Reproduction: process of producing offspring from selected parents by applying crossover and/or mutation operators **Crossover** is the process of creating one or more new individuals through the combination of genetic material randomly selected from two or more parents**Mutation**:The process of randomly changing the values of genes in a chromosome The main objective is to introduce new genetic material into the population, thereby increasing genetic diversity Mutation probability and step sizes should be small; **high rate**: risk no converge/**low rate**:stuck local optima **Genotype:/Chromosome**: a set of parameters which define a proposed solution to the problem the GA try to solve genetic composition of an individual, as inherited from its parents/ DNA molecule with part or all of the genetic material of an organism. **Phenotype:** behavioral traits of an individual in a specific environment/ **Simulated annealing** (**SA**) is a probabilistic technique for approximating the global optimum of a given function, often used when the search space is discrete Elitism vs Generalition Ensures that the best individuals of the current population survive to the next generation The best individuals are copied to the new population without being mutated The more individuals that survive to the next generation, the less the diversity of the new population **Evolutionary Computation**: (natural selection)computer-based problem solving systems that use computational models of evolutionary processes When to Stop Limit the number of generations, or fitness evaluations / Stop when population has converged: Terminate when no improvement is observed over a number of consecutive generations Terminate when there is no change in the population Terminate when an acceptable solution has been found Terminate when the objective function slope is approximately zero **Generic Algorithm** (scheduling/TSP)**Population** (Chromosome) initialisation, **fitness** assignment(evaluate population quality), <EnD:classifier performance,> Then: **selection**(pick Better solution by Proportional/Tournament selection scheme(care order)), **crossover**(combine good solution by 1/2 point/uniform/ordered crossover), **mutation**(explore new solution by guardian mutation/ Shuffle indices/Flip bit), get new fitness value, proceed chromosomes to next generation Feature Slection: GA; Correlation Method for Feature Selection (Pseudo-independent Feature Selection algorithm (**PISA**)) measure for strength of linear relationship between feature. Both used for SVM input Effects of Control Parameters : Dynamic mutation rates, more diversity, more computation time & search space, Large initial mutation rate/mutation rate decreases favours exploration, Large mutational step size – large jumps in search space, better exploration,Initially large step sizes, with step sizes reduced over time,step size be proportional to fitness of individual,worse an individual, the larger the mutational step size Generational genetic algortihms (GGA) : replaces all parents with offspring after create & mutate all offsp,No overlap btw populations of different generations Steady state genetic algorithms (SSGA) : Immediately(Replace worst/random/oldest/Kill tournament of the current population), Some overlap exists btw populations of different generations.FIFO:Conservative selection/Replace oldest . Interactive Evolution: involves a human user online into the selection and variation processes; Requires that the phenotype of individuals be generated from the genotype Phenotype is then visualized Based on the visual representations of candidate solutions, the user selects those individuals that will take part in reproduction User also selects individuals that will survive to the next generation**/// Generic Programming**: A specialization of GAs, Each **chromosome** respresents a solution as a **tree. Initial Population:** Randomly generated within the restrictions of a maximum depth and semantics as expressed by the given grammar For each individual, a root is randomly selected from the set of function elements Non-terminal nodes are randomly selected from the function set Terminal nodes selected from the terminal set Start with small initial trees. **Fitness Function:** Fitness calculation requires the program to be evaluated against a number of test cases(Boolean/MSE) may include a penalty to penalize individuals with undesirable structural properties: invalid solutions/tree size ->CrossOver->Mutation…Building-Block Approach to GP: Expansion occurs by adding a randomly generated building block (i.e. a new node) to individuals, reduce