2014 NASA Cost Analysis Symposium

Utilizing Artificial Neural Networks for CER Development

Betsy Turnbull

Tom Parkey

Jon Drexler

Glenn Research Center

August 14, 2014

Agenda

- Background
- Introduction to Artificial Neural Networks (ANNs)
- Training the Network
- Applying ANNs to CER Development
- Performance/Discussion of Results
- Summary
- Forward Work

Background

- Inspired by a paper in the Journal of Cost Analysis and Parametrics: Salam, Defersha, Bhuiyan and Chen, "*A Case Study on Target Cost Estimation Using Back-Propagation and Genetic Algorithm Trained Neural Networks*", December 2012
- Arrival of intern in the summer of 2013 provided resource to investigate applicability to NASA cost estimating

Introduction to Artificial Neural Networks (ANNs)

- ANNs are a mathematical construction designed to mimic the connections in the brain
  - Network is “trained” to return expected responses given known data, which can then make generalized predictions based on a new set of parameters
- Network consists of:
  - Layer of input nodes, which are the independent variables
  - One or more layers of hidden nodes
  - Layer of output nodes, which for our purpose is a single node (cost)
- Hidden nodes and output nodes are calculated from the values of ALL of the nodes in the preceding layer (*feed forward*) and the weights that connect them, typically modeled as a sigmoid function:

$$Node(H) = f\big(net(H)\big) = \frac{1}{1+\exp\big(-net(H)\big)}, \text{ where}$$

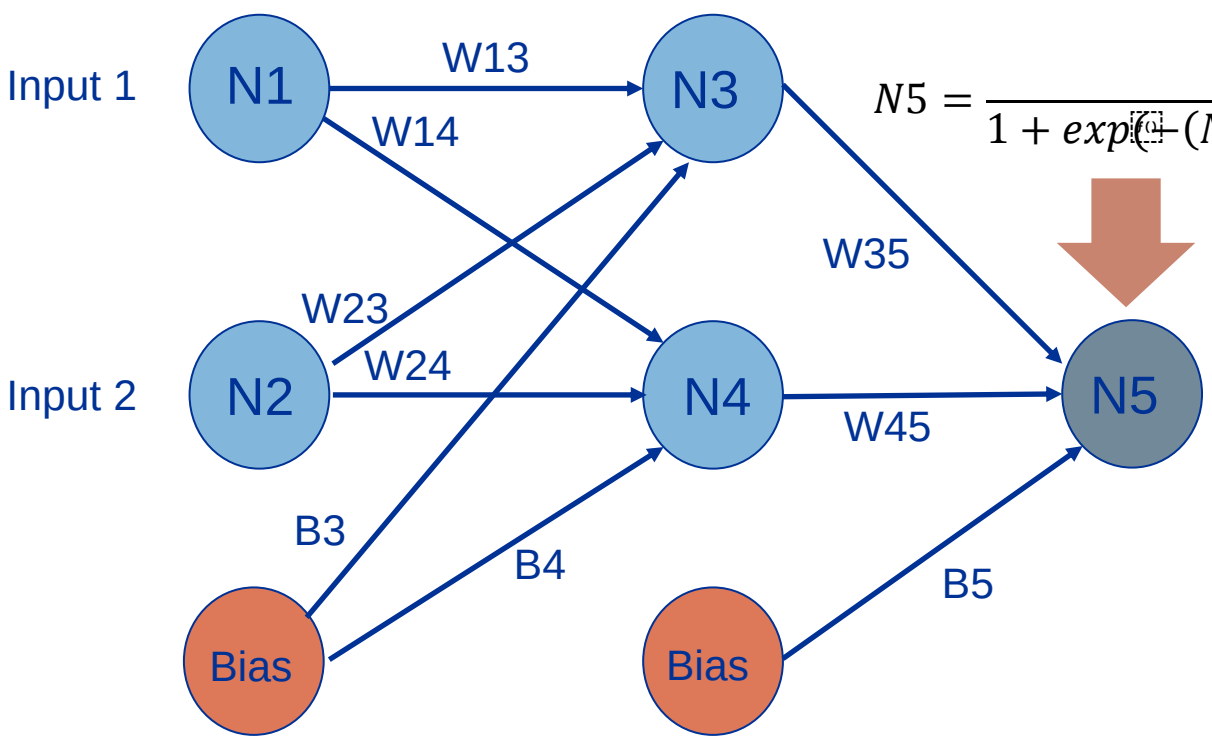
$$net(H) = \sum value\ of\ nodes\ in\ previous\ layer * connecting\ weight + bias(H)$$

- Input and output data is scaled between 0 and 1

Introduction to ANNs (*cont.*)

$$N3 = \frac{1}{1 + exp(-(N1 \cdot W13 + N2 \cdot W23 + B3))}$$

$$N5 = \frac{1}{1 + exp(-(N3 \cdot W35 + N4 \cdot W45 + B5))}$$



Example ANN with 2 input nodes (N1, N2), 2 hidden nodes (N3, N4) and a single output (N5).

