information. The strategy involves an iterative process, where a population of diverse and high-quality candidate solutions that are partitioned into subsets and linearly recombined to create weighted centroids of sample-based neighborhoods. The results of recombination are refined using an embedded heuristic and assessed in the context of the reference set as to whether or not they are retained.
9	**Tabu Search**	The objective for the Tabu Search algorithm is to constrain an embedded heuristic from returning to recently visited areas of the search space, referred to as cycling. The strategy of the approach is to maintain a short term memory of the specific changes of recent moves within the search space and preventing future moves from undoing those changes. Additional intermediate-term memory structures may be introduced to bias moves toward promising areas of the search space, as well as longer-term memory

		structures that promote a general diversity in the search across the search space.
10	**Reactive Tabu Search**	The objective of Tabu Search is to avoid cycles while applying a local search technique. The Reactive Tabu Search addresses this objective by explicitly monitoring the search and reacting to the occurrence of cycles and their repetition by adapting the tabu tenure (tabu list size). The strategy of the broader field of Reactive Search Optimization is to automate the process by which a practitioner configures a search procedure by monitoring its online behavior and to use machine learning techniques to adapt a techniques configuration.
11 Evolutionary Algorithms	**Genetic Algorithm**	The Genetic Algorithm is inspired by population genetics (including heredity and gene frequencies), and evolution at the population level, as well as the Mendelian understanding of the structure (such as chromosomes, genes, alleles) and mechanisms (such as recombination and mutation). The objective of the Genetic Algorithm is to maximize the payoff of candidate solutions in the population against a cost function from the problem domain. The strategy for the Genetic Algorithm is to repeatedly employ surrogates for the recombination and mutation genetic mechanisms on the population of candidate solutions, where the cost function (also known as objective or fitness function) applied to a decoded representation of a candidate governs the probabilistic contributions a given candidate solution can make to the subsequent generation of candidate solutions.
12	**Genetic Programming**	The objective of the Genetic Programming algorithm is to use induction to devise a computer program. This is achieved by using evolutionary operators on candidate programs with a tree structure to improve the adaptive fit between the population of candidate programs and an objective function. An assessment of a candidate solution involves its execution. The Genetic Programming algorithm was designed for inductive automatic programming and is well suited to symbolic regression, controller design, and machine learning tasks under the broader name of function approximation.
13	**Evolution Strategies**	Evolution Strategies is inspired by the theory of evolution by means of natural selection. Specifically, the technique is inspired by macro-level or the species-level process of evolution (phenotype, hereditary, variation) and is not concerned with the genetic mechanisms of evolution (genome, chromosomes, genes, alleles). The objective of the Evolution Strategies algorithm is to maximize the suitability of collection of candidate solutions in the context of an objective function from a domain. The objective was classically achieved through the adoption of dynamic variation, a surrogate for descent with modification, where the amount of variation was adapted dynamically with performance-based heuristics. Contemporary approaches co-adapt parameters that control the amount and bias of variation with the candidate solutions.
14	**Differential**	The Differential Evolution algorithm involves maintaining a population of candidate solutions subjected to iterations of recombination, evaluation, and selection. The recombination approach involves the creation of new candidate solution components