the computational complexity of the evolution process, and to produce smaller individuals(✓) **Evolutionary Algorithms:** (Art,like Simulated Annealing is randomised search) are programs which solve 'problems' by simulating Darwinian selection among solutions Individuals are assessed for fitness and create 'offspring'Over time, accumulate good components: better solutions Randomness in creating images = minor creativity (or at least an illusion of it) better gene is darker Why hibridize? **Evolutionary Algorithms** – Explore large, rough search spaces – Difficulties with fine-tuning **Local search techniques –** Optimise/converge fast stuck in local optima There are costs of the learning, learning not always good (✕) When to Use? problem-specific algorithm known but no time/ expertise, Advantages/Disa of EA: easy to apply/use/test, run in acceptable time, no success no harm(✓) no proven upper bound for run time, optimal solution not guaranteed (✕) **Swarming:** aggregation of similar animals, generally cruising in the same direction,To migrate/forage better/defense against predators Collision Avoidance: Avoid Collision with neighboring boids/Match the velocity of neighboring boids/Stay near neighboring boids Particle Swarm Optimization: Bird flocking. individuals interact with one another while learning from their own experience, and gradually move towards the goal. PSO: Each particle (or agent) evaluates the function to maximize at each point it visits in spaces.Each agent remembers the best value of the function found so far by it (pbest) and its co- ordinates. Secondly, each agent know the globally best position that one member of the flock had found, and its value (gbest). 1D: best cord; 2D:cognitive+social component. **Ant Colony Optimization:** Inspired by foraging behavior of ants. Ants find shortest path to food source from nest. Ant Colony Metaheuristic: **Construct Ant Solutions**: Partial solution extended by adding an edge based on stochastic and pheromone considerations. **Apply Local Search**: problem-specific, used ACO algorithms. **Update Pheromones:** increase pheromone of good solutions, decrease that of bad solutions (pheromone evaporation) MAX-MIN Ant System (MMAS) Vs. Ant System (AS): Convergence to optimal solutions has been proved. Cannot predict how quickly optimal results will be found. Suffer from stagnation and selection bias. **SOM algorithm**:Initialize input nodes, output nodes, and connection weights(input vector: top (most frequently occurring) N terms;output nodes: create a 2-dimensional map (grid) of M/ Initialize weights wij from N input nodes to M output nodes to small random values) Present each document in order, Compute distance to all nodes,Select winning node i* and update weights to node i* and its neighbors(reduce the distances between them and the input vector)SOM Purpose: Map from higher number of dimensions down/Unsupervised learning/Want similar input vectors represented by nearby neurons A geometric analogy: Neighbourhood decreases over time: Some similarity to simulated annealing. **Bacterial memetic algorithm** (Fast converge)Generating the initial population randomly Bacterial mutation is applied for each bacterium Levenberg-Marquardt method is applied for each bacterium Gene transfer is applied in the population If a stopping condition is fullfilled then the algorithm stops, otherwise it continues with the bacterial mutation step. Conclude: Increasing the number of generations and the number of individuals also increases the performance of the algorithm (however, it causes additional computational effort) Gene transfer: population is divided into two halves One bacterium is randomly chosen from the superior half superior (source bacterium) and half another from the inferior half (destination bacterium) A part from the source bacterium is chosen and this part can overwrite a part of the destination bacterium. <but also the length of the destination bacterium> GA Vs. BEA: GA based on the evolution process of mammals, while BEA based on the evolution process of bacteria GA uses crossover, BEA uses gene transfer for the information flow in the population Bacterial mutation is more effective than mutation in GA (The gene transfer operator can be realized easier than the crossover operator in genetic algorithms) There is no selection in BEA, but there is multiplication by fission (cloning method) Memetic algorithms: local search operators(avoid local mini)+ EA algo, better results EC vs CO The search process: CO uses deterministic rules to move from one point in the search space to the next point EC uses probabilistic transition rules EC applies a parallel search of the search space, while CO uses a sequential search /Search surface information: CO uses derivative information, usually first-order or second-order, of the search space to guide the path to the optimum EC uses no derivative information, but fitness information