Training the Network

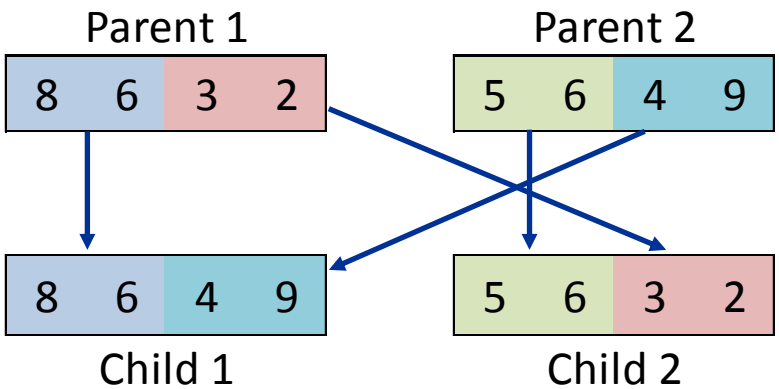
- As in the paper by *Salam et al*, both training by back propagation (BP) and genetic algorithm (GA) were examined
- In all cases, the objective was to minimize SPE (standard percent error) of predicted cost
- Back Propagation
  - Uses gradient descent to find an optimal solution for the network
  - Network is initially populated with arbitrary weights and biases
  - Weights are back-adjusted based on the output error iteratively through each point in the data set. This is repeated until the stopping conditions are met.
  - Advantage: relatively straightforward optimization routine
  - Disadvantage: risk of being trapped in a local optimum

Training the Network - Genetic Algorithms

- Inspired by biological systems and Darwin's Theory of Evolution
- Genetic operators, based on reproduction and mutation, are applied to the population of the ANN's weights and biases
- Process:
  - Create initial population of random rational numbers between an upper and lower bound where <u>each member is a full set of weights and biases</u>
  - Rank each member based on a fitness calculation; one with lowest output error is #1
  - Select members of the mating pool using *k*-way tournament selection until the size of the mating pool equals the population size
    - Ensures fitter members will have more copies in the pool
  - Members of the mating pool are randomly paired off to reproduce two offspring
    - Top ranked member is guaranteed to be selected at least once

Training the Network - Genetic Algorithms

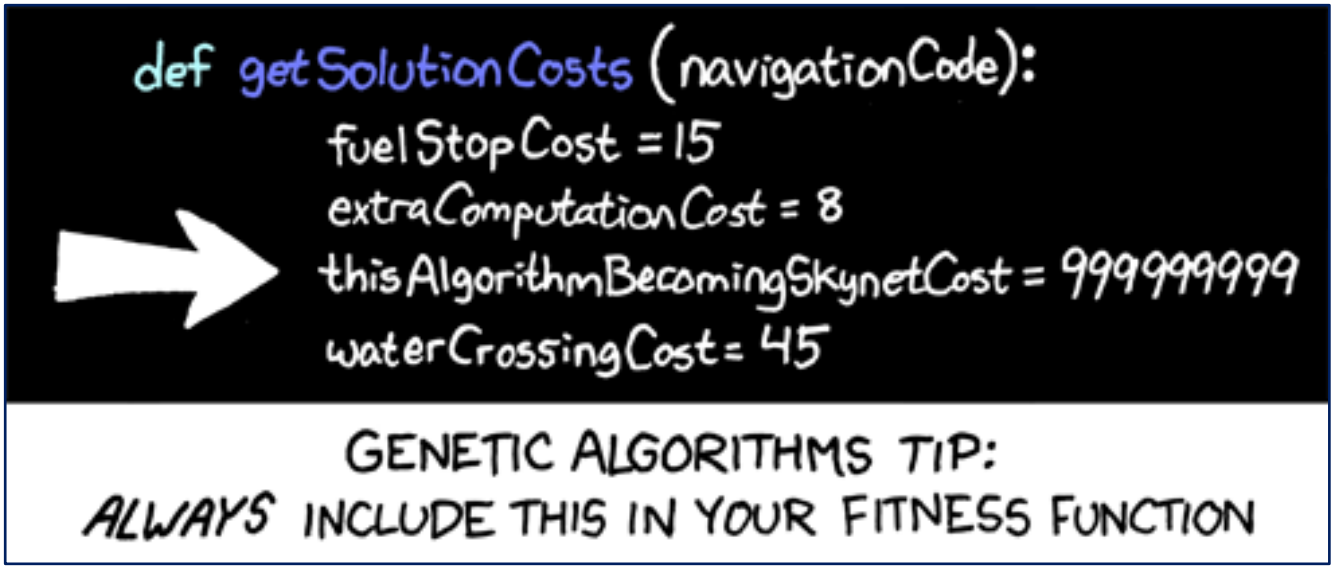
- Process (*cont.*):
  - Values of the weights and biases for the offspring are determined by one of four (randomly determined) crossover operations applied to the parents' values. Example:



  - Lastly, some values in the children sets are "mutated" based on a user-defined probability
  - Top ranked member from previous generation is passed through unmodified to the next generation
  - The new generation is ranked by fitness and the process repeats until the stopping criteria are met
- Population size, tournament selection size, and mutation rate parameters are critical to the success of the algorithm and can be tuned by the user

Training the Network - Genetic Algorithms

- Advantages:
  - Avoid local optima better than back propagation
  - Have been shown to generate good predictive power for neural networks
    - Outperformed the BP-ANNs in almost all cases
- Disadvantages:
  - Require tuning of parameters to be most effective
  - Computationally demanding (can be slow)
  - Self-awareness risk?



(taken from xkcd.com)

Training the Network – Stopping Conditions

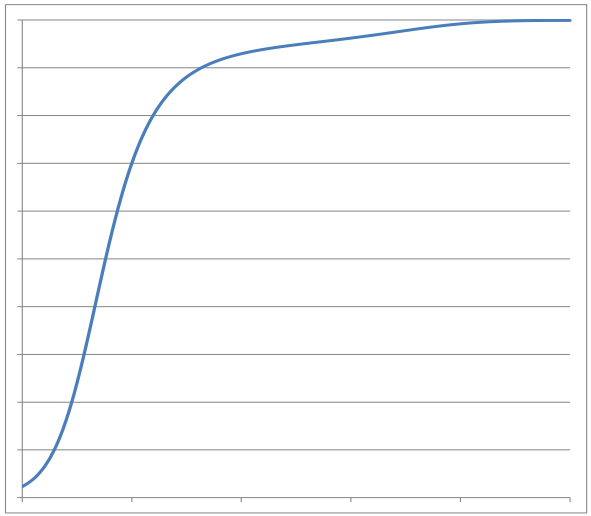
- Relevant to both back-propagated and genetic algorithm ANNs is the issue of when to end the process
- Models are coded to run for a specified number of iterations or to stop when a SPE goal is achieved
- User can stop the routine at any time
- Question is a bit trickier with GA-ANNs
  - BP-ANNs tend to settle on a recognizable steady state where additional improvement is very unlikely/minimal
  - GA-ANNs can at times make significant improvements on a single iteration after having slight/no improvement for hundreds of iterations
- Because the process is slow and not continuously monitored (if at all), created a graphing function that records SPE by iteration
  - Aids the analyst in determining when to stop the routine

Training – Lessons Learned

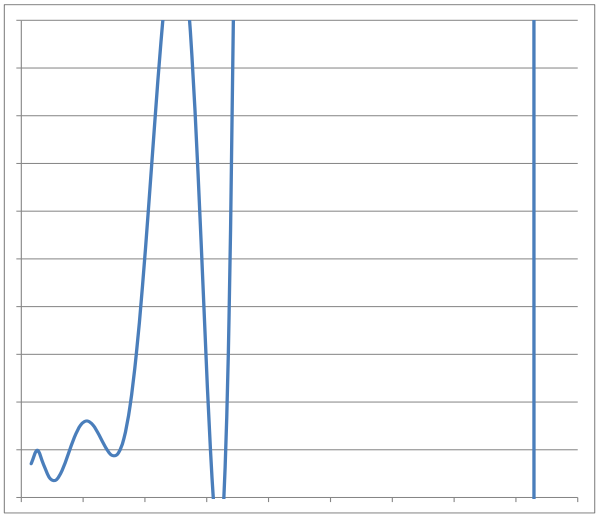
- ANNs in general:
  - Don't make the network any larger than necessary. In most cases, only one or two nodes in the hidden layer was found to be adequate.
  - Avoid using more than one logical variable (do not seem to work well with sigmoid functions).
- Genetic Algorithms
  - Population size of 200 seemed to work best
    - May not be the case on all GA models- dependent on code efficiency and computing power
  - Recommend 2-way or 3-way tournaments for mating pool selection
    - Higher numbers reduce diversity of the mating pool
  - Initial population is critical- if SPE is not improving much, start over with a new population
  - Mutation rate value of 0.05 to 0.15 is recommended