	Evolution	based on the weighted difference between two randomly selected population members added to a third population member. This perturbs population members relative to the spread of the broader population. In conjunction with selection, the perturbation effect self-organizes the sampling of the problem space, bounding it to known areas of interest.
15	**Evolutionary Programming**	A population of a species reproduces, creating progeny with small phenotypical variation. The progeny and the parents compete based on their suitability to the environment, where the generally more fit members constitute the subsequent generation and are provided with the opportunity to reproduce themselves. This process repeats, improving the adaptive fit between the species and the environment. The objective of the Evolutionary Programming algorithm is to maximize the suitability of a collection of candidate solutions in the context of an objective function from the domain. This objective is pursued by using an adaptive model with surrogates for the processes of evolution, specifically hereditary (reproduction with variation) under competition. The representation used for candidate solutions is directly assessable by a cost or objective function from the domain.
16	**Grammatical Evolution**	The Grammatical Evolution algorithm is inspired by the biological process used for generating a protein from genetic material as well as the broader genetic evolutionary process. The genome is comprised of DNA as a string of building blocks that are transcribed to RNA. RNA codons are in turn translated into sequences of amino acids and used in the protein. The resulting protein in its environment is the phenotype. The objective of Grammatical Evolution is to adapt an executable program to a problem specific objective function. This is achieved through an iterative process with surrogates of evolutionary mechanisms such as descent with variation, genetic mutation and recombination, and genetic transcription and gene expression. A population of programs are evolved in a sub-symbolic form as variable length binary strings and mapped to a symbolic and well-structured form as a context free grammar for execution.
17	**Gene Expression Programming**	Gene Expression Programming is inspired by the replication and expression of the DNA molecule, specifically at the gene level. The expression of a gene involves the transcription of its DNA to RNA which in turn forms amino acids that make up proteins in the phenotype of an organism. Gene Expression Programming uses a linear genome as the basis for genetic operators such as mutation, recombination, inversion, and transposition. The genome is comprised of chromosomes and each chromosome is comprised of genes that are translated into an expression tree to solve a given problem. The robust gene definition means that genetic operators can be applied to the sub-symbolic representation without concern for the structure of the resultant gene expression, providing separation of genotype and phenotype.
18	**Learning Classifier System**	Learning Classifier Systems make use of a Genetic Algorithm. The Learning Classifier System is a theoretical system with a number of implementations. The two main approaches to implementing and investigating the system empirically are the Pittsburgh-style that seeks to optimize the whole classifier, and the Michigan-style that optimize

		responsive rulesets. The Michigan-style Learning Classifier is the most common and is comprised of two versions: the ZCS (zeroth-level classifier system) and the XCS (accuracy-based classifier system). The processing loop for the Learning Classifier system is as follows: 1. Messages from the environment are placed on the message list. 2. The conditions of each classifier are checked to see if they are satisfied by at least one message in the message list. 3. All classifiers that are satisfied participate in a competition, those that win post their action to the message list. 4. All messages directed to the effectors are executed (causing actions in the environment). 5. All messages on the message list from the previous cycle are deleted (messages persist for a single cycle). The algorithm may be described in terms of the main processing loop and two sub-algorithms: a reinforcement learning algorithm such as the bucket brigade algorithm or Q-learning, and a genetic algorithm for optimization of the system.
19	**Non-dominated Sorting Genetic Algorithm**	The Non-dominated Sorting Genetic Algorithm is a Multiple Objective Optimization (MOO) algorithm and is an instance of an Evolutionary Algorithm The objective of the NSGA algorithm is to improve the adaptive fit of a population of candidate solutions to a Pareto front constrained by a set of objective functions. The algorithm uses an evolutionary process with surrogates for evolutionary operators including selection, genetic crossover, and genetic mutation. The population is sorted into a hierarchy of sub-populations based on the ordering of Pareto dominance. Similarity between members of each sub-group is evaluated on the Pareto front, and the resulting groups and similarity measures are used to promote a diverse front of non-dominated solutions.
20	**Strength Pareto Evolutionary Algorithm**	The objective of the algorithm is to locate and maintain a front of non-dominated solutions, ideally a set of Pareto optimal solutions. This is achieved by using an evolutionary process (with surrogate procedures for genetic recombination and mutation) to explore the search space, and a selection process that uses a combination of the degree to which a candidate solution is dominated (strength) and an estimation of density of the Pareto front as an assigned fitness. An archive of the non-dominated set is maintained separate from the population of candidate solutions used in the evolutionary process, providing a form of elitism.
21 **Physical Algorithms**	**Simulated Annealing**	Simulated Annealing is inspired by the process of annealing in metallurgy. In this natural process a material is heated and slowly cooled under controlled conditions to increase the size of the crystals in the material and reduce their defects. This has the effect of improving the strength and durability of the material. Each configuration of a solution in the search space represents a different internal energy of the system. Heating the system results in a relaxation of the acceptance