Fuzzy Inference: use **monotonic selection**(The value of a truth membership grade of the rule consequent can be estimated directly from a corresponding truth membership grade in the antecedent) Mamdani Fuzzy Inference 4 steps: **Fuzzification** of the input variables/**Rule evaluation** (inference: antecedent ->consequent<clipping/scaling>)/**Aggregation** of the rule outputs (composition)/**Defuzzification** Fuzzification: definition of fuzzy sets, and determination of the degree of membership of crisp inputs in appropriate fuzzy sets. Inference: evaluation of fuzzy rules to produce an output for each rule. Composition: aggregation or combination of the outputs of all rules. Defuzzification: computation of crisp output(Centroid gravity of fuzzy set/ mean value of maximum/smallest(largest) (absolute) value of maximum) Clipping(alpha-cut): The most common method of correlating the rule consequent with the truth value of the rule antecedent is to cut the consequent membership function at the level of the antecedent truth, Since top membership func is sliced, the *clipped fuzzy set* loses some information (✕) .it involves less complex and faster mathematics, and generates an aggregated output surface that is easier to defuzzify (✓) Scaling: The original membership function of the rule consequent is adjusted by multiplying all its membership degrees by the truth value of the rule antecedent. Better approach for preserving the original shape of the fuzzy set, less info lose (✓). Singleton: a fuzzy set with a membership function that is unity at a single particular point on the universe of discourse and zero everywhere else. Hedges: A linguistic variable carries with it the concept of fuzzy set qualifiers, are terms that modify the shape of fuzzy sets. They include adverbs such as very/ more/lessFuzzy intersection: the lower membership in both sets of each element. Fuzzy rule interpolation: provide conclusions where no overlap with even the supports of existing rules in the rule base, low computat. (✓) / lead to distorted/abnormal fuzzy rules (✕) **Fuzzy** representations: sigmoid, Gaussian (high computation time) Triangular and Trapezoidal(derivatives are not defined for all points) IF wind is strong THEN sailing is good. Properties:

Equality of two fuzzy sets/Inclusion of one set into another fuzzy set /**Cardinality** of a fuzzy set<Cardinality of a non-fuzzy set, Z, is the number of elements in Z/Cardinality of a fuzzy set A is the sum of the values of the membership function of A>/An empty fuzzy set/ α**-cuts** < a fuzzy set A⊆X is an ORDINARY SET $A_\alpha \subseteq X$>Fuzzy Set Normality: A fuzzy subset of X is called normal if there exists at least one element x∈X such that μA(x) = 1.A fuzzy subset that is not normal is called subnormal.The height of a fuzzy subset A is the large membership grade of an element in A. Hard Clustering: **K-mean** <edge points are not accurate> Soft clustering: Fuzzy cluster(**C-mean**:Max/Min similarity within/between clusters, Objective function(measure dissimilarity within cluster): min obj func to optimise), generalise[0,1],bad at computation, noise Classical Vs. Fuzzy Rule: IF speed > 100km/h IF speed < 40km/h THEN stopping_distance is >88m **Vs.** IF speed is fast THEN stopping_distance is long Mamdani Vs. Sugeno Inference: Mamdani needs the centroid of a 2D shape by integrating across a continuously varying function<slow computation> Sugeno uses a single spike, a singleton, as the membership function of the rule consequent, good at time, dynamic nonlinear systems.<Diff>. changed only a rule consequent.IF x is A AND y is B THEN z is f(x, y) zero-order Sugeno fuzzy model: IF x is A AND y is B THEN z is k <Same> Fuzzy Hierarchical Problem: |R| = O(Tk) **Decrease T**: allow sparse fuzzy rule bases use nearby rules - fuzzy interpolation **Decrease k**: hierarchical fuzzy rule bases, full cover in bordering domains (complexity no change)**Decrease T and k**: hierarchical sparse fuzzy rule bases:interpolate between different branches of hierarchical rule tree, can omit bordering domains and just interpolate from nearby rules. Fuzzy Signature: a vector of fuzzy values, where each vector component can be another vector, Sparse and hierarchical,Simplifies the approximation of aggregation(min/max), robust and flexible under perturbed input data Compare Fuzzy signatures: Truncate to find a common structure/Use operations conjunction or disjunction/Aggregate resulting fuzzy signature Polymorphic Fuzzy Signatures: Find a common signature structure for set of data points, Use fuzzy constrains/events at leaves/Use specialised aggregations operators (such as WRAO) to fuse data (Fuzzy C-Means clustering to identify better subspaces for PFS/Fuzzy Signatures for single document analysis)