ANN Applicability to Cost Estimating

- Primary advantage is the flexibility
  - Do not need to solve and compare multiple equation forms
- Overfitting is a potential concern. However:
  - the resultant curve/area forms are quite subtle
  - for example, a single parameter CER derived using an ANN with three hidden nodes would create a sigmoid function with 10 estimated constants (6 weight, 4 bias)- NOT comparable to more typical equation forms with 10 constants:



ANN-derived cost curve, with 10 estimated constants



9th-order polynomial, regressed on the same data

ANN Applicability (*cont.*)

- Degrees of Freedom Question
  - It became evident that ANN approach was not developed by statisticians when we were unable to find any guidance on D.o.F. assumptions in the available literature
    - *Salam, et al* does not address either standard error or standard percent error
  - No choice (currently) other than to assume

    $df = n - estimated\ constants - 1$ when calculating percent error
  - Very limiting in cases where only a relatively small number of data points are available
- Is the result of an ANN-derived CER an unbiased estimator?
  - Literature search on this question was inconclusive
  - As the objective was to minimize percent error of predicted cost, it was unsurprising when we started that the results had a tendency to over-estimate the data set (the absolute value of the percent error of a single data point cannot be greater than 100% if it is overestimated).
  - Applied solution to the bias problem was to add a penalty term to the fitness function
    - Penalty is greater the further the sum of the percent errors is from zero
    - As would be expected this has a negative effect on SPE

Results

- Compared to *Salam, et al*
  - First ran the ANNs with the data from their paper primarily to validate that the code was functioning the way it should
  - Paper tested the two ANN methods using 13 data points of main landing gear cost with weight, aircraft maximum takeoff weight (MTOW), and height as the input variables *
  - The ANNs were run for three different cases, each with 3 different data points removed from the training data
    - The 3 removed points were used as test cases to evaluate the algorithm
  - Results from GRC-coded algorithm compared favorably
- ANNs trained using genetic algorithms outperformed back-propagated

Training	Set 2	BP-ANN	Article	BP-ANN	GRC	GA-ANN	Article	GA-ANN	GRC
Program	Cost	Estimate	% Error	Estimate	% Error	Estimate	% Error	Estimate	% Error
1	63816	63820	0.01	64243	0.66	62400	2.22	64617	1.26
2	69465	69299	0.24	64484	7.17	64409	7.28	65003	6.42
4	125657	125878	0.18	124874	0.62	117966	6.12	124625	0.82
5	78516	78766	0.32	77054	1.86	82251	4.76	77802	0.91
6	117834	117436	0.34	105822	10.19	105175	10.74	103774	11.93
8	103552	103944	0.38	106925	3.25	110559	6.77	105152	1.55
10	114082	108483	4.91	107080	6.13	107305	5.94	105279	7.72
11	102595	107277	4.56	107067	4.35	104773	2.12	105265	2.60
12	104400	105177	0.74	107541	3.01	107047	2.54	105793	1.33
13	104408	103989	0.40	107604	3.06	105380	0.93	105927	1.46
3	73794	81744	10.77	79370	7.56	75425	2.21	73988	0.26
7	104635	112593	7.61	106978	2.24	111013	6.10	105175	0.52
9	103173	109321	5.96	107074	3.78	110109	6.72	105272	2.03

* Weight and MTOW are unfortunately highly correlated (0.84).

Results (*cont.*)

- Developed ~30 CERs using GA-ANN and compared results to CERs derived using ZMPE (zero bias/minimum percent error) regression
- On average, the GA-ANN CERs reduced the SPE by about 8% compared to ZMPE-derived
  - A portion of that reduction is very likely attributable to the higher bias relative to the ZMPE method
- Results relative to ZMPE varied
  - In 2 cases ZMPE regression produced a lower SPE

Summary/Recommendations

- Artificial neural networks using genetic algorithms can be used to develop cost estimating relationships
- Results are mixed
  - An approach to consider when unable to attain satisfactory results using other methods
  - Not quite ready to be used as first methodology
- For more information, contact elizabeth.r.turnbull@nasa.gov

Forward Work

- Need more speed
  - Current code (Excel VBA) can be improved
  - MATLAB may be better
- Refine bias penalty
- Simplify transfer of CER to a cost model