		criteria of the samples taken from the search space. As the system is cooled, the acceptance criteria of samples are narrowed to focus on improving movements. Once the system has cooled, the configuration will represent a sample at or close to a global optimum.
22	**Extremal Optimization**	Extremal Optimization is a stochastic search technique that has the properties of being a local and global search method The objective of the information processing strategy is to iteratively identify the worst performing components of a given solution and replace or swap them with other components. This is achieved through the allocation of cost to the components of the solution based on their contribution to the overall cost of the solution in the problem domain. Once components are assessed they can be ranked and the weaker components replaced or switched with a randomly selected component.
23	**Harmony Search**	Each musician corresponds to an attribute in a candidate solution from a problem domain, and each instrument's pitch and range corresponds to the bounds and constraints on the decision variable. The harmony between the musicians is taken as a complete candidate solution at a given time, and the audiences aesthetic appreciation of the harmony represent the problem specific cost function. The musicians seek harmony over time through small variations and improvisations, which results in an improvement against the cost function
24	**Cultural Algorithm**	The Cultural Algorithm is inspired by the principle of cultural evolution. Culture includes the habits, knowledge, beliefs, customs, and morals of a member of society. Culture does not exist independent of the environment, and can interact with the environment via positive or negative feedback cycles. The study of the interaction of culture in the environment is referred to as Cultural Ecology. The algorithm operates at two levels: a population level and a cultural level. The population level is like an evolutionary search, where individuals represent candidate solutions, are mostly distinct and their characteristics are translated into an objective or cost function in the problem domain. The second level is the knowledge or believe space where information acquired by generations is stored, and which is accessible to the current generation. A communication protocol is used to allow the two spaces to interact and the types of information that can be exchanged.
25	**Memetic Algorithm**	Memetic Algorithms are inspired by the interplay of genetic evolution and memetic evolution. Universal Darwinism is the generalization of genes beyond biological-based systems to any system where discrete units of information can be inherited and be subjected to evolutionary forces of selection and variation. The term 'meme' is used to refer to a piece of discrete cultural information, suggesting at the interplay of genetic and cultural evolution. The objective of the information processing strategy is to exploit a population based global search technique to broadly locate good areas of the search space, combined with the repeated usage of a local search heuristic by individual solutions to locate local

		optimum. Ideally, memetic algorithms embrace the duality of genetic and cultural evolution, allowing the transmission, selection, inheritance, and variation of memes as well as genes.
26 Probabilistic Algorithms	**Population-Based Incremental Learning**	Population-Based Incremental Learning is a population-based technique without an inspiration The information processing objective of the PBIL algorithm is to reduce the memory required by the genetic algorithm. This is done by reducing the population of a candidate solutions to a single prototype vector of attributes from which candidate solutions can be generated and assessed. Updates and mutation operators are also performed to the prototype vector, rather than the generated candidate solutions.
27	**Univariate Marginal Distribution Algorithm**	The information processing strategy of the algorithm is to use the frequency of the components in a population of candidate solutions in the construction of new candidate solutions. This is achieved by first measuring the frequency of each component in the population (the univariate marginal probability) and using the probabilities to influence the probabilistic selection of components in the component-wise construction of new candidate solutions. UMDA was designed for problems where the components of a solution are independent (linearly separable).
28	**Compact Genetic Algorithm**	The information processing objective of the algorithm is to simulate the behavior of a Genetic Algorithm with a much smaller memory footprint (without requiring a population to be maintained). This is achieved by maintaining a vector that specifies the probability of including each component in a solution in new candidate solutions. Candidate solutions are probabilistically generated from the vector and the components in the better solution are used to make small changes to the probabilities in the vector.
29	**Bayesian Optimization Algorithm**	The information processing objective of the technique is to construct a probabilistic model that describes the relationships between the components of fit solutions in the problem space. This is achieved by repeating the process of creating and sampling from a Bayesian network that contains the conditional dependencies, independencies, and conditional probabilities between the components of a solution. The network is constructed from the relative frequencies of the components within a population of high fitness candidate solutions. Once the network is constructed, the candidate solutions are discarded and a new population of candidate solutions are generated from the model. The process is repeated until the model converges on a fit prototype solution.
30	**Cross-Entropy Method**	The information processing strategy of the algorithm is to sample the problem space and approximate the distribution of good solutions. This is achieved by assuming a distribution of the problem space (such as Gaussian), sampling the problem domain by generating candidate solutions using the distribution, and updating the distribution based on the better candidate solutions discovered. Samples are constructed step-wise (one component at a time) based on the summarized distribution of good solutions. As the algorithm progresses, the distribution becomes more refined until it focuses on the area or scope of optimal solutions in the domain.