Feature federalisation DL vs. tradition: Deep learning generates its own features through representational learning from the raw data. Traditional techniques were fed hand-engineered features that were prepared ahead of time. How local receptive field diff: Two neurons in the same filter have DIFFERENT local receptive fields. They look at different slices of the input data. pooling layer: To condense the feature maps produced by a convolutional layer. This works to reduce the number of parameters in the network, therefore reducing size and computational cost. Kernel: apply the weights and biases of a filter,As many filters (features) as there are,Once for every neuron in the filter, output 1 each time, stored In a feature or activation map DropOut: reduce the overfitting of the model to the training data. **RNN:** nh + h2 + h, vanishing or exploding gradient during backpropagation, long term dependencies. **LSTM:** 4(nh + h2 + h) + 3h2, a 'forget' gate which controls what should be removed; an 'input gate' which determines what should be added; a 'write' gate which determines what data should be updated. Process: Loading data, Process input through the network and get output, Compute the loss (output, target pair) Cross Entropy loss, Propagate the gradient back (gradient = 0), Update the weights of the network (Stochastic Gradient Descent),Testing our trained network. Q learning: a type of value iteration method which aims to approximate the Q function (the long term reward from taking action a in state s under policy π), while Policy Gradients is a method to directly optimize in the action space Q Learning (DQN) Vs. Policy Gradients: DQN estimates the value of taking a particular action in a particular state, whereas Policy Gradients is taking a softmax over actions to predict the probability that a particular action in a particular state is the best action. Policy-Based RL: Better convergence properties. Effective in high-dimensional or continuous action spaces, Can learn stochastic policies. **GAN**: Discriminative Model D: Learns to classify the sample - CNN; Generative Model G: Deconvolutional/Conv
Transpose NN,As similar as possible to the real data Variational Autoencoder (VAE): an encoder that encodes the data into latent variables z and a decoder that reconstructs the data given z**GAN Stability**: Track failures early (D loss goes to 0: failure mode), max log D avoid vanishing gradients, Normalize the images,Sample from a Gaussian distribution,Construct different mini-batches for real and fake, Avoid Sparse Gradients: ReLU, MaxPool (For Downsampling, use: Average Pooling, Conv2d + stride/For Upsampling, use: ConvTranspose2d + stride)Use Soft and Noisy(discriminator) Labels WGANs have loss functions that correlate with image quality **Transfer Learning:** reduce both the amount of labelled data and time needed for training a Deep NN, Limited labelled data but large amounts of unlabelled data from the same or similar dataset, Learning about the input distribution can help with learning a mapping from input to output, The choice of initial parameters for a DNN can have a significant regularising effect. Helps to reduce overfitting Restricting autoencoders: Limit the dimension,Regularisation **Semi-supervised learning** is used in the case where we have large amounts of unlabelled data and limited labelled data all from the same domain /// **Internal Valid**: Validate your model on your current data set (cross-valida&on) **External Valid**: Validate your model on a completely new dataset/ 10-fold cross-valida: Randomly divide your data into 10 pieces, 1 through k.Treat the 1st tenth of the data as the test dataset. Fit the model to the other nine-tenths of the data,Apply the model to the test data and calculate error etc,Repeat this procedure for all 10 tenths of the data.Calculate satisfy of model accuracy and fit from the test data only. **Cascade Correlation:** an architecture generative,feedforward, supervised learning algorithm for ANN, add new hidden units one by one. Faster than backpropogation, dynamic, use correlation/weight freezing. CasPer: overcome generalisation of CasCor, use PROPO to train.
Advantages for **generalisation** (Z function): for shared weights and bidirectional training probably derive from Reduction in free parameters Faster training of input to hidden weights the weights receive stronger feedback than with standard backprop. training. **Bidirectional Net**: Allow mapping: outputs to inputs； functional symmetry； Network – weights same both dirs. Input neurons bias weights for use only in reverse dir. **Shared Weights**: link A —> B = B —> C (no. weights same) Reduce space of network weight configurations – require the **compression** function to be invertible. Weight decay: decrease weights during training – eliminates unnecessary connections where f is weight calculated normally for current model. **Softmax**: Can handle Multiple (e.g 1000 classes) classification, output is the classes distribution probabilities. **Non-linear** Activation function. **Rectified Linear Units (RELU)**: When the input is positive, the derivative is 1, don't lose data, Can be used in deep learning, Fragile during training and die **Sigmoid Function:** It's derivable, less calculation and building time, Lose data, (e.g lose the maximum value after derivate) Cannot used in deep learning as an activation function lose more data, get "big" error produced by output layer **Alternative for Back Propagation:** K-Nearest Neighbour**:** new vector is compared to all exemplars and closest is chosen. **Few data points** – use nearest neighbour. **Back Propagation** may require lots of training data for good generalisation**Noisy data** – use back-propagation as it is less sensitive to noise**Training time** – none for nearest neighbour**Classification speed** – back-propagation faster, and more efficient **Accuracy** – both nearest neighbour and back-propagation can be made arbitrarily accurate if enough data is available Decision Trees: faster than back- propagation, **generalisation** often not as good Radial Basis Function: this approach is based on sums of gaussian functions, The gaussians are centred on selected training points, Training time is faster than back-prop, but more neurons may be required, so RBFs are less efficient in practice. **Training step:** Initiate random weights, Input new patterns, present desired output,Go through activation function along weighted links, calculate actual output, adapt weights recursively, **starting** from **output** layer and working **backwards** towards the **input** layer, repeat from input. **Learning Rate:** The learning rate, which controls how much the weights are adjusted at each update. η changes as training proceeds for all training methods except RBFN, where eta remains constant. **Initial** η is the starting value of η. During training, η starts at Initial η, **decreases** to **Low** η, then is **reset** to **High** η and **decreases** to **Low** η again. The last two steps are **repeated** until training is complete. **Momentum**: used in updating the weights during training. **Momentum** tends to keep the weight changes moving in a consistent direction. Specify a value between 0 and 1. **Bad Hidden units:** weights very small, Opposite output for all patterns, same output for all patterns, sum of vectors is zero. Autoassociative Network: any type of memory that enables one to retrieve a piece of data from only a tiny sample of itself. It is often misunderstood to be only a form of back propagation or other neural networks, input to hidden: compression phase, hidden to output: decompression **Neural Network:** multiple processing neurons, each neuron has simple function, speed of process can be slow, complex interconnections between neutrons, **ANN Bio**: Artificial neural networks are inspired by the biological brain, and are designed to mimic how learning happens in the brain through a series of interconnected neurons. ANNs are engineered to duplicate the computational principles behind the brain and hence duplicate its functionality, as the brain is evidence that intelligent behaviour is possible through such a structure. Single Neuron: Input, Bias,, weight, activation Func, output. **Middle layer**: referred to as **hidden** as their values are not given in the data; instead the model must determine which concepts are useful for explaining the relationships in the observed data. **Back propagation:** Apply inputs to network and work out the output (randomise initial weights),Calculate errors of output neurons,Change output layer weights to adjust for errors,Calculate (back-propagate) errors from output layer to the hidden layer neurons. Take errors from output neurons and run them back through the weights to get the hidden layer errors,Change hidden layer weights,Repeat