Relearn: connectivity_update() **Single Parameter Strategy: Use first points found** **Multi Parameter Strategy: Increasing cost**

		model	total cost	percent cost	RSS	Adj. R^2	SMAPE	RE	Comments
sparse	***full matrix***	+ -27.2757 + 4.62272e-06*log2^1(p)*(size^1)*log2^1(size)*(t^-2.66667)	9801124.17	100.00	1150000.00	0.97			*Sparse Modeler shows the same result as the old modeling approach with the full matrix.*
	min	+ -1606.52 + 228.933*log2^1(p) + 0.00752181*(size^1)*log2^1(size) + 0.873338*(t^-2.66667)	1199777.91	12.24	12400.00	0.99			
	min +1	+ -1580.3 + 224.93*log2^1(p) + 0.00706355*(size^1)*log2^1(size) + 0.946491*(t^-2.66667)	1199983.56	12.24	20200.00	0.99			
	Min +1 with theta = 0.1	+ -1609.39 + 229.256*log2^1(p) + 0.00754542*(size^1)*log2^1(size) + 0.866074*(t^-2.66667)	1245492.98	12.71	12900.00	0.99			*This experiment is based on the cheapest measurements with a theta value of 0.1. The result shows that the lower theta value does not help to increase the prediction accuracy for the other terms. These are also not the most expensive measurements, as the cost is calculated by c=p*time.*
	Min +2	+ -1484.68 + 215.169*log2^1(p) + 0.00590998*(size^1)*log2^1(size) + 1.06143*(t^-2.66667)	1200221.49	12.25	41000.00	0.98			
	Min +3	+ -1351.84 + 203.278*log2^1(p) + 0.00448621*(size^1)*log2^1(size) + 1.16863*(t^-2.66667)	1200493.61	12.25	67000.00	0.97			
	Min +4	+ -1227.75 + 193.003*log2^1(p) + 0.00324528*(size^1)*log2^1(size) + 1.24263*(t^-2.66667)	1200799.63	12.25	90000.00	0.97			
	Min +5	+ -14.0638 + 3.79907e-06*log2^1(p)*(size^1)*log2^1(size)*(t^-2.66667)	1201182.58	12.26	344000.00	0.87			*With 5 or more additional points we get the same result as for the full matrix with the selected strategies. Less points are not sufficient to get the same model as the full matrix.*
	Min +6	+ -12.8326 + 3.79335e-06*log2^1(p)*(size^1)*log2^1(size)*(t^-2.66667)	1201635.12	12.26	344000.00	0.88			
	Min +7	+ -12.1561 + 3.79026e-06*log2^1(p)*(size^1)*log2^1(size)*(t^-2.66667)	1202088.55	12.26	344000.00	0.89			
	agent	+ -94.466 + 15.6015*log2^1(p) + 0.00020889*(size^1)*log2^1(size) + 1.10006*(t^-2.66667)	104598.10	1.07	135.00	1.00			*This experiment uses the measurement points suggested by the parameter value selection ai for modeling. The results show that the ai solution can not be generically applied to each application/problem as we do not get the same result as with the full matrix. Though it is also improtant to consider that it uses only 1% of the cost compared to 12% (min+5).*
normal	***full matrix***	+ -27.2757 + 4.62272e-06*log2^1(p)*(size^1)*log2^1(size)*(t^-2.66667)	9801124.17	100.00	1150000.00	0.97			

size	**9000**	834.353	1186.72	1877.45	2125.61	2535.05	t = 0.1
	8000	721.454	1026.09	1640.74	1879.76	2135.01	
	7000	616.449	869.846	1409.64	1623.84	1853.32	
	6000	509.49	714.298	1169.15	1376.47	1557.19	
	5000	406.181	574.101	939.328	1122.32	1276.17	
		32	**64**	**128**	**256**	**512**	
		p					

size	**9000**	125.576	165.549	252.633	292.946	331.335	T = 0.2
	8000	109.749	145.304	222.521	257.346	292.461	
	7000	94.1432	125.065	191.532	227.713	253.495	
	6000	79.7909	105.774	162.859	189.832	215.294	
	5000	65.2061	86.0282	133.189	156.264	177.149	
		32	**64**	**128**	**256**	**512**	
		p					

size	**9000**	40.5282	52.4854	72.8099	83.302	93.7872	T = 0.3
	8000	35.823	46.1819	64.4428	73.7924	83.0524	
	7000	31.0822	40.1958	55.8965	64.8183	72.2867	
	6000	26.3713	34.1409	48.2509	55.6438	62.3198	
	5000	21.8088	28.2151	39.4186	46.3484	52.2848	
		32	**64**	**128**	**256**	**512**	
		p					

size	**9000**	18.1768	23.4612	29.4351	34.3621	38.7553	T = 0.4
	8000	16.0401	20.8859	26.3724	30.7503	34.839	
	7000	14.1697	18.467	23.0918	34.3621	31.1879	
	6000	11.9672	15.4639	19.8202	30.7503	27.0231	
	5000	9.9311	12.8381	16.4604	27.3721	23.2562	
		32	**64**	**128**	**256**	**512**	
		p					

size	**9000**	9.56317	12.3732	14.9944	17.9984	20.5119	T = 0.5
	8000	8.50372	11.0746	13.498	16.2909	18.7238	
	7000	7.43541	9.67296	11.7859	14.4767	17.0101	
	6000	6.4267	8.37619	10.2848	12.9167	15.1167	
	5000	5.39257	7.0709	8.78632	11.2207	13.4616	
		32	**64**	**128**	**256**	**512**	
		p					

size = c*p
In order to analyze the measurements with weak scaling we have to calculate the real size out of the local value.

32*6000 64*6000
32*5000 64*5000

* Data was taken from connectivity_update() measurements with full matrix

In normal productive use theta values are only between 0 and 0.3. Even though the differences in the data are really big, selecting measurements with a lower theta that are more expensive does not help to increase the accuracy of the models.
In general it seems like we need more additional points instead! It looks like we need 5 additional points to correctly identify the model for relearn with using the cheapest points as a base for modeling and then adding points one by one from the cheapest to the expensivest point (increasing cost).

This should be further investigated. For additive terms we check the term contribution, so we should have already check this and can be sure that there it should not be a problem. For multiplicative functions this is not the case.
We need to check if one parameters contribution is 99% how many points in addition we generally need for a correct model.
So we also have an answer for cases like this. So our modeling strategy/ approach works for all kind of scenarios that are known to us…