31 Swarm Algorithms	**Particle Swarm Optimization**	Particle Swarm Optimization is inspired by the social foraging behavior of some animals such as flocking behavior of birds and the schooling behavior of fish. The goal of the algorithm is to have all the particles locate the optima in a multi-dimensional hyper-volume. This is achieved by assigning initially random positions to all particles in the space and small initial random velocities. The algorithm is executed like a simulation, advancing the position of each particle in turn based on its velocity, the best known global position in the problem space and the best position known to a particle. The objective function is sampled after each position update. Over time, through a combination of exploration and exploitation of known good positions in the search space, the particles cluster or converge together around an optima, or several optima.
32	**Ant System**	The Ant system algorithm is inspired by the foraging behavior of ants, specifically the pheromone communication between ants regarding a good path between the colony and a food source in an environment. This mechanism is called stigmergy. Ants initially wander randomly around their environment. Once food is located an ant will begin laying down pheromone in the environment. Numerous trips between the food and the colony are performed and if the same route is followed that leads to food then additional pheromone is laid down. Pheromone decays in the environment, so that older paths are less likely to be followed. Other ants may discover the same path to the food and in turn may follow it and also lay down pheromone. A positive feedback process routes more and more ants to productive paths that are in turn further refined through use. The Ant Systems algorithm was designed for use with combinatorial problems such as the TSP, knapsack problem, quadratic assignment problems, graph coloring problems and many others.
33	**Ant Colony System**	The Ant Colony System algorithm is inspired by the foraging behavior of ants, specifically the pheromone communication between ants regarding a good path between the colony and a food source in an environment. This mechanism is called stigmergy. Algorithm provides the main Ant Colony System algorithm for minimizing a cost function. The probabilistic step-wise construction of solution makes use of both history (pheromone) and problem-specific heuristic information to incrementally construct a solution piece-by-piece. Each component can only be selected if it has not already been chosen (for most combinatorial problems),
34	**Bees Algorithm**	The Bees Algorithm is inspired by the foraging behavior of honey bees. Honey bees collect nectar from vast areas around their hive (more than 10 kilometers). Bee Colonies have been observed to send bees to collect nectar from flower patches relative to the amount of food available at each patch. Bees communicate with each other at the hive via a waggle dance that informs other bees in the hive as to the direction, distance, and quality rating of food sources. The information processing objective of the algorithm is to locate and explore good sites within a problem search space. Scouts are sent out to randomly sample the problem space and locate good sites. The good sites are exploited via the application of a local search, where a small number of good sites are explored more than the others. Good sites are continually exploited, although many scouts are sent out each iteration always in search of additional good sites .

35	**Bacterial Foraging Optimization Algorithm**	The Bacterial Foraging Optimization Algorithm is inspired by the group foraging behavior of bacteria such as E.coli and M.xanthus. Specifically, the BFOA is inspired by the chemotaxis behavior of bacteria that will perceive chemical gradients in the environment (such as nutrients) and move toward or away from specific signals. The information processing strategy of the algorithm is to allow cells to stochastically and collectively swarm toward optima. This is achieved through a series of three processes on a population of simulated cells: 1) 'Chemotaxis' where the cost of cells is derated by the proximity to other cells and cells move along the manipulated cost surface one at a time (the majority of the work of the algorithm), 2) 'Reproduction' where only those cells that performed well over their lifetime may contribute to the next generation, and 3) 'Elimination-dispersal' where cells are discarded and new random samples are inserted with a low probability.
36 Immune Algorithms	**Clonal Selection Algorithm**	The Clonal Selection algorithm is inspired by the Clonal Selection theory of acquired immunity. The theory proposes that antigens select-for lymphocytes (both B and T-cells). When a lymphocyte is selected and binds to an antigenic determinant, the cell proliferates making many thousands more copies of it and differentiates into different cell types (plasma and memory cells). Plasma cells have a short lifespan and produce vast quantities of antibody molecules, whereas memory cells live for an extended period in the host anticipating future recognition of the same determinant. The important feature of the theory is that when a cell is selected and proliferates, it is subjected to small copying errors (changes to the genome called somatic hypermutation) that change the shape of the expressed receptors and subsequent determinant recognition capabilities of both the antibodies bound to the lymphocytes cells surface, and the antibodies that plasma cells produce. The information processing principles of the clonal selection theory describe a general learning strategy. This strategy involves a population of adaptive information units (each representing a problem-solution or component) subjected to a competitive processes for selection, which together with the resultant duplication and variation ultimately improves the adaptive fit of the information units to their environment.
37	**Negative Selection Algorithm**	The Negative Selection algorithm is inspired by the self-nonself discrimination behavior observed in the mammalian acquired immune system. An interesting aspect of this process is that it is responsible for managing a population of immune cells that do not select-for the tissues of the body, specifically it does not create self-reactive immune cells known as auto-immunity. This problem is known as 'self-nonself discrimination' and it involves the preparation and ongoing maintenance of a repertoire of immune cells such that none are auto-immune. This is achieved by a negative selection process that selects-for and removes those cells that are self-reactive during cell creation and cell proliferation. The information processing principles of the self-nonself discrimination process via negative selection are that of a anomaly and change detection systems that model the anticipation of variation from what is known. The principle is achieved by building a model of changes, anomalies, or unknown (non-normal or non-self) data by generating patterns that do not match an existing corpus of available (self or normal) patterns. The prepared non-normal model is then used to either monitor the existing normal data or

		streams of new data by seeking matches to the non-normal patterns.
38	**Artificial Immune Recognition System**	The theory suggests that starting with an initial repertoire of general immune cells, the system is able to change itself (the compositions and densities of cells and their receptors) in response to experience with the environment. Through a blind process of selection and accumulated variation on the large scale of many billions of cells, the acquired immune system is capable of acquiring the necessary information to protect the host organism from the specific pathogenic dangers of the environment. The information processing objective of the technique is to prepare a set of real-valued vectors to classify patterns. The Artificial Immune Recognition System maintains a pool of memory cells that are prepared by exposing the system to a single iteration of the training data. Candidate memory cells are prepared when the memory cells are insufficiently stimulated for a given input pattern. A process of cloning and mutation of cells occurs for the most stimulated memory cell. The clones compete with each other for entry into the memory pool based on stimulation and on the amount of resources each cell is using. This concept of resources comes from prior work on Artificial Immune Networks, where a single cell (an Artificial Recognition Ball or ARB) represents a set of similar cells. Here, a cell's resources are a function of its stimulation to a given input pattern and the number of clones it may create.
39	**Immune Network Algorithm**	The Artificial Immune Network algorithm is inspired by the Immune Network theory of the acquired immune system. The clonal selection theory of acquired immunity accounts for the adaptive behavior of the immune system including the ongoing selection and proliferation of cells that select-for potentially harmful (and typically foreign) material in the body. The objective of the immune network process is to prepare a repertoire of discrete pattern detectors for a given problem domain, where better performing cells suppress low-affinity (similar) cells in the network. This principle is achieved through an interactive process of exposing the population to external information to which it responds with both a clonal selection response and internal meta-dynamics of intra-population responses that stabilizes the responses of the population to the external stimuli.
40	**Dendritic Cell Algorithm**	The Dendritic Cell Algorithm is inspired by the Danger Theory of the mammalian immune system, and specifically the role and function of dendritic cells. The information processing objective of the algorithm is to prepare a set of mature dendritic cells (prototypes) that provide context specific information about how to classify normal and anomalous input patterns. This is achieved as a system of three asynchronous processes of 1) migrating sufficiently stimulated immature cells, 2) promoting migrated cells to semi-mature (safe) or mature (danger) status depending on their accumulated response, and 3) labeling observed patterns as safe or dangerous based on the composition of the sub-population of cells that respond to each pattern.
41 Neural Algorith	**Perceptron**	The Perceptron is inspired by the information processing of a single neural cell (called a neuron). A neuron accepts input signals via its dendrites, which pass the electrical signal down to the cell body. The axon carry the signal out to synapses, which are the

ms		connections of a cell's axon to other cell's dendrites The information processing objective of the technique is to model a given function by modifying internal weightings of input signals to produce an expected output signal. The system is trained using a supervised learning method, where the error between the system's output and a known expected output is presented to the system and used to modify its internal state. State is maintained in a set of weightings on the input signals. The weights are used to represent an abstraction of the mapping of input vectors to the output signal for the examples that the system was exposed to during training.
42	**Back-propagation**	The information processing objective of the technique is to model a given function by modifying internal weightings of input signals to produce an expected output signal. The system is trained using a supervised learning method, where the error between the system's output and a known expected output is presented to the system and used to modify its internal state. State is maintained in a set of weightings on the input signals. The weights are used to represent an abstraction of the mapping of input vectors to the output signal for the examples that the system was exposed to during training. Each layer of the network provides an abstraction of the information processing of the previous layer, allowing the combination of sub-functions and higher order modeling
43	**Hopfield Network**	The Hopfield Network algorithm is inspired by the associated memory properties of the human brain Through the training process, the weights in the network may be thought to minimize an energy function and slide down an energy surface. In a trained network, each pattern presented to the network provides an attractor, where progress is made towards the point of attraction by propagating information around the network.
44	**Learning Vector Quantization**	The Learning Vector Quantization algorithm is related to the Self-Organizing Map which is in turn inspired by the self-organizing capabilities of neurons in the visual cortex The information processing objective of the algorithm is to prepare a set of codebook (or prototype) vectors in the domain of the observed input data samples and to use these vectors to classify unseen examples. An initially random pool of vectors is prepared which are then exposed to training samples. A winner-take-all strategy is employed where one or more of the most similar vectors to a given input pattern are selected and adjusted to be closer to the input vector, and in some cases, further away from the winner for runners up
45	**Self-Organizing Map**	The Self-Organizing Map is an unsupervised neural network that uses a competitive (winner-take-all) learning strategy. The Self-Organizing Map is inspired by postulated feature maps of neurons in the brain comprised of feature-sensitive cells that provide ordered projections between neuronal layers, such as those that may exist in the retina and cochlea. For example, there are acoustic feature maps that respond to sounds to which an animal is most frequently exposed, and tonotopic maps that may be responsible for the order preservation of acoustic resonances. The information processing objective of the algorithm is to optimally place a topology

		(grid or lattice) of codebook or prototype vectors in the domain of the observed input data samples. An initially random pool of vectors is prepared which are then exposed to training samples. A winner-take-all strategy is employed where the most similar vector to a given input pattern is selected, then the selected vector and neighbors of the selected vector are updated to closer resemble the input pattern. The repetition of this process results in the distribution of codebook vectors in the input space which approximate the underlying distribution of samples from the test dataset. The result is the mapping of the topology of codebook vectors to the underlying structure in the input samples which may be summarized or visualized to reveal topologically preserved features from the input space in a low-dimensional projection
46	**Truncated BPTT - algorithm**	Truncated backpropagation through time (BPTT) was developed in order to reduce the computational complexity of each parameter update in a recurrent neural network. In summary, it allows us to train networks faster (by performing more frequent parameter updates), for a given amount of computational power. It is recommended to use truncated BPTT when your input sequences are long (typically, more than a few hundred time steps).
47	**k-nearest neighbours (kNNs) - algorithm**	k-NN can be used for both classification and regression predictive problems. However, it is more widely used in classification problems in the industry. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection. It is widely disposable in real-life scenarios since it is non-parametric, meaning; it does not make any underlying assumptions about the distribution of data (as opposed to other algorithms such as GMM, which assume a Gaussian distribution of the given data).

Disclaimer:

Several details have been sourced from the web and we have no intention to infringe any names, copyrights and data that might be presented in this document. This is for information and educational purpose only and no monetary claims can be made. Please advise us for anything that you feel to let us know.

Few of the references are:

Jason Brownlee, PhD, June (2012). Clever Algorithms: Nature-Inspired Programming Recipes by, Revision 2. Retrieved from http://www.cleveralgorithms.com/nature-inspired/index.html

Van Veen, F. & Leijnen, S. (2019). The Neural Network Zoo. Retrieved from http://www.asimovinstitute.org/neural-network-zoo

Details for few of the topics were referred at Wikipedia, Towards Data Science, KDNuggets, Medium, Arxiv, and Data Flair training, to arrive at our own content

For more details, contact below:

Contact

Bhadale IT Developers Pvt. Ltd
CTO: Vijay Mohire, 9741040195; Email: vijaymohire@gmail